

---

Intelligente Codes auf Basis von sensorischen Tinten mittels  
Smart Devices und ihre Evaluierung über das Internet der Dinge

---

Dissertation zur Erlangung des akademischen Grades  
**DOKTOR DER INGENIEURWISSENSCHAFTEN (DR.-ING.)**

An der  
Fakultät für Elektrotechnik, Informationstechnik und Medientechnik  
der Bergischen Universität Wuppertal

vorgelegt von Herrn  
**MUSTAFA BILGIN, M.Ed.**

geboren in  
BIELEFELD

Datum der Einreichung: 30. Dezember 2020

Datum der mündlichen Prüfung: 18. Mai 2021

1. Gutachter: Professor Dr.-Ing. Johannes Backhaus
2. Gutachter: Professor Dr.-Ing. Peter Urban
3. Gutachter: Professor Dr. rer. nat. Daniel Neumaier

Die Dissertation kann wie folgt zitiert werden:

urn:nbn:de:hbz:468-20210712-090734-4

[<http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20210712-090734-4>]

DOI: 10.25926/yqb2-0w50

[<https://doi.org/10.25926/yqb2-0w50>]

»Bilginler suyu bol bir yere benzerler. Nereye kazma vurulursa oradan su çıkar.«<sup>1</sup>

---

<sup>1</sup> „Gelehrte ähneln Orten, reich an Wasser. Wo auch immer man ansetzt zu graben, quillt Wasser hervor.“  
Has Hacib, Yusuf. (2010). *Kutadgu Bilig*. İskele Verlag. Istanbul. S. 41.

# Danksagung


Ich möchte Herrn Professor Dr.-Ing. Johannes Backhaus und Herrn Professor Dr.-Ing. Peter Urban für die Unterstützung meines Promotionsvorhabens danken. Außerdem möchte ich Herrn Professor Dr. rer. nat. Daniel Neumaier für seine fachliche Unterstützung danken.

Vom Herzen danke ich meinen Eltern Rabia und Ali und meinem Bruder Selçuk, die immer für mich da waren und sind. Çok teşekkür ederim! Ich bedanke mich herzlichst bei meiner Partnerin Bettina für ihre Unterstützung und Liebe. Köszönöm szépen! Zudem bedanke ich mich bei meinem Sohn İlyas, der für uns alle eine wunderschöne Bereicherung ist und uns sehr viel Liebe gibt. Herzlichen Dank!

Ein besonderer Dank gilt an meine Assistentinnen aus der Chemie Kristin Wörpel, Jaqueline Schrank und Nicole Paulke, die mir in Angelegenheiten der Chemie fachlich im Labor zur Seite standen und mich immer fachwissenschaftlich unterstützt haben. Zudem danke ich meinen Assistenten aus der IT und Informatik Thomas Dridiger und Kai Mortsiefer, die mich mit ihrer Kompetenz, fachwissenschaftlich tatkräftig unterstützt haben. Ebenfalls gilt ein Dank meinen Studierenden, die ich im Verlaufe ihres Studiums begleiten durfte.

Abschließend möchte ich mich bei allen Menschen bedanken, die mich auf meinem langen und hürdenreichen Weg hierhin unterstützt und begleitet haben. Danke!

Wuppertal, Dezember 2020

  
(Mustafa Bilgin)

# Zusammenfassung

Die automatisierte Produktidentifikation und Datenerfassung setzt seit vielen Jahren maschinenlesbare Codes ein, die sich unter anderem in der Produktion, Materialwirtschaft und Logistik aufgrund ihrer hohen optischen Speicherdichte und schnellen sowie effizienten Lesbarkeit etabliert haben. Die binären Speicherinhalte eines Codes werden mithilfe von optischen Lesegeräten oder dafür entwickelten Applikationen ausgelesen. Ein solcher Code beinhaltet statische, sich nicht verändernde Inhalte. Diese Art der Codes wird vielfach auf verpackten Waren appliziert.

Im Zusammenhang mit einer immer besser werdenden Verfolgung des Weges von Waren bis an ihren Bestimmungsort steigt die Bedeutung dynamischer und somit sekundärer Daten. Solche Daten können bspw. Auskunft über den Zustand von Waren, wie während des Transports, geben. Anhand dynamischer Daten kann bspw. nachverfolgt werden, ob und inwieweit ein Packgut während seiner Transporthistorie schädlichen Umwelteinflüssen ausgesetzt war und, im Fall von Lebensmitteln, ob die verpackte Ware weiterhin für den Verzehr geeignet ist. Für den Einsatzzweck der Erfassung von äußeren Einflussfaktoren werden vermehrt Indikatoren eingesetzt, die bspw. auf temperatur- oder feuchtigkeitsempfindlichen Farbstoffen basieren. Diese Indikatoren werden auf ein Haftpapier zum Applizieren auf eine Ware oder direkt auf eine Verpackung gedruckt. Der Einsatz dieser Indikatoren gibt jedoch häufig lediglich eindimensionale Aussagen über jeweils einen Einflussfaktor Auskunft und ist nicht für weitere Indikatoren erweiterbar.

Im Rahmen dieser Dissertation wird ein mehrdimensionales Sensorsystem entwickelt. Dieses beruht auf einem mehrdimensionalen und erweiterbaren intelligenten Code. Der eigens entwickelte intelligente Code wird mit mehreren sensorischen Tinten, anhand des kontaktlosen Inkjetdrucks auf ein geeignetes Haftpapier appliziert und auf seine spezifischen Eigenschaften hin untersucht. Grundlage dieses Sensorsystems sind die aus unterschiedlichen sensorischen Farben bestehenden dynamischen Bereiche. Diese dynamischen Bereiche umranden den statischen Code, der aus dieser Kombination hervorgehend als intelligenter Code fungiert und bei Bedarf (On-Demand) über eine serverseitige Leseapplikation mittels eines Endgeräts (Smart Device) erfasst, analysiert und ausgewertet wird. Zu den Smart Devices zählen bspw. Smartphones, Tablets, Smart Glasses und weitere kommunikationsfähige Endgeräte. Diese können mithilfe der in den Smart Devices integrierter Kameras den intelligenten Code erfassen und die erfassten Informationen, mittels Internet der Dinge – miteinander vernetzte Objekte in einem autonomen Netzwerk – austauschen. Für das Erfassen, Analysieren und Auswerten des intelligenten Codes stellt der Benutzer On-Demand mittels Smart Device eine Verbindung zum Server her. Hierzu wird über eine lokale Client Applikation (Thin Client), welche sich standardmäßig auf dem Smart Device befindet, eine Remote-Verbindung zur serverseitigen eigens entwickelten Leseapplikation (Client-Server-Model) aufgebaut. Die Leseapplikation wiederum greift verschlüsselt auf die integrierte Kamera des Smart Devices zu, wobei die Codeerfassung, Analyse und Datensicherung über das maschinelle Sehen (Computer Vision) erfolgt. Die Serveranwendung berücksichtigt die Spezifikationen und Anforderungen der unterschiedlichen Smart Devices verschiedenster Fabrikate. Hierdurch ist der Leseprozess geräteunabhängig und dynamisch einsetzbar.

Sensorische Tinten, Inkjet, 2D-Code, Computer Vision, Smart Device, Colormangement

# Abstract

Automated product identification and data collection have been using machine-readable codes for many years. These codes have established in production, materials management and logistics, along with other areas due to their high optical storage density including fast and efficient readability. The binary contents of a symbol code can be read by means of optical readers. This type of code contains static, non-changing content and is often applied on packaged goods.

Within the scope of this PhD thesis, a multidimensional sensor system is to be developed. This system is based on a multidimensional and extendible intelligent code. Moreover, the specially developed intelligent code was printed with printable sensors through an inkjet printer. Dynamic areas consist of various printable sensors that form the basis of the aforementioned sensor system. These dynamic areas surround the static symbol code and form the intelligent code. Through a server-side application, this intelligent code can be detected, analyzed and evaluated. Smart devices such as smartphones, tablets, smart glasses and other communication-enabled devices are able to connect to the server, access the code reading application and scan the code using the integrated camera.

More precisely, a local client application (thin client), which is located on the smart device as default, establishes a remote connection to the server-side reading application (client-server model). The reading application accesses the smart device's integrated camera in encrypted form. The reading application accesses the smart device's integrated camera in encrypted form. During this process, the code is captured and analyzed by computer vision. The server application takes into account the different camera color information of the end devices so that the reading process is device-independent and dynamic.

Printable Sensors, inkjet, 2D code, computer vision, client-server, color management

# Inhaltsverzeichnis

Danksagung .....	I
Zusammenfassung .....	II
Abstract .....	III
Notation.....	IV
Abkürzungsverzeichnis.....	V
Kapitel 1 .....	1
<b>Einleitung</b> .....	1
1.1 Zielsetzung.....	2
1.2 Ausgangssituation.....	5
1.3 Konzept.....	7
1.4 Aufbau .....	9
Kapitel 2 .....	10
<b>Stand der Forschung</b> .....	10
2.1 Sensorische Farben .....	10
2.2 Intelligente Code-Technologien .....	13
2.3 Smart Labels.....	20
Kapitel 3 .....	27
<b>Methoden</b> .....	27
3.1 Materialien .....	27
3.1.1 Additive.....	28
3.1.2 Substrate .....	29
3.2 Bestimmung der Tintenanforderungen .....	30
3.2.1 Viskosität .....	30
3.2.2 Oberflächenspannung.....	30
3.2.3 Partikelgrößenverteilung.....	31
3.2.4 Filtrationsprozess.....	31
3.3 Inkjet Drucksystem .....	32
3.4 Farbmétrische Untersuchung und Datenerhebungen .....	33
3.5 Lichtarten.....	34

3.5.1 Untersuchung der Lichtechtheit .....	34
3.6 Standardisierung.....	35
3.7 Programmiertechnische Methoden .....	36
3.7.1 Geräteneutrale Auswertung.....	36
3.7.2 Colormanagement.....	36
3.7.3 Auswahl und Bewertung geeigneter Smart Devices .....	38
3.7.4 Helligkeits- und Kontrastanpassung.....	39
3.7.5 Farbkorrektur und -transformation.....	39
3.8 Soft- und Hardwarearchitektur.....	41
<b>Experimentelle Umsetzung.....</b>	<b>44</b>
<b>Kapitel 4 .....</b>	<b>45</b>
<b>Sensorische Tinten .....</b>	<b>45</b>
4.1 Grundlagen sensorischer Farben .....	47
4.2 Farbmétrische Grundlagen .....	47
4.2.1 Farbreiz.....	47
4.2.2 Farbvalenzen.....	48
4.2.3 Farbabstand.....	48
4.3 Substrate .....	49
4.4 Inkjetdruck.....	50
4.4.1 Grundlagen der Viskosität .....	51
4.4.2 Grundlagen der Oberflächenspannung .....	52
4.4.3 Reynolds-, Weber- und Ohnesorge-Zahl .....	52
4.5 Chemische Grundlagen .....	55
4.5.1 Anregung von Atomen .....	55
4.5.2 Orbitalmodell.....	56
4.5.3 Anregung von Molekülen .....	57
4.6 Untersuchung sensorischer Farben.....	59
4.6.1 Hydrochrom .....	59
4.6.2 Photochrom .....	67
4.6.3 Halochrom.....	71
4.7 Entwicklung sensorischer Tinten .....	75
4.7.1 Hydrochrom .....	75
4.7.2 Photochrom .....	79



4.7.3 Halochrom .....	85
4.8 Untersuchung eines geeigneten Haftpapiers .....	88
4.8.1 Hydrochrome .....	88
4.8.2 Photochrome .....	89
4.8.3 Halochrome .....	89
4.8.4 Zusammenfassung .....	90
Kapitel 5 .....	91
<b>Aufbau und Funktion eines intelligenten Codes .....</b>	<b>91</b>
5.1 Grundlagen von Code-Technologien .....	91
5.1.1 DataMatrix Code .....	93
5.1.2 Lesbarkeit .....	95
5.1.3 Color Lookup Table .....	95
5.1.4 CCD und CMOS .....	96
5.1.5 Arten von Codes .....	97
5.2 Kriterien für einen intelligenten Code .....	101
5.3 Entwicklung und Aufbau eines intelligenten Codes .....	103
5.4 Ausgabe des intelligenten Codes .....	105
Kapitel 6 .....	106
<b>Softwarearchitektur .....</b>	<b>106</b>
6.1 Grundlagen der Bildverarbeitung und Computer Vision .....	106
6.1.1 Pixel .....	107
6.1.2 Schwellenwertverfahren .....	108
6.1.3 Morphologische Transformationen .....	109
6.1.4 Lineare Filteroperationen (Gradient) .....	110
6.1.5 Hough-Transformation .....	110
6.1.6 Suzuki-Abe Algorithmus .....	112
6.1.7 Hierarchische Konturen .....	113
6.1.8 Konvexe Hüllen .....	113
6.1.9 Perspektivische Transformation .....	114
6.2 Erkennungs-, Analyse- und Auswertungsprozess .....	115
6.3 Erkennungsprozess des intelligenten Codes .....	117
6.4 Separierung des intelligenten Codes .....	122
6.5 Auswerteeinheit .....	124

6.6 Beseitigung von Störparametern .....	127
6.6.1 Farbanpassung.....	127
6.6.2 Helligkeits- und Kontrastanpassung.....	130
6.6.3 Geräteerkennung.....	131
6.6.4 Colormanagement.....	133
6.7 Kommunikationstechnologien.....	136
6.7.1 Internettechnologien .....	136
6.7.2 Internet der Dinge.....	137
6.7.3 Datensicherheit und Verschlüsselung.....	137
6.8 Serverarchitektur .....	140
6.9 Praxisnahe Umsetzung.....	140
Kapitel 7 .....	143
<b>Validierung.....</b>	<b>143</b>
7.1. Gesamtprozess.....	144
7.2. Grenzen.....	148
7.3. Risiken .....	149
Kapitel 8 .....	150
<b>Ausblick .....</b>	<b>150</b>
<b>Anhang.....</b>	<b>151</b>
<b>Literaturverzeichnis .....</b>	<b>182</b>
<b>Abbildungsverzeichnis .....</b>	<b>197</b>
<b>Tabellenverzeichnis .....</b>	<b>200</b>
<b>Veröffentlichungen .....</b>	<b>201</b>
<b>Lebenslauf.....</b>	<b>203</b>

# Notation

Zeichen	Bezeichnung	Referenz
nm	1 nm = $10^{-9}$ m	Seite 34
mJ/cm <sup>2</sup>	Bestrahlungsdosis	Seite 34
mW/cm <sup>2</sup>	Bestrahlungsstärke	Seite 34
$\rho$	Dichte	Seite 53
$\Delta p$	Druckdifferenz	Seite 31
$p_{\max}$	Druckmaximum	Seite 30
d	Durchmesser bspw. Inkjet-Düse / Tintenkanal	Seite 54
e <sup>-</sup>	Elektron	Seite 55
eV	Elektronenvolt, 1eV = $1,602\ 176\ 634 \cdot 10^{-19}$ J	Seite 55
E	Energie	Seite 55
$\Delta E_{76}$	Farbabstand, 1976	Seite 48
$\Delta E_{00}$	Farbabstand, 2000	Seite 48
A	Fläche	Seite 51
f	Frequenz	Seite 55
v	Geschwindigkeit	Seite 54
$\Delta E_{GF}$	Grenzflächenenergie	Seite 52
$\Delta W_{GF}$	Grenzflächenmolekül	Seite 52
K	Kelvin	Seite 38
L*a*b*	Koordinaten im CIE L*, a*, b* - Farbraum	Seite 48
UV	Lichtart	Seite 34
A <sub>l</sub>	Lichtart	Seite 40
m <sub>l</sub>	Magnetische Quantenzahl des Drehimpulses	Seite 56
M	Menge	Seite 113
N	Newton, 1 N $\equiv$ 1 kg·m/s <sup>2</sup>	Seite 52
k	Normierungskonstante	Seite 38
D50	Normlichtart	Seite 151
D65	Normlichtart	Seite 34
TL84	Normlichtart	Seite 34
$\sigma$	Oberflächenspannung in mN/m	Seite 52
Oh	Ohnesorge-Zahl	Seite 53
$\gamma$	Photonen	Seite 67
h	Planck'sches Wirkungsquantum, $4,135\ 6692\ (12) \cdot 10^{-15}$ eV	Seite 55
p <sup>+</sup>	Proton	Seite 55
r	Radius	Seite 31
S( $\lambda$ )	Relative spektrale Verteilungsfunktion einer Lichtart	Seite 47
$\varphi(\lambda)$	Relative Spektralverteilung der Farbreizfunktion	Seite 47
Re	Reynolds-Zahl	Seite 53
$\dot{\gamma}$	Schergeschwindigkeit	Seite 51
F	Scherkraft	Seite 51

$\tau$	Scherspannung in N/m <sup>2</sup>	Seite 51
H	Schnittpunkt	Seite 111
$\beta(\lambda)$	Spektraler Strahldichtefaktor	Seite 47
$\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$	Spektralwertfunktionen, CIE 2°-farbmetrischer Normalbeobachter	Seite 48
$m_s$	Spinquantenzahl, Eigendrehimpuls eines Elektrons	Seite 56
I	Strahlungsintensität	Seite 84
$\vartheta$	Temperatur in °C	Seite 76
$\eta$	Viskosität in mPa·s	Seite 51
We	Weber-Zahl	Seite 53
$\Delta\lambda$	Wellenlängenintervall	Seite 48

Tabelle 1 Notation der verwendeten Zeichen und Symbole

## Abkürzungsverzeichnis

Abkürzung	Bezeichnung	Referenz
API	Application Programming Interface	Seite 40
AIM	Automatic identification industry	Seite 97
BSED	Backscattered Electron Detector	Seite 31
BOM	Browser Object Model	Seite 131
CSS	Cascading Style Sheets	Seite 115
CLUT	Color Lookup-Table	Seite 95
CMS	Colormanagementsystem	Seite 133
CIJ	Continuous Inkjet	Seite 50
CSRF	Cross-Site-Request-Forgery	Seite 138
XSS	Cross-Site-Scripting	Seite 139
CMYK	Cyan, Magenta, Yellow, Key	Seite 33
DTLS	Datagram Transport Layer Security	Seite 137
DDoS	Distributed-Denial-of-Service	Seite 149
DNS	Domain Name System	Seite 136
DOD	Drop-On-Demand	Seite 50
EU	Europäische Union	Seite 60
EAN	European Article Number	Seite 16
GPS	Global Positioning System	Seite 16
GUI	Graphical User Interface	Seite 115
HTML	Hypertext Markup Language	Seite 115
PHP	Hypertext Preprocessor	Seite 116
HTTP(S)	Hypertext Transfer Protocol (Secure)	Seite 138
IR	Infrarotstrahlung	Seite 47
ICC	International Color Consortium	Seite 37
IETF	Internet Engineering Task Force	Seite 42
IoT	Internet of Things	Seite 137
IP	Internet Protocol	Seite 136
IPv4	Internet Protocol Version 4	Seite 136

IPv6	Internet Protocol Version 6	Seite 136
IVCT	Intervallenz Charge Transfer	Seite 68
JS	JavaScript	Seite 115
LED	Light-Emitting Diode	Seite 23
NFC	Near Field Communication	Seite 13
NIP	Non Impact Printing	Seite 50
OSI	Open Systems Interconnection	Seite 136
OS	Operating System	Seite 36
PIL	Python Imaging Library	Seite 134
QR	Quick Response	Seite 91
RFID	Radio-Frequency Identification	Seite 20
REM	Rasterelektronenmikroskop	Seite 31
RI	Rendering Intent	Seite 37
RGB	Rot, Grün und Blau	Seite 33
(S)RTP	(Secure) Real-Time Transport Protocol	Seite 42
SSL	Secure Sockets Layer	Seite 138
SIP	Session Initiation Protocol	Seite 42
SDK	Software Development Kit	Seite 14
SQL	Structured Query Language	Seite 42
STP	Synchronizer Token Pattern	Seite 139
TTI	Time-Temperature-Indicator	Seite 5
TCP	Transmission Control Protocol	Seite 136
TLS	Transport Layer Security	Seite 115
URL	Uniform Resource Locator	Seite 138
UDP	User Datagram Protocol	Seite 137
VIS	Visible Spectrum	Seite 47
WebAR	Web Augmented Reality	Seite 150
WebRTC	Web Real-Time Communication	Seite 42
WebVR	Web Virtual Reality	Seite 150
W3C	Wide Web Consortium	Seite 42

*Tabelle 2 Abkürzungsverzeichnis*

## Kapitel 1

# Einleitung

Sensoren sind die Schnittstelle nach außen. Sie interagieren zwischen der Datenverarbeitung und der Außenwelt eines Systems. Im täglichen Leben sind Sensoren in unterschiedlichsten Gegenständen, wie bspw. elektrischen Haushaltsgeräten, Rauchmeldern, automatischen Türen, Aufzügen, Smart Devices, in Fahrzeugen und, neben vielen weiteren Einsatzbereichen, auch in der Luft- und Raumfahrt allgegenwärtig. Der Terminus *Sensus* lässt sich dabei aus dem Lateinischen als *Sinneseindruck* oder *Wahrnehmung* übersetzen, wohingegen *Sensorik* die Wissenschaft und Anwendung von Sensoren zur quantitativen oder qualitativen Messung sowie der Kontrolle von Eigenschaften, Zusammensetzungen und deren Veränderung bezeichnet (BAM, 2018). Sensoren werden eingesetzt, um physikalische oder chemische Faktoren unmittelbar in einem System (bspw. in Flüssigkeiten) zu bestimmen und schließlich in elektrische Signale umzusetzen (Scholz, 2005, S. 32). So werden für den Einsatz von elektrischen Sensoren eine Energieversorgung und für die anschließende digitale Datenverarbeitung ein elektrisches Ausgangssignal vorausgesetzt. Dabei kann es sich bei den physikalischen Parametern unter anderem um Licht einer spezifischen Wellenlänge, Feuchtigkeit, Temperatur, Druck oder, bei (bio-)chemischen Sensoren, um den pH-Wert sowie chemische Stoffe wie Gase, Ionen oder um biologische Moleküle handeln.

Andere Sensoren, wie die hier untersuchten sensorischen Tinten, die unter anderem auch als Smart Materials bezeichnet werden, basieren auf unterschiedlichen sensorischen Farben (*Definition in Kapitel 2.1*) und können bei einer ansprechenden Anpassung auf das Druckverfahren, wie bspw. dem Inkjet-Flexo-, oder Siebdruck auf geeignete Substrate appliziert werden. Diese sensorischen Farben reagieren auf (bio)chemisch-physikalische Einflussfaktoren z. B. aus der Umwelt mit einer Stimuli induzierten Farbveränderung. Der Einsatz der in dieser Arbeit verwendeten sensorischen Tinten setzt voraus, dass die verwendeten sensorischen Farben nach einer Reaktion dementsprechend unumkehrbar (irreversibel) in ihrem neuen Zustand verbleiben und nicht in ihren ursprünglichen Zustand zurückführbar sind (reversible). Das Ausgangssignal des Sensors ist eine Farbveränderung. In der vorliegenden Arbeit wurden sensorische Farben entwickelt, die mit einer irreversiblen Farbveränderung auf einen jeweiligen Stimulus reagieren. Die Quantifizierbarkeit des entsprechenden Farbwechselverhaltens der sensorischen Farben kann angesichts unterschiedlicher Reaktionszustände und den damit korrespondierenden Farbwerten durch ein Kamerasystem in Smart Devices erfasst und mittels einer komplexen Serveranwendung referenziert werden.

Gegenwärtig sind leistungsfähige Smartphones weit verbreitet. Die darin verbauten Kamerasysteme weisen bereits einen sehr hohen technischen Standard auf (Gabbai, 2015). So sind Rauschverhalten, Farbtintensität und Dynamikumfang bereits sensor- und softwareseitig hinreichend kompensiert. Zusätzlich ist die Vernetzung der Endgeräte mit dem Internet weitgehend gewährleistet, sodass ein Informationsaustausch, eine serverseitige Auswertung und Datensicherung im Kontext des Internet der Dinge möglich wird. Entsprechend eignen sich Smartphones, für die Quantifizierbarkeit eines intelligenten Codes, auf der Grundlage von sensorischen Farben.

## 1.1 Zielsetzung

Das übergeordnete Ziel dieser Dissertation war es, Instrumente zu entwickeln, mit denen die Auswirkungen von Umwelteinflüssen auf ein Objekt kontrolliert, erfasst und validiert werden können. Es gilt herauszufinden, wie gedruckte sensorische Farben auf Umwelteinflüsse reagieren und wie Farbinformationen zu einer validen Aussage über die Umwelthistorie umgesetzt werden können. So können diese Instrumente als gesamtes System zur Veranschaulichung der jeweiligen Funktionen von sensorischen Farben sowie von anderen Wissenschaftlern für weitere Untersuchungen an ihren funktionellen Materialien genutzt werden.

Die Komplexität dieser Dissertation erfordert es, das übergeordnete Ziel in einzelne Teilziele mit möglichst geringer Interdependenz zu strukturieren und darauf aufbauend adäquate Teillösungen zu erforschen und zu entwickeln. Im Folgenden werden die jeweiligen Teilziele im Zusammenhang mit ihrer jeweiligen Bedeutung und Auswirkung auf das Gesamtsystem vorgestellt:

### 1) ENTWURF EINES GESAMTSYSTEMS

In diesem Unterziel wird die Ausgangslage mit ihren jeweiligen Hintergrundinformationen und den einhergehenden Herausforderungen für ein zu entwerfendes Gesamtsystem, das aus dem *Kapitel 1.2 Ausgangssituation* entnommen werden kann, vorgestellt. Zudem wird der konzeptionelle Architekturentwurf unter dem *Kapitel 1.3 Konzept* vorgestellt, in dem die zu entwickelnden Komponenten und ihre Kohärenz für eine technische Realisierbarkeit dieses Gesamtsystems aufgezeigt werden.

### 2) UNTERSUCHUNG UND ENTWICKLUNG SENSORISCHER FARBEN

Die Untersuchung und Entwicklung geeigneter sensorischer Farben, die unter dem Einfluss eines externen Stimulus ihre Farbe irreversibel von einem farblichen Zustand in einen anderen ändern und in diesem unumkehrbar verbleiben, sind Fokus dieses Unterziels. So werden geeignete sensorische Farben und ihre chemischen Mechanismen identifiziert, entwickelt und untersucht. Das *Kapitel 2.1 Sensorische Farben* behandelt den technischen sowie wissenschaftlichen Stand der Forschung. Anschließend werden die notwendigen Grundlagen im *Kapitel 4.1 Grundlagen sensorischer Farben* vermittelt und das physikalische Farbwechselverhalten in *Kapitel 3.4 Farbmétrische Untersuchung und Datenerhebungen* methodisch erläutert. Die Ergebnisse der Untersuchungen werden in *Kapitel 4.1 Grundlagen sensorischer Farben* vorgestellt.

### 3) ENTWICKLUNG UND UNTERSUCHUNG SENSORISCHER TINTEN FÜR DEN INKJETDRUCK

Die zuvor identifizierten und entwickelten sensorischen Farben werden hinsichtlich ihrer Eignung als Farbstoffadditiv in einer Tintenmatrix untersucht und für ihr Anwendungsfeld als Inkjettintente modifiziert. Hierbei liegt der Fokus des Unterziels bei den jeweiligen Methoden des *Kapitels 3.2 Bestimmung der Tinten Anforderungen*. Diese dienen der Analyse notwendiger drucktechnischer Parameter, die auf die Inkjetfähigkeit der sensorischen Tinten angepasst wird. Die Ergebnisse dieses Entwicklungsprozesses werden in *Kapitel 4.7 Entwicklung* vermittelt.

4) UNTERSUCHUNG INTELLIGENTER CODES

Die Identifizierung und Untersuchung eines geeigneten intelligenten Codes zur Speicherung dynamischer sowie statischer Inhalte ist Fokus dieses Unterziels. Im *Kapitel 2* wird zunächst die aktuelle *Stand der Forschung* sowie gegenwärtige und vergangene wissenschaftliche und technische Veröffentlichungen vorgestellt. Diese werden kategorisiert, gebündelt und unter einer kritischen Perspektive betrachtet. Zudem werden verschiedene sensorische Farben, intelligente Codes und Smart Labels unter der Perspektive ihrer Eignung als intelligenter Code untersucht und erste existierende Ansätze intelligenter Codes mit ihren jeweiligen Vor- und Nachteilen näher untersucht. Die für die Entwicklung eines intelligenten Codes notwendigen Grundlagen werden im *5.1 Grundlagen von Code-Technologien* näher erläutert.

5) ENTWICKLUNG EINES INTELLIGENTEN CODES

Die identifizierten Komponenten für den Aufbau eines intelligenten Codes fügen sich in das *Kapitel 5 Aufbau und Funktion eines intelligenten Codes* ein. Hierin werden alle Funktionalitäten und Komponenten, die für die Entwicklung eines intelligenten Codes notwendig sind, behandelt und dargestellt. Aus den einzelnen Komponenten werden die notwendigen Kriterien einer programmier-technischen Umsetzung für eine serverseitige Leseapplikation synthetisiert.

6) DRUCKTECHNISCHE UMSETZUNG UND AUSGABE EINES INTELLIGENTEN CODES

Dieses Teilziel liegt in dem Vorgang der drucktechnischen Ausgabe des eigens entwickelten intelligenten Codes. Hierbei wird das methodische Vorgehen in *Kapitel 3.3 Inkjet* näher erläutert. Der intelligente Code, der neben einem statischen Bereich auch unterschiedliche Sensoren im dynamischen Bereich beinhaltet, wird entsprechend des Ausgabeformats für den Inkjetdruck angepasst und ist Teil des *Kapitels 5.4 Ausgabe des intelligenten Codes*.

7) ERKENNUNGSPROZESS EINES INTELLIGENTEN CODES

In diesem Teilziel stehen die Erkennung und Erfassung des eigens entwickelten intelligenten Codes im Vordergrund. So werden die zuvor synthetisierten programmiertechnischen Kriterien in diesem Teilziel umgesetzt. Die hierzu notwendigen Grundlagen werden in *Kapitel 6.1 Grundlagen der Bildverarbeitung und Computer Vision* vermittelt. Das *Kapitel 6.2 Erkennungs-, Analyse- und Auswertungsprozess* greift auf Algorithmen für eine nachfolgende Mustererkennung zu, mithilfe dieser der intelligente Code identifiziert, ausgeschnitten und einer weiteren Programmroutine zur Separierung übergeben wird.

8) BESEITIGUNG VON STÖRPARAMETERN

Unterschiedliche Algorithmen assistieren bei der Beseitigung von möglichen Störfaktoren, die einen Einfluss auf die spätere Analyse des intelligenten Codes haben. Das *Kapitel 3.7.1 Geräteneutrale Auswertung* befasst sich mit den möglichen Störparametern und dem methodischen Vorgehen bei der Beseitigung dieser. Die für diesen Zweck entwickelten Algorithmen sind Bestandteil des *Kapitels 6.6 Beseitigung von Störparametern*.



9) SEPARIERUNG DES INTELLIGENTEN CODES

Der intelligente Code wird nach dem Beseitigen der Störparameter in seine Bestandteile separiert. Dieser Vorgang dient der nachfolgenden Auswertung der Codebestandteile, die aus einem dynamischen und einem statischen Bereich bestehen. Die hierzu notwendigen programmiertechnischen Schritte werden in *Kapitel 6.4 Separierung des intelligenten Codes* näher erläutert.

10) ANALYSE UND AUSWERTUNG SENSORISCHER FARBINFORMATIONEN

Die Farbinformationen der zuvor separierten dynamischen Bereiche werden in diesem Teilziel berechnet und ausgewertet. Dieser Prozess wird in *Kapitel 6.5 Auswerteeinheit* näher behandelt. So werden die Farbinformationen verwendet, um Aussagen über den Zustand der jeweiligen sensorischen Farben zu treffen.

11) KOMMUNIKATIONS- UND INTERNETTECHNOLOGIEN

In diesem Teilziel wird das methodische Vorgehen für die benötigten Soft- und Hardwarekomponenten in *Kapitel 3.8 Soft- und Hardwarearchitektur* näher erläutert. Die notwendigen Grundlagen und die sicherheitstechnischen Anforderungen werden in *Kapitel 6.7 Kommunikationstechnologien* näher behandelt. Die benötigte Hardwarearchitektur für die Realisierung einer serverseitigen Softwarearchitektur wird in *Kapitel 6.8* vertiefend erläutert.

12) FUNKTIONSTESTS UND IHRE EVALUATION

Nach einer abschließenden Realisierung des Gesamtsystems mit allen Komponenten und Strukturen werden die notwendigen Schritte für eine Validierung vorgestellt. Hierbei werden die damit einhergehenden Anforderungen, Grenzen und Risiken dieses Gesamtsystems im *Kapitel 7 Validierung* näher erläutert und kritisch betrachtet.

## 1.2 Ausgangssituation

Maschinenlesbare Codes wie bspw. DataMatrix Codes, QR-Codes oder Aztec-Codes haben sich bereits aufgrund ihrer binären Inhalte mit integrierter Fehlerkorrektur, ihrer hohen optischen Speicherdichte sowie schnellen und effizienten Lesbarkeit in der automatisierten Produktidentifikation und Datenerfassung etabliert. Der Einsatzbereich erstreckt sich von der Materialwirtschaft, Produktion, Logistik, bis hin zu weiteren gesonderten Einsatzgebieten. So finden sich erste Ansätze, die speziell zugeschnittene Codes durch einzelne sensorische Farben versuchen zu ergänzen, um neben statischen Informationen auch dynamische Informationen über häufig nur einen externen Einflussfaktor zu kommunizieren. Diese Ergänzung erweist sich jedoch als schwierig, da:

- Die binär codierten Daten in einem Code mittels einer optischen Leseinheit mit entsprechend speziell zugeschnittener Leseapplikation ermittelt werden, die bei einer strukturellen Veränderung des Codes eine Anpassung der Leseapplikation voraussetzt.
- Die Fehlerkorrektur, die integrativer Bestandteil eines Codes wie bspw. eines QR-Codes ist, ist bei einer Ergänzung des Codes um mehrere dynamische Bereiche zu deaktivieren, da andernfalls dynamische Bereiche irrtümlich korrigiert und somit beseitigt werden.

Neben den informationstechnischen Aspekten sind ebenfalls Anforderungen an die sensorischen Farben zu setzen. Die im dynamischen Bereich eingesetzten sensorischen Farben werden vermehrt für den Flexo- und Siebdruck hergestellt und häufig mit eben diesem Druckverfahren appliziert.

- Nachteilig anzusehen sind in diesem Druckverfahren neben der aufwendigen Datenaufbereitung, die relativ geringen Ausstoßzeiten, eine kürzere Durchlaufzeit und die Formherstellung, da für jede dieser Farben eine Form hergestellt werden muss. Zudem ist die erschwerte und eingeschränkte Personalisierbarkeit der Codes in eben diesem Druckverfahren zu kritisieren.

Entgegengesetzt verhält es sich beim kontaktlosen Inkjetdruckverfahren. Für dieses Druckverfahren finden sich kaum geeignete sensorische Tinten, da folgende Defizite aufzuzeigen sind:

- Die Tinten sind häufig reversible und sind nach einem entsprechenden Stimulus farblich umkehrbar. Diese sind somit als Nachweis für einen externen Einflussfaktor ungeeignet.
- Irreversible sensorische Farben, die nach einem externen Stimulus unumkehrbar in ihrem Zustand verbleiben, stehen kaum für das Inkjetdruckverfahren zur Verfügung und müssen speziell rezeptiert werden.

Dahingehend finden sich, neben den um die dynamischen Bereiche ergänzten Codes, eine Vielzahl an qualitativen Labels zur Überwachung des Produktzustands. So geben diese qualitativen Labels häufig nur eindimensionale Aussagen über ausschließlich einen Einflussfaktor, ignorieren den Gesamtzustand der Waren und die qualitativen Labels lassen sich häufig nicht maschinell auslesen. Informationen, die zu Zwecken einer Auswertung und Optimierung der Lieferkette genutzt werden können, werden häufig vernachlässigt. Ein bekanntes Beispiel dafür sind Time-Temperature-Indikatoren (TTI), die aufgebracht auf eine Lebensmittelverpackung, die Unterbrechung einer Kühlkette registrieren und anzeigen. Neben der Temperatur nehmen jedoch auch weitere äußere Faktoren Einfluss auf die Qualität eines Lebensmittels:

- So können bei beschädigter Verpackung, trotz intakter Kühlkette weitere Veränderungen wie bspw. bakterielle Kontaminationen am Produkt selbst entstehen.
- Empfindliche Produkte können während des Transports durch Stöße oder falsche Lagerung in ihrer Struktur beschädigt werden. Feuchtigkeit kann in das Packgut eindringen oder UV-Licht kann die Optik des Lebensmittels oder der Ware verändern.

Eine weitere Möglichkeit der Zustandserfassung kann über elektronische Sensoren realisiert werden, die sich durch präzisere Messverfahren und aussagekräftigere Ergebnisse auszeichnen. Hierbei sind jedoch einige Faktoren aufzuführen, die als nachteilig aufzuzeigen sind:

- Elektronische Sensorelemente benötigen eine Auswerteeinheit, die das elektronische Ausgangssignal in ein Sensor-Ausgangssignal umwandelt. Hierzu ist eine konstante Energieversorgung notwendig, da bei einer Langzeitmessung ansonsten eine vorzeitige Beendigung auftreten kann.
- Des Weiteren sind die Herstellungskosten von konventionellen Sensoren höher und kleinere Abmessungen häufig nicht möglich.

Für die Erfassung und Analyse eines intelligenten Codes spielt die eingesetzte Plattform eine wesentliche Rolle, dessen Bereitstellung und Verfügbarkeit ebenfalls ein relevanter Faktor für die barrierefreie Nutzbarkeit ist. Diese Verfügbarkeit kann eingeschränkt werden durch:

- Die eingeschränkte Bereitstellung von plattformabhängigen Applikationen (Apps), wie die zu diesem Zeitpunkt verbreiteten mobilen Betriebssysteme Android und IOS. Weitere Betriebssysteme werden häufig bei der App-Entwicklung nicht berücksichtigt, sodass eine Einschränkung der Nutzbarkeit bestehen kann.
- Das häufige Vernachlässigen weiterer Smart Devices wie Laptops, smarten Brillen, und anderen Endgeräten, die eine Kamera aufweisen. So würde eine plattformunabhängig eingesetzte Software auf den Smart Devices die geräteunabhängige Erfassung und Analyse von intelligenten Codes erleichtern.

Im Laufe der Gerätegenerationen haben sich auch die Smart Device Kameras weiterentwickelt. Ihre Verfügbarkeit hat sich gesteigert und die Qualitätsunterschiede der Endgeräte haben sich verringert. Dahingehend fokussierten sich viele große Smartphone-Hersteller aufgrund dieses Qualitätskriteriums ausgehend von den Nutzern auf Display- und Kameratechnologien und brachten immer qualitativere und leistungsstärkere Geräte auf den Markt. Dennoch sind ältere Smartphone Typen mit unzureichender Kameraauflösung und -qualität in Verwendung, die für eine Erfassung und Analyse der intelligenten Codes unzureichend sind.

### 1.3 Konzept

Im Folgenden wird ein konzeptioneller Architekturentwurf (Abbildung 1) vorgestellt, dessen technische Realisierbarkeit durch die Umsetzung eines prototypischen intelligenten Codes auf der Grundlage sensorischer Farben realisiert wird. Die zuvor vorgestellten Unterziele in *Kapitel 1.1 Zielsetzung* vervollständigen die wesentlichen Bausteine des hier behandelten Konzepts.

Die in diesem Konzept (Bilgin & Backhaus, 2017b) vorgestellten sensorischen Tinten basieren auf eigens entwickelten sensorischen Farben, die speziell für das Inkjetdruckverfahren entwickelt wurden. Die Zusammensetzung, Funktionalität und die Inkjetfähigkeit der sensorischen Farben ist hierbei ein Forschungsschwerpunkt dieser Arbeit. Diese sensorischen Tinten werden wie im Konzept vorgesehen im piezoelektrischen Inkjet, einem kontaktlosen Druckverfahren, auf einem speziell ausgewählten Substrat verdruckt. Das Substrat dient als Informationsträger und kann in Form eines Haftpapiers auf Waren oder Verpackungen aufgebracht werden. Die für das Verdrucken der Sensoren notwendige Codevorlage basiert neben dem integrierten statischen Bereich auf einer Kodierung, die dynamische Bereiche innerhalb des intelligenten Codes beinhaltet. So baut das Verdrucken der Codevorlage darauf auf, dass die farbigen Tintenkartuschen durch die drei entwickelten sensorischen Tinten für den Inkjetdruck ersetzt werden, wobei das Schwarz für das Verdrucken der statischen Bereiche nicht ersetzt wird. So wird nachfolgend der auf ein Haftpapier verdruckte intelligente Code mit allen Bestandteilen in seinen originalen unveränderten Zustand als Referenz serverseitig hinterlegt und dient dem nachträglichen Vergleich des sich veränderten Codes.

Die sensorischen Farben innerhalb des intelligenten Codes reagieren auf chemisch-physikalische Einflussfaktoren, wie z. B. aus der Umwelt mit einer Stimuli induzierten Farbveränderung. So ändert sich die Farbigekeit der sensorischen Farben mit dem jeweiligen Umwelteinfluss von einem farblichen Zustand in den anderen, sodass sich der intelligente Code während des Transports bei möglich vorkommenden Einflussfaktoren dynamisch mitverändert und diese Einflussfaktoren auf die Ware anzeigt.

Der intelligente Code kann dann mit einer hier vorgestellten serverseitigen Leseapplikation über einen bereits vorinstallierten Browser von einem Smartphone oder einem anderen internetfähigen Smart Device mit integrierter Kamera nach jedem Transportschritt erfasst, ausgewertet und in eine Datenbank eingetragen werden. Diese Leseapplikation bildet einen weiteren Forschungsschwerpunkt. Die eingetragenen Daten über den aktuellen Zustand des intelligenten Codes werden über ein spezielles Verfahren mit den Referenzdaten verglichen und die jeweilige Abweichung vor und nach der Kontamination in Form eines aufgearbeiteten Ergebnisberichts an das Smart Device zurückgesandt. Somit kann die Transporthistorie der Ware nach jedem Erfassungsvorgang im Laufe des Transports zurückverfolgt werden. Die serverseitige Leseapplikation dient zum einen der Dekodierung des statischen Bereichs, welcher Produktinformationen oder andere Informationen beinhaltet. Zum anderen dient sie der Auswertung und Analyse des dynamischen Bereichs in dem sich die sensorischen Farben befinden. Des Weiteren verortet sich neben einem Farbkorrekturalgorithmus ein Colormanagementsystem ebenfalls auf dem Server, welches für eine einheitliche farbverbindliche Erfassung durch die Smart Device Kameras mit entsprechender ICC Profilierung zuständig ist, da die Bilderfassung verschiedener Endgeräte in ihrer Farbgebung stark variieren kann. Zudem verorten sich unterschiedlich entwickelte Algorithmen, die sich mit Störparametern beschäftigen, die die Analyse des intelligenten Codes behindern können.

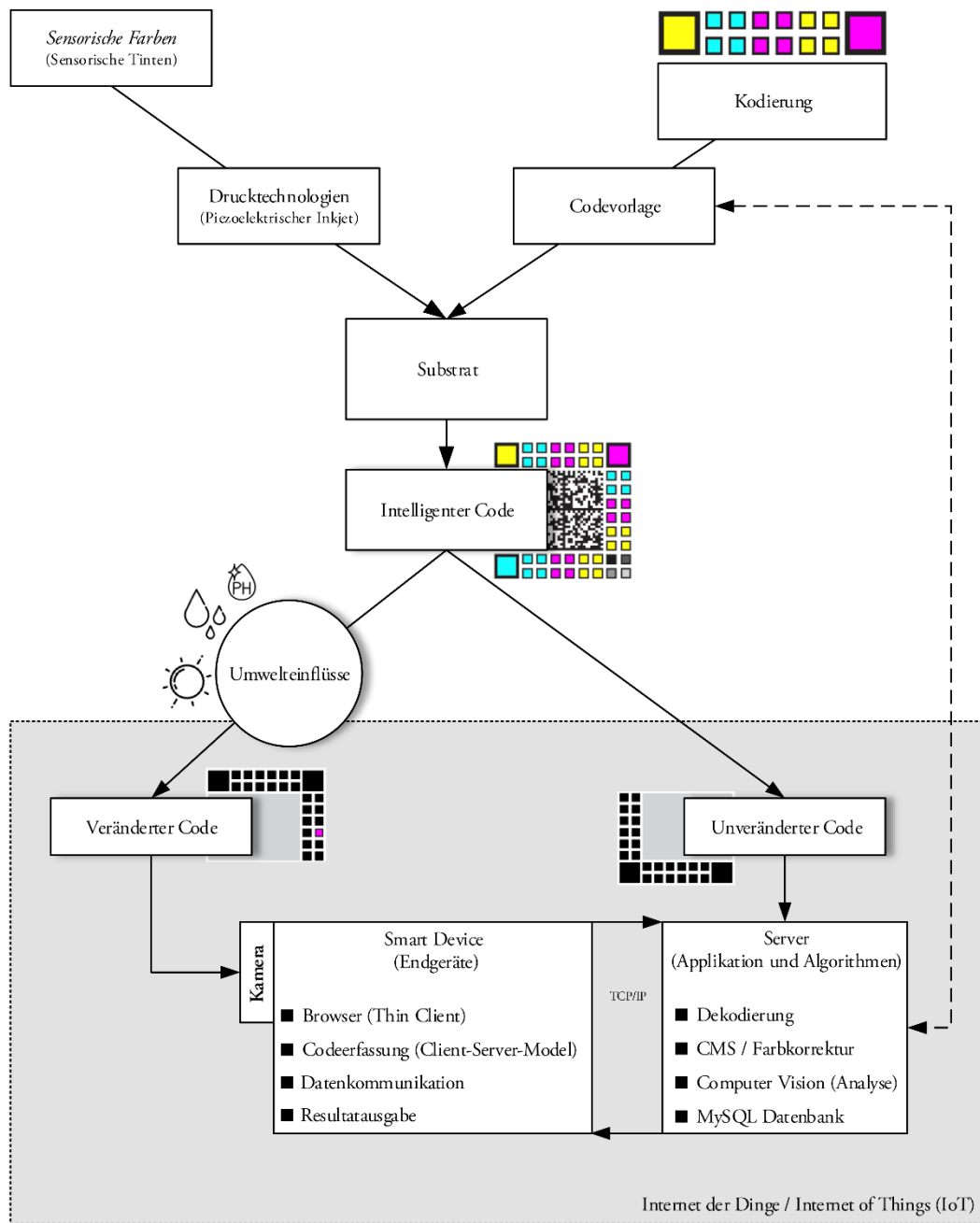


Abbildung 1 Konzeptioneller Architektorentwurf (Bilgin & Backhaus, 2017a)

## 1.4 Aufbau

Nachfolgend werden die zum Forschungsthema publizierten Werke und wissenschaftlichen Veröffentlichungen im *Kapitel 2 Stand der Forschung* näher behandelt und thematisch in das *Kapitel 2.2 Intelligente Code-Technologien* und das *Kapitel 2.1 Sensorische Farben* unterteilt.

In *Kapitel 3* werden die für die Untersuchung verwendeten Materialien und Methoden vorgestellt. Zunächst werden die Methoden zur Bestimmung der Tintenanforderungen, aber auch die der farbmetrischen Untersuchung und Datenerhebung vorgestellt. Darauf folgend werden außerdem das verwendete Inkjetsystem, die eingesetzten programmiertechnischen Methoden und unter anderem das methodische Vorgehen beim Aufbau einer Soft- und Hardwarearchitektur beschrieben.

Das *Kapitel 4* behandelt die *sensorischen Tinten*, die in *4.6 Untersuchung sensorischer Farben* und *4.7 Entwicklung* unterteilt werden. Hierin werden die entwickelten sensorischen Farben und die entsprechend für das Drucksystem modifizierten sensorischen Tinten vorgestellt.

Die Grundlagen der Code-Technologien werden im *Kapitel 5 Aufbau und Funktion eines intelligenten Codes* behandelt. Das *Kapitel* setzt sich zum einen aus dem *Unterkapitel 5.2 Kriterien für einen intelligenten Code* und zum anderen aus dem *Unterkapitel 5.3 Entwicklung und Aufbau eines intelligenten Codes* zusammen.

Das *Kapitel 6 Softwarearchitektur* behandelt alle notwendigen Grundlagen für die Bilderfassung und Bildanalyse. Im *Unterkapitel 6.1 Grundlagen der Bildverarbeitung und Computer Vision* werden die programmiertechnischen Grundlagen für eine nachfolgende Bilderfassung mit dem Einsatz des maschinellen Sehens (Computer Vision) erarbeitet. Weitergehend wird unter *Kapitel 6.4 Separierung des intelligenten Codes* die Identifizierung von Teilssegmenten für das nachfolgende *Kapitel 6.5 Auswerteeinheit* behandelt. In *Kapitel 6.6 Beseitigung von Störparametern* werden Farbanpassungen, Helligkeits- und Kontrastanpassungen und die Implementierung eines Colormanagementsystems ausgeführt.

*Kapitel 6.7 Kommunikationstechnologien*, setzt sich mit den notwendigen Grundlagen der Internettechnologien, des Internet der Dinge sowie der Datensicherheit und Verschlüsselung auseinander. Die entsprechende *Hardwarearchitektur* wird im *Kapitel 6.8* behandelt. Im *Kapitel 7* erfolgt die *Validierung* des Gesamtsystems. Schließlich zeigt *Kapitel 8* einen *Ausblick* auf die Relevanz der Ergebnisse dieser Arbeit und weitere Forschungsbedarfe.

## Kapitel 2

# Stand der Forschung

Der folgende wissenschaftliche Diskurs stellt nicht nur die gegenwärtigen Erkenntnisse im Forschungsstand dieser Dissertation dar, sondern kategorisiert, bündelt und betrachtet auch die diesbezüglich relevanten aktuellen wissenschaftlichen und technischen Veröffentlichungen unter einer kritischen Perspektive. Hierbei gilt es, den Forschungsstand detailliert und so vollständig wie möglich wiederzugeben und die Arbeit in den Forschungskontext einzuordnen. Damit kommt die im *Kapitel 1.1* festgelegte *Zielsetzung* zum Tragen.

Dieser Forschungsstand setzt sich mit dem wissenschaftlichen sowie technischen Kontext der dreigeteilten Themenbereiche der *sensorischen Farben*, der *intelligenten Code-Technologien* und der *Smart Labels* auseinander und stellt die im *Kapitel 1.2* genannten kritischen Aspekte und die gegenwärtigen Forschungsstände in eine wechselseitige Beziehung. Diese Dreiteilung erfolgt auf der Grundlage der konzeptionellen Ausarbeitung, welche die Forschungsschwerpunkte dieser Dissertation darstellen.

Beginnend werden im *Unterkapitel 2.1 Sensorische Farben* die wesentlichen Veröffentlichungen und Forschungsansätze im Zusammenhang mit den sensorischen Farben, hier insbesondere unter dem Fokus ihrer Eignung für irreversible sensorische Tinten, vorgestellt. Zudem werden Veröffentlichungen, die auf eine Inkjet-Druckfähigkeit hinweisen, hervorgehoben. Darauf folgend werden im *Unterkapitel 2.2 Intelligente Code-Technologien* alle wesentlichen Konzepte, Veröffentlichungen und Forschungsansätze, insbesondere im Zusammenhang mit den intelligenten Codes, vorgestellt. So werden ausgewählte intelligente Codes, welche auf ein Lesegerät bzw. auf eine Leseapplikation und das Inkjetdruckverfahren zurückgreifen, näher behandelt. Abschließend werden im *Unterkapitel 2.3 Smart Labels* alle wesentlichen Konzepte und Veröffentlichungen zum Themenbereich der bereits entwickelten und vermarkteten Smart Labels vorgestellt und kritisch betrachtet. Hierzu werden ausgewählte Smart Labels, welche auf ein Messgerät zurückgreifen, näher vorgestellt. In allen drei Themenbereichen werden zudem die jeweiligen Funktionen und Eigenschaften ausgewählter besonders charakteristischer Ergebnisse näher erörtert.

## 2.1 Sensorische Farben

Bestimmte funktionelle Materialien werden durch einen Stimulus aus der unmittelbaren Umwelt in ihrer chemischen Struktur verändert, wodurch eine Farbveränderung induziert wird. Bei dem Stimulus kann es sich um Licht einer spezifischen Wellenlänge (*Photochrome*), Wasser bzw. Feuchtigkeit (*Hydrochrome*), Temperatur (*Thermochrome*), Druck (*Piezochrome*) oder jedoch auch bei (bio-) chemischen Einflüssen, um den pH-Wert (*Halochrome*), chemischer Stoffe wie Gase (*Gasochrome*) und Ionen oder biologische Moleküle handeln. Das Suffix „-chrome“ wird mit unterschiedlichen funktionellen Stoffnamen verbunden. Innerhalb der vorliegenden Arbeit werden diese verschiedenen „-chromen“ Materialien – vom griech. χρώμα (chroma) für Farbe (Gottwein, 2020) – durch den Terminus *sensorische Farben* ersetzt, um einen einheitlichen Oberbegriff für die Vielzahl der verschiedenen funktionellen Materialien zu verwenden. Sensorische Farben, die für das Inkjetdruckverfahren als druckfähige Tinte angepasst

werden, werden nachfolgend innerhalb dieser Dissertation als *sensorische Tinten* bezeichnet, wobei diese nach dem Verdrucken erneut als sensorische Farben bezeichnet werden. Die Verwendung als sensorische Tinten setzt voraus, dass diese nach einer induzierten Farbreaktion unumkehrbar in ihrem neuen farblichen Zustand verbleiben (*irreversibel*) und sich nicht in ihren ursprünglichen farblichen Zustand zurückführen lassen (*reversibel*). Aus diesem Grund liegt der Fokus der nachfolgenden Recherche auf irreversiblen sensorischen Farben. Dabei wird die Thematik der „Liquid crystals“ aufgrund ihrer Komplexität und ihren physikalischen Eigenschaften ausgeklammert, da diese für die vorliegende Arbeit keine Relevanz haben.

Im Folgenden werden neben wissenschaftlichen Veröffentlichungen und Patenten auch Entwicklungen von Unternehmen vorgestellt, die sich thematisch mit sensorischen Farben auseinandergesetzt haben. Ein exemplarisches Beispiel hierfür ist das Drittmittelprojekt (BAY IGF–1311-0004) der Papiertechnischen Stiftung, welches sich thematisch mit irreversiblen sensorischen Farben auseinandersetzte. Das abgeschlossene Forschungsvorhaben mit dem Titel „Entwicklung eines innovativen Manipulationskennzeichens für Hot-Melt-Verklebungen in pharmazeutischen Faltschachteln“ hatte das Ziel der „Verhinderung von unberechtigter und nicht sichtbarer Manipulation des Inhaltes von Pharma-Verpackungen“. So wurden „verschiedene Farbstoffsysteme identifiziert und detailliert hinsichtlich ihrer Eignung als Manipulationsindikator für Hot-Melt-Verklebungen untersucht“ (Eiblmeier & Metz, 2015, S. 2). Hierzu wurden drei verschiedene irreversible thermochrome Pigmente (TMC Hallcrest „Kromagen“ Reihe) eingeholt. Diese wurden mittels Inkjet-, Offset-, Flexodruck- und einem Sprühverfahren auf ihre Auftragseigenschaften untersucht. Die erhöhte Partikelgrößenverteilung in den Materialien führten, trotz eines zusätzlichen Verdünnungsprozesses, zum Verstopfen der Düse und somit zum Ausscheiden für das Inkjetdruckverfahren. Das Flexodruckverfahren war ebenfalls mäßig geeignet und bei diesem konnte lediglich eine mäßige Auftragsmenge realisiert werden. Als geeignetes Verfahren wurde der Auftrag durch das Versprühen identifiziert.

In Tabelle 3 sind verschiedene irreversible sensorische Farben mit Angaben zu Autoren / Unternehmen, Erscheinungsjahr, Herkunftsland, Einsatzzweck (Art), farblichem Verhalten, verwendetem Druckverfahren und ggf. Patentierung dargestellt. Außerdem sind irreversible sensorische Farben aufgrund ihrer Inkjetdruckfähigkeit grau hervorgehoben. Auf exemplarische Beispiele wird aufgrund der chemischen Komplexität der Materialien und der zum Verständnis notwendigen vorauszusetzenden Grundlagen verzichtet. Anhand der Tabelle ist zu erkennen, dass vermehrt sensorische Farben für den Sieb- und Flexodruck angeboten oder erforscht werden. Aktuell finden sich einige Veröffentlichungen, die sich mit sensorischen Farben auseinandersetzten und an der Entwicklung dieser beschäftigen (Tabelle 3). Dahingehend ist es relativ einfach, sensorische Farben für den Sieb- oder Flexodruck nach einer Suchmaschinenanfrage zu erwerben. Vermehrt sind thermo- und photochrome sensorische Farben aufzufinden. Entgegengesetzt verhält es sich mit sensorischen Farben für den Inkjetdruck. Wie auch zuvor in *Kapitel 1.2* beschrieben, sind diese aufgrund der vergleichsweise aufwendigeren Zusammensetzung von Inkjettinten kaum (oder nur auf Anfrage entwickelt) oder nicht für den gesuchten Einsatzbereich bzw. die Funktionalität zu erwerben oder gar nicht erwerbbar. Aus diesem Grund wurde innerhalb dieser Dissertation die eigene Entwicklung von drei verschiedenen sensorischen Tinten auf Basis sensorischer Farben (Photochrome, Hydrochrome, Halochrome) angestrebt (*Kapitel 4*), welche eine Irreversibilität aufweisen und sich für das verwendete kontaktlose Inkjetdruckverfahren eignen.



Autoren / Unternehmen	Jahr	Land	Art	Farbe	Druckverfahren	Patent
Atlanta Chemical Engin.	2020	USA	TC	Versch.	SD,SP	Nein*
Bielek und Barker	2006	USA	TC	Versch.	n. a.	Ja
Branda et al.	2006	USA	PC	Trans.-Blau	n. a.	Ja
Brettle und Trundle	1989	GBR	PC	Trans.-Farb.	SD	Ja
CTI Inks	2020	USA	PC	Grün-Braun	SD,FD,OD	Ja
Elhard und Heggs	2004	USA	HC	Weiß-Farb.	n. a.	Ja
Fischer et al.	1994	DEU	PC	Trans.-Blau	OD	Ja
Fischer et al.	1993	DEU	PC	Trans.-Farb.	OD	Ja
FX Pigments	2020	IND	TC,PI	Versch.	SD,FD	Nein**
Fyvie et al.	2011	USA	TC	Versch.	SD,ID	Ja
Hajime und Kang	2014	USA	HC	Trans.-Farb.	n. a.	Ja
Hanke	2004	DEU	TC	Gelb-Orange	Alle*	Ja
Hirokawa et al.	2011	JAP	PC	Trans.-Blau	Alle*	Ja
Insilico	2020	KOR	PC	Trans.-Farb.	SD, FD, OD,TD	Nein*
Kanakanatt et al.	2012	USA	HC	Trans.-Farb.	SD,FD	Ja
Kaplan et al.	2017	TUR	TC	Mag.-Blau	n. a.	Ja
Kim et al.	2015	KOR	HC	Blau-Rot	ID	Ja
Kurihara et al.	2012	JAP	TC	Farb.-Trans.	ID	Ja
Kwan	2013	USA	TC	Farb.-Trans.	ID*	Ja
Lawrence Industries	2020	GBR	TC	Versch.	SD,FD	Nein**
LCR Hallcrest	2020	USA	TC	Trans.-Farb.	SD,FD	Nein**
LCR Hallcrest	2020	USA	HC	Schw.-Trans.	SD,FD	Nein**
Lent und Elue	1988	USA	TC	Versch.	ID	Ja
Lucht und Euler	2009	USA	TC	Gelb-Lila	n. a.	Ja
Malic et al.	2009	AUS	PC	Trans.-Farb.	n. a.	Ja
Matsui	2020	USA	TC	Versch.	SD,FD	Ja
Miyata et al.	2003	JAP	TC	Versch.	SD,OD,TD	Ja
Nanomatrixsecure	2020	HKG	TC	Trans.-Farb.	SD,FD	Nein**
NCC New Prismatic Ent.	2020	TWN	TC	Versch.	SD,FD	Nein**
Newcolorchem	2020	CHN	TC	Trans.-Farb.	SD	Nein**
Nguyen et al. et al.	2018	KOR,VIET	TC	Lila-Weiß	n. a.	Nein**
NiGK Corporation	2020	JAP	RS,TC	Versch.	Alle*	Ja
OliKrom	2020	FRA	X*	Versch.	SD,OD	Ja
Pietras	2017	POL	PC	Trans.-Farb.	SD,FD,OD,TD	Ja
Pudewills	2009	DEU	PC	Lila-Gelb	ID	Nein**
Riethmüller et al.	2016	DEU	TC	Versch.	SD,OD,TD,ID	Ja
SFXC	2020	GBR	TC	Schw.	SD,FD	Nein**
Sintal Chemie	2020	DEU	TC	Versch.	n. a.	Nein**
Skyrad	2020	ISR	PC	Versch.	n. a.	Ja
Skyrad	2020	ISR	TC	Versch.	SD,OD	Ja
Smarol	2020	CHN	TC	Trans.-Farb.	SD,FD	Nein**
Smarol	2020	CHN	PI	Weiß-Schw.	SD,FD	Nein**
Tanimoto	1992	JAP	HC	Trans.-Blau	FD,TD	Ja
Tateoka et al.	1990	JAP	PC	Trans.-Farb.	SD	Ja
Wang et al.	2015	USA	TC	Versch.	n. a.	Ja
Winskowicz	1998	USA	HC	Trans.-Blau	n. a.	Ja
Yoon et al.	2011	KOR	TC	Blau-Rot	ID	Nein**

Tabelle 3 Irreversible Sensorische Farben und die Hervorhebung ihrer Inkjetfähigkeit (grau)

Legende:

X\*: TC, PC, SC, GC, PiC; \*laut Patent geeignet für alle Druckverfahren; \*\*kein zugehöriges Patent gefunden

Versch. (Verschieden), Trans. (Transparent), Schw. (Schwarz), Farb. (Farbig), n. a. (nicht angegeben)

SD (Siebdruck), SP (Sprühverfahren), FD (Flexodruck), TD (Tiefdruck), ID (Inkjetdruck)

TC (Thermochrome), PC (Photochrome), HC (Hydrochrome), RS (Radiation Sensor), PI (Piezochrome)

## 2.2 Intelligente Code-Technologien

Neben den klassischen eindimensionalen Codes (1D), die Basisdaten in Form eines Barcodes oder auch eines Strichcodes beinhalten und zur Identifizierung eines Produktes dienen, gibt es darüber hinaus zweidimensionale Codes (2D) in Form von Composite-, Punkt- oder Matrix-Codes. Letztere weisen eine höhere Speicherkapazität, fallweise Algorithmen zur automatischen Fehlerkorrektur auf und können individuelle Informationen, z. B. Namen und Anschriften, beinhalten. Darüber hinaus finden sich zudem dreidimensionale Codes (3D), die als dritte Dimension zudem farbige Elemente verwenden. Diese erhöhen zum einen die Speicherkapazität und zum anderen können sie in Form von dynamischen Bereichen die Funktionalität der Codes erweitern.

Intelligente Code-Technologien, die auch unter der Bezeichnung, *Intelligent Codes*, *Smart Codes* oder auch unter dem Sammelbegriff der *Smart Tags* aufzufinden sind, bestehen häufig aus klassischen maschinenlesbaren Code-Technologien, die innerhalb ihrer statischen Bereiche erweitert werden. Zumeist sind in einem solchen statischen Bereich binäre Daten verortet, die durch einzelne dynamische Zonen in Form von sensorischen Farben in das Gesamtsystem des vorher statischen Codes integriert werden. So fokussiert sich dieser Forschungsstand vielmehr auf *intelligente Codes*, die neben einem binär statischen Bereich, ebenso wie der für die vorliegende Arbeit entwickelte intelligente Code, auch mindestens einen dynamischen Bereich für sensorische Farben beinhalten. Diese sensorischen Farben zeigen die Einwirkungen äußerer Einflüsse auf ihren Zustand. Diese Einflüsse gelten ggf. auch für ihr Umfeld, z. B. ein Produkt, in dem sie sich farblich unumkehrbar ändern, wenn ein speziell definierter externer Reiz auf sie einwirkt.

Nachfolgend werden vertiefend alle wesentlichen wissenschaftlichen Konzepte und Veröffentlichungen, insbesondere im Zusammenhang mit denen der intelligenten Codes, vorgestellt. Das Hauptziel des EU-weiten Projekts „TagItSmart! Smart Tags driven service platform for enabling ecosystems of connected objects“ mit 15 Projektpartnern war es, eine Reihe von Werkzeugen und zugehörige Basistechnologien mit offenen Schnittstellen zu schaffen, die in eine Plattform integriert werden können. Durch den Einsatz funktioneller Tinten in Kombination mit digitalen und elektronischen Markern, welche in der Lage sein sollten ihre eigene Umgebung zu erfassen, sollten diese Marker mit kostengünstigen Abtastrfunktionen ausgestattet und mit jeweiligen Produkten verbunden werden. Diese Marker, die auch durch die Projektteilnehmer „Smart Tags“ (z. B. NFC-Tags) oder auch „FunCodes“ (erweiterte Symbolcodes) genannt werden, sollten in eine cloudbasierte Plattform integriert werden, um den Lebenszyklus von unterschiedlichen Waren lückenlos zu verfolgen. Kernelement sind Smartphones mit integrierten Kameras und integrierten NFC-Lesegeräten, die eine Brücke zwischen der Interaktion von Nutzern und Objekten bauen. Die Architektur der TagItSmart! Plattform basierte auf einem Gesamtsystem, das sich mit der Kodierung, Speicherung, Abfrage, dem Drucken und der Erfassung von Smart Tags oder FunCodes auseinandersetzt (H2020). Nachfolgend wird die Architektur des Konzepts dargestellt, das drei grundlegende Schichten aufweist:

- 1) Benutzer- und Entwicklerschicht (User / Developer level):  
In dieser Schicht befinden sich eine grafische Oberfläche (Frontend) zur Visualisierung, die Datenverwaltung, das Benutzermanagement und neben weiten Bereichen ein Software Development Kit (SDK) mit einer Sammlung an Programmierwerkzeugen und -bibliotheken für das Erstellen einer Android App.
- 2) Serviceschicht (Service level):  
Diese Schicht beschäftigt sich zum einen mit der Datenverarbeitung und der Verwaltung von Smart Tags und zum anderen mit der Kodierung, Erstellung und dem Druck von Smart Tags. So finden sich innerhalb dieser Schicht zudem Serviceausführungs- und Sicherheitsdienste, die bspw. den Zugang zur Plattform verwalten und den Benutzer authentifizieren. Zuletzt finden sich im Konzept unterschiedliche Datenbanken, welche die Daten verwalten.
- 3) Zugriffsschnittstelle für virtuelle Elemente (VE level):  
Diese letzte Schicht ist für den Zugriff der Sensoren, Lesegeräte (Scanner) und zum Drucken eines „Smart Tags“ oder „FunCodes“ zuständig.

Somit stellt das TagItSmart-Konzept eher einen serverbasierten Ansatz dar, der seinen Fokus als cloud-basierte Plattform auf das Verwalten von verschiedenen Werkzeugen und Dienstleistungen legt. Der Projektbericht erwähnt zudem Materialtests mit drei unterschiedlichen funktionellen Tinten (sowie eine vierte, in diesem Kontext nicht erwähnenswerte Tinte), um ihre Druckbarkeit und Eignung als „Fun-Code“ in Form von QR- und DataMatrix Codes zu untersuchen. Für die Untersuchung wurde auf unterschiedliche Druckverfahren, insbesondere Inkjet-, Sieb- und Flexodruck zurückgegriffen. Hierbei wurden reversible thermochrome und photochrome Tinten (LCR Hallcrest) sowie eine Tinte mit einem irreversiblen Sauerstoffindikator eingesetzt. Die für das Flexodruckverfahren formulierten schwarzen funktionellen Tinten erwiesen sich aufgrund zu großer Pigmente für das Inkjetdruckverfahren als ungeeignet. Im Gegensatz dazu ließ sich mit der reversiblen roten pigmentbasierten funktionellen Flexodruckfarbe nach zehnmalem Überdrucken annähernd eine Sättigung erzielen. Der Untersuchung einer Partikelgrößenverteilung der verwendeten Materialien wurde nicht nachgegangen. Zudem konnte kein brauchbares Ergebnis für die Eignung der sensorischen Tinten geliefert werden, die größtenteils aufgrund ihrer Reversibilität als Nachweis unbrauchbar waren. Das Gesamtsystem der Plattform zeigt einen ersten Ansatz, um verschiedenen Code-Technologien eine Managementplattform zu bieten, die jeweiligen Daten zu verwalten und zu strukturieren. Eine praktische Umsetzung in Bezug auf intelligente Codes, die ihren Inhalt dynamisch ändern, ist zu diesem Zeitpunkt jedoch immer noch nicht gegeben.

In der nachfolgenden Tabelle 4 werden intelligente Codes vorgestellt. Hierzu werden neben wissenschaftlichen Veröffentlichungen auch Patente vorgestellt, die intelligente Codes beinhalten. Dahingehend werden in der nachfolgenden Tabelle die Autoren, Herkunft (Land), Einsatzzweck (Art), Art des Codes, Funktion, Farbe (Farbveränderung), ihre Nutzung des Inkjetdruckverfahrens (Inkjet), die Notwendigkeit eines Lesegeräts oder Leseapplikation zum Erfassen und Lesen der Codes und das Vorhandensein eines Patents dargestellt.

Autoren	Land (ISO 3166)	Art	Code	Funk.	Farbe	Inkjet	Lesegerät	Patent
Aida et al.	JPN	TTI	Eigenentw.	Irrevers.	Trans.-Blau	Ja	Ja	Ja
Burklund et al. et al.	USA	BS	QR Code	Irrevers.	Trans.-Blau	Ja	Ja**	Nein***
Cameron	USA	TC	Barcode	Revers.	Weiß-Schw.	Nein	Nein	Ja
Chen, Y. et al.	USA	GS	QR Code	Irrevers.	Trans.-Farbig	Nein	Ja**	Nein***
Đurđević et al.	SRB,+	TC	QR Code	Revers.	Trans.-Schw.	Nein	Ja	Nein***
Ecker und Pretsch	DEU	TC	QR Code	Revers.	Farbig-Weiß	Nein	Nein	Ja
Edinger et al.	DNK	AI	QR Code	n. N.	Farbig	Ja	Nein	Nein***
Gao et al. et al.	CHN	PC	QR Code	Revers.	Trans.-Blau	Ja	Nein	Nein***
Gligoric et al.	SRB,+	TC*	QR Code,+	Revers.	Rot-Trans.	Ja	Ja	Ja
Haarer und Levy	DEU, ISR	FI	Barcode,+	Revers.	Schw.-Weiß	Nein	Ja	Ja
Heilmann et al.	FIN	HA,+	Barcode,+	Irrevers.	k. A.	Ja	Ja	Ja
Hyde und Pan	USA	TTI	QR Code	Revers.	Rot-Gelb	Nein	Nein	Ja
Khatri, Z. et al.	JPN,+	PC	QR Code	Revers.	Trans.-Rot	Nein	Nein	Nein***
Lee et al.	KOR	BS	QR Code,+	Irrevers.	Gelb-Violett	Nein	Ja	Ja
Lewis	USA	RS	Barcode	Revers.	Trans.-Schw.	Nein	Nein	Ja
Li et al.	CHN	TC	QR Code	Revers.	Trans.-Schw.	Nein	Nein	Nein***
Liu, S. et al.	CHN	TC	QR Code	Revers.	Schw.,Bl.,Tü.	Nein	Nein	Nein***
Kawasaki und Kohhei	JPN	TC	QR Code	Irrevers.	Rot-Violett	Ja	Nein	Ja
Meng et al.	CHN	pK	QR Code	Revers.	Trans.-Schw.	Nein	Nein	Nein***
Nicholas et al.	GBR	TTI	Barcode	Irrevers.	Trans.-Farbig	Ja	Nein	Ja
Norrby und Nygård	SWE	TTI	Barcode	Irrevers.	Trans.-Farbig	Nein	Nein	Ja
Petriashvili et al.	GEO, ITA	PC	QR Code	Revers.	Trans.-Violett	Nein	Nein	Nein***
Pretsch und Fritzsche	DEU	TTI	QR Code,+	Irrevers.	Materialdef.	Nein	Nein	Ja
Prusik et al. et al.	USA	TC	QR Code	Revers.	Weiß-Schw.	Ja	Ja	Ja
Ramalho et al.	POR	TC	QR Code	Revers.	Orange-Rot	Nein	Ja	Nein***
Ribi	USA	TC,+	Barcode,+	Irrevers.	Weiß-Blau	Ja*	Ja**	Ja
Soga et al. et al.	JPN	GS	QR Code	Irrevers.	Orange-Gelb	Ja	Nein	Nein***
Hashizume et al. et al.	JPN	TTI	QR Code,+	Irrevers.	Trans.-Farbig	Nein	Ja	Ja
Weston et al. et al.	AUS	BS	QR Code	Irrevers.	Trans.,Bl.,Ro.	Ja	Nein	Nein***
Wu et al.	USA	TC	QR Code	Revers.	Grau-Violett	Ja	Nein	Ja
Yang, Y. et al.	CHN	PC,+	QR Code	Revers.	Gelb-Blau	Nein	Nein	Nein***
Yang, W. et al.	CHN	TC	QR Code	Revers.	Trans.,Ro.,Vi.	Nein	Nein	Nein***
Yuan et al. et al.	USA	BS	QR Code	Irrevers.	Trans.-Braun	Ja	Nein	Nein***
Yoon et al.	KOR	TC	QR Code	Irrevers.	Bl.-Ro.-Gelb	Ja	Nein	Nein***

Tabelle 4 Veröffentlichungen und Patente Intelligenter Code-Technologien

Legende:

Funk. (Funktion), Ro. (Rot), Bl. (Blau), Vi. (Violett), Schw. (Schwarz), Tü. (Türkis), Trans. (Transparent), + (weitere genannt), Materialdef. (Materialdeformierung) BS (Bio Sensor), FI (Freshness Indicator), GS (Gas Sensor), TTI (Time-Temperature-Indicator), HA (Halochromic), PC (Photochrome), TC (Thermochromic), RS (Radiation Sensor), AI (Active Ingredient), pK (Photonische Kristalle), \* laut Angaben auch geeignet für den Inkjetdruck, \*\*keine eigens entwickelte App, \*\*\*kein zugehöriges Patent gefunden

Die jeweiligen Funktionen und Eigenschaften ausgewählter intelligenter Codes, die ähnlich dem Ziel dieser Arbeit, auf ein Lesegerät und den Angaben der Erfinder oder Autoren zufolge auf das Inkjetdruckverfahren zurückgreifen (grau markiert), werden im Folgenden kritisch betrachtet und näher behandelt:

#### 1. Aida et al.

Der intelligente Code von Aida et al. (2020) ist ein kohärentes TTI System. Es besteht aus einem dynamischen TTI und einem gesonderten Symbolcodebereich für statische Informationen (Abbildung 2). Genauer betrachtet besteht der TTI aus unterschiedlichen mehrschichtigen Materialien inklusive der drei unterschiedlichen Materialien zur Temperaturerfassung. Diese setzen sich laut Patentschrift aus einem „Leukofarbstoff“, einem „Farbentwicklungs“ und einem „Entfärbungsmittel“ zusammen. Die Temperaturerfassungsmaterialien weisen in unterschiedlichen Temperaturbereichen verschiedene Kristallisationsspitzen auf, die sich in einer Farbveränderung und somit in unterschiedlichen Farbdichten äußern, wobei sich das Material teilweise entfärbt. Hierbei kann der

Oberflächenfarbton (in  $L^*a^*b^*$ ) erfasst und mit einer Referenztafel abgeglichen werden, um Aussagen zu unterschiedlichen Temperaturbereichen zu treffen. Hierzu wird ein sogenanntes „quality control terminal“ (Lesegerät) eingesetzt, das neben den unterschiedlichen Symbolcodes auch den jeweiligen Farbwert der Indikatoren und die GPS Position (Global Positioning System) erfasst, an einen Server sendet und dem Benutzer einen Zwischenbericht auf dem integrierten Display ausgibt.

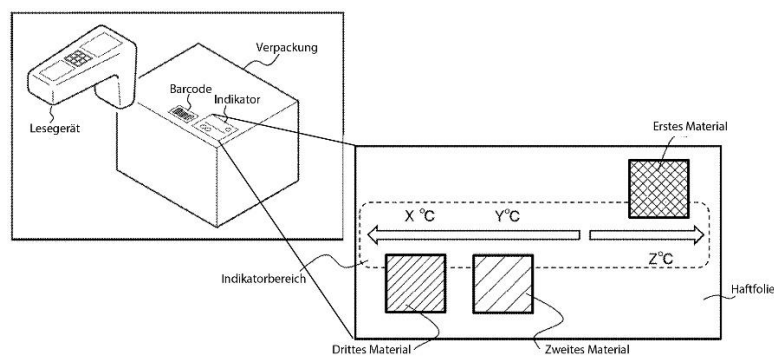


Abbildung 2 QCS (Quality Control System) (Aida et al., 2020)

Kritisch anzumerken ist, dass das Quality Control System nicht offen und universell ist, sondern ein speziell entwickeltes externes Gerät benötigt, das speziell für das Auslesen des TTIs und der unterschiedlich eingesetzten Symbolcodes entwickelt wurde. Funktionell beschränkt es sich lediglich auf das Detektieren von Temperaturschwankungen und erkennt keine weiteren Einflussfaktoren aus der Umwelt.

## 2. Gligoric et al.

Gligoric et al. (2019) berichten über das oben erwähnte Projekt „TagItSmart“. Bei diesem soll eine Methode für die Identifikation und den Datenaustausch zur Verfolgung des Lebenszyklus eines Produkts über das Internet der Dinge (eine Plattform) ermöglicht werden. Dabei erwähnen sie die Entwicklung zweier intelligenter Codes, wobei einer der sogenannte „Upcode“ und der andere ein unbenannter, um einen sensorischen Bereich erweiterten, QR-Code sein soll. Das Farbmittel für den erweiterten QR-Code war laut der Autoren eine reversible thermochrome „Chamäleon“ Pigmentdispersion vom Hersteller LCR Hallcrest, die ursprünglich für den Flexodruck entwickelt und durch einen Verdünnungsprozess für das Inkjetdruckverfahren angepasst wurde. Der exemplarische Upcode (Abbildung 3) basiert hauptsächlich auf einem binär statischen DataMatrix Code, der mit einem nicht zusammenhängenden reversiblen Streifen im Hintergrund (grün gekennzeichnet) mittels Inkjetdruckverfahren verdruckt wurde. Vorgeblich soll der Hintergrund des DataMatrix Codes seit Dezember 2017 um zusätzlich bis zu drei dynamische Bereiche erweitert werden können (siehe Abb., links). Hierzu sind jedoch außer der Abbildung keine weiteren Informationen verfügbar. Bei einem externen Einfluss auf den Upcode soll sich dieser Streifen im Hintergrund farblich verändern und kann theoretisch mit einer erweiterten optischen Barcode-Scanner-App ausgelesen werden. Hierzu war jedoch keine praktische Umsetzung aufzufinden. Es findet sich lediglich eine klassische Barcode-Scanner App, die laut Google Play Store folgende statische Symbolcode-Formate „2D-, Bidi- und 1D-Barcodes (Datamatrix / QR / EAN 8 & 13)“ unterstützt. Trotz eingehender Recherche sind keine Informationen über eine Realisierung des Beispiels einer Scanner-App für dynamische Informationen oder eines Upcodes, dessen Funktionalität gezeigt wird, aufzufinden. Lediglich finden sich Informationen über eine Managementplattform, welche bereits zuvor beschrieben wurde.



Abbildung 3 Funktionsdarstellung eines „Upcode“ Mockups (Gligoric et al., 2019)

3. Heilmann et al.

Der intelligente Code von Heilmann et al. (2008) dient der Überwachung einer Warenverpackung (Abbildung 4). Das Datenfeld besteht aus einem Code, einem Indikator, der sich im Hintergrund des Codes befindet und einem Bereich für die Kalibrierung der Indikatorfarben. Die Erfassung des intelligenten Codes basiert auf einem Mobiltelefon mit integrierter Kamera, mit denen die Daten mittels einer auf dem Endgerät installierten App (Prozessor) verarbeitet werden. Mithilfe von Referenzwerten können die Farbänderungen des Indikators gemessen und bewertet werden. Der Ergebnisbericht wird auf dem Display des Mobiltelefons angegeben. Die Erfindung soll zur Überwachung der Haltbarkeit eines Produkts, zur Prüfung der Echtheit sowie zu weiteren Einsatzzwecken eingesetzt werden. Hierzu beschreibt Heilmann et al., dass der Indikator für unterschiedliche Umwelteinflüsse bestückt werden soll, formuliert dies jedoch nicht weiter aus. Ebenfalls sind unklare Aussagen zum Indikator derart zu finden, der aus einer sensorischen Farbe und anderen Zusätzen bestehen soll. Die Idee eines Monitoring Systems wird innerhalb der Patentschrift gesichert, jedoch sind keine Informationen aufzufinden, wie die Umsetzung genauer erfolgen soll.

Eine bemerkenswerte Ähnlichkeit ist zwischen dem intelligenten Code von Heilmann et al. und dem Upcode zu erkennen. Beide greifen auf einen DataMatrix Code zu, der im Hintergrund des Codes mit einem einzelnen Indikator versehen wurde.

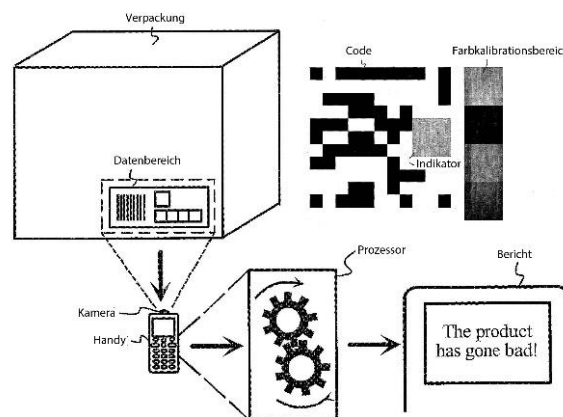


Abbildung 4 Monitoring system (Heilmann et al., 2008)

4. Prusik et al.

Der intelligente Code von Prusik et al. (2016) dient, wie die oben vorgestellten Verfahren, ebenfalls der Überwachung von unterschiedlichen Waren (Abbildung 5). Dieses Verfahren kombiniert vorgedruckte statische Informationen und dynamisch kodierte Sensorinformationen innerhalb eines zweidimensionalen Codes. Die Besonderheit im Gegensatz zu anderen erweiterten Symbolcodes besteht darin, dass hier die dynamischen Informationen in den statischen Code (bspw. QR-Code

oder Data-Matrix Code) integriert werden und die Fehlerkorrekturfunktion für die Wiederherstellung partieller Bereiche wirksam bleibt. Somit kann dieser intelligente Code durch unterschiedliche Lesegeräte erfasst und ausgelesen werden. Informationen über den Grad einer Kontamination, bspw. durch eine farbliche Veränderung eines dynamischen Bereichs, kann anhand der Erzeugung eines Binärbildes im Dekodierungsprozess nicht gegeben werden. Dieser zeigt lediglich das Vorhandensein eines neuen Bereichs an – wie stark dieser Bereich ausgeprägt ist, wird jedoch nicht ausgesagt. Generell ist der intelligente Code lediglich für einen dynamischen Bereich ausgelegt. Gemäß der Patentschrift kann der einzelne dynamische Bereich durch die Veränderung der Größe des Codes und somit der Speicherkapazität, um einen weiteren dynamischen Bereich erweitert werden. Somit ist die Erweiterbarkeit auf die Speicherkapazität und Größe des jeweiligen Codes eingeschränkt.

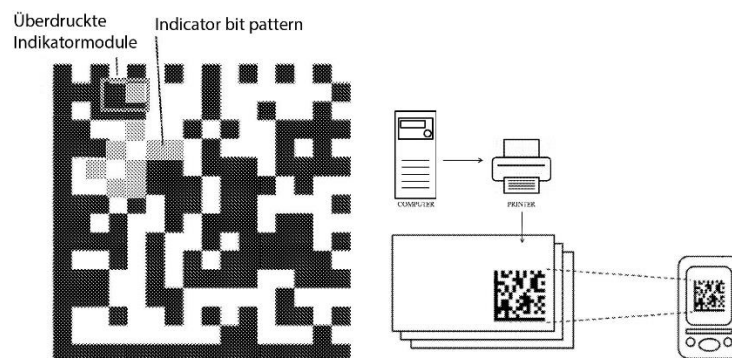


Abbildung 5 Erweiterter 2-D Code (Prusik et al., 2020)

## 5. Ribi

Ribi (2012) stellte eine Methode vor, die einen Barcode durch einen eigens entwickelten TTI Indikator erweitert und somit die Hinweise auf eine mögliche Einschränkung der Verzehrbareit des jeweiligen Produktes visuell darstellt (Abbildung 6). Somit kann seine sensorische Farbe als Hinweis über die Frische von Lebensmitteln auf Lebensmittelverpackungen eingesetzt werden. Die Indikatorzusammensetzung der Erfindung enthält eine thermochrome Polymerzusammensetzung, die je nach Stimulus einer Farbänderung, die reversibel oder irreversibel sein kann, unterliegt. Weitere Einflussfaktoren werden jedoch außer Acht gelassen.

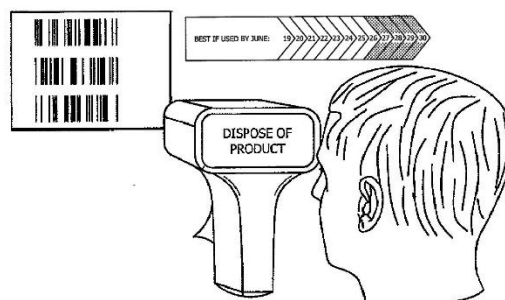


Abbildung 6 Co-topo-polymeric indicator (Ribi, 2012)

Die hier exemplarisch vorgestellten intelligenten Codes und auch die in Tabelle 4 gelisteten Teilelemente stellen Gemeinsamkeiten und Unterschiede der jeweiligen Umsetzungen intelligenter Codes dar. Diesbezüglich kann zusammenfassend festgehalten werden, dass die meisten Verfahren einen klassischen Symbolcode um einen oder mehrere dynamische Bereiche erweitern. Allerdings, das wird zumeist verschwiegen, führt eine derartige Integration von dynamischen Bereichen in einem klassischen Symbol-

code nach Ausführung des spezifischen Codes zu Verlusten bestimmter Funktionalitäten (bspw. Deaktivierung der Fehlerkorrektur, um Bereiche des Codes als dynamische Bereiche freizulegen). Zur Erläuterung: Ändert sich nur ein Bit eines Codes, was ja gerade die spezifische Eigenheit eines dynamischen Codes ist, sind die Korrekturalgorithmen unbrauchbar, da sie, anhand der im Code gespeicherten Daten zur Fehlerkorrektur, das geänderte Bit in den Ausgangszustand korrigieren. Somit gehen, wenn keine Vorkehrungen getroffen werden, die gewonnenen dynamischen Informationen beim Auslesen des Codes und Korrigieren der Fehler wieder verloren. Symbolcodes, wie bspw. QR-Codes, Barcodes, Data-Matrix Codes etc. weisen speziell zugeschnittene binäre Datenbereiche auf. Diese wurden nicht dafür ausgelegt, durch dynamische Bereiche erweitert zu werden. Verschiedene Ansätze behelfen sich damit, einen Defekt in einen Symbolcode herbeiführen (bspw. durch das Entfernen eines Datenbereichs). Dabei wird der Defekte Teil des Codes mittels einer sensorischen Tinte, im nicht sichtbaren Bereich verdruckt und erscheint nach einem definierten Stimulus einer Kontamination (bspw. Temperaturüberschreitung) im notwendigen farbigen Kontrastverhältnis, damit der Symbolcode wieder vollständig und somit überhaupt gelesen werden kann. Andere Codes, wie der „Upcode“ wiederum verwenden für den dynamischen Bereich den Hintergrund des Symbolcodes, der sich je nach Einfluss farblich verändert und die statischen Daten im DataMatrix Code unangetastet lässt. Fürs Verdrucken der sensorischen Tinten ist jedoch eine erhöhte Farbintensität notwendig, ohne die das Kontrastverhältnis und somit die Lesbarkeit des Codes beeinträchtigt wird.

Sobald ein klassischer Symbolcode in seiner Struktur verändert wird, kann nicht mehr auf eine klassische Barcode-Scanner App zurückgegriffen werden. So finden sich auch Codes, welche die Fehlerkorrektur ausnutzen und einen dynamischen Bereich in die Mitte des QR-Codes setzen. Ribi lässt Teilsegmente (Streifen) eines Barcodes mittels sensorischer Farben erscheinen und verändert dadurch den Inhalt des Barcodes. Dieses System ist jedoch in sich geschlossen und kann nicht erweitert werden. Prusik et al. wiederum integrieren einen dynamischen Bereich in den Code, ohne diesen zu beeinträchtigen oder die Fehlerkorrektur auszuschalten. Dies ist zwar vorteilhaft, schränkt jedoch die Erweiterbarkeit und Aussagekraft des sensorischen Bereichs aufgrund der zuvor erwähnten Binarisierung ein.

Heilmann et al. ergänzen den Barcode zumindest theoretisch, in dem sie diesem einen Indikator und einen Farbkalibrationsbereich hinzufügen und diesen mit der integrierten Mobiltelefon-App entsprechend auslesen lassen.

Zum Zeitpunkt der thematischen Auseinandersetzung mit dem Thema dieser Dissertation war noch keine praktische Umsetzung bekannt, in der mehrere sensorische Farben in einem zweidimensionalen Code, insbesondere einem QR-Code, integriert wurden, die aufgrund ihrer Stimuli indizierten Anregung die gespeicherten Daten innerhalb des Codes irreversibel verändern (Steffens, 2017). Ein um mehrere dynamische Bereiche erweiterter QR-Code wurde gemeinsam mit Steffens (2017) realisiert.

Einen eigenen Ansatz liefert der eigens entwickelte intelligente Code, welcher im *Kapitel 5 Aufbau und Funktion eines intelligenten Codes* unter den Gesichtspunkten des *Kapitels 5.2 Kriterien für einen intelligenten Code* zu finden ist.



## 2.3 Smart Labels

Smart Labels, die ebenfalls unter dem Begriff *Smart Tags* bekannt sind, finden häufig ihre Anwendung in Form von passiven RFID- oder NFC-Transpondern, die oft mit maschinenlesbaren eindimensionalen Codes versehen werden. Smart Labels, die auf elektromagnetischer Induktion oder auch auf leitfähigen Tinten basieren, werden an dieser Stelle nicht behandelt, weil sie vom Kern der vorliegenden Forschungsarbeiten inhaltlich zu weit entfernt sind. Neben dem Begriff *Smart Label*, finden sich ebenfalls weitere Bezeichnungen wie bspw. *Smart Packaging*, *Active Packaging* und *Intelligent Packaging*, die sich thematisch überschneiden bzw. ähneln. Das historisch gewachsene Verständnis von „Verpackung“ hat sich von der reinen Schutzverpackung mit aufgedruckten graphischen Elementen (Text und Bild), die vor äußeren Einwirkungen wie Feuchtigkeit oder Sauerstoff schützt, gewandelt. Es wurde um Elemente, die eine aktive Funktion der Verpackung ermöglichen, erweitert. Dieses erweiterte Verständnis von Verpackungen zeichnet sich durch physikalische, chemische oder biologische Maßnahmen zur Veränderung der Wechselwirkungen zwischen der Verpackung und dem Produkt aus (Yam et al., 2005). Bisher finden sich in der einschlägigen Fachliteratur noch keine klaren, einheitlichen Definitionen und Bezeichnungen für die gleichen Bezeichnungen wie bspw. Smart Packaging, weshalb eine einheitliche Verwendung von Bezeichnungen bei verschiedenen Autoren nicht vorzufinden ist. Dieser Problematik widmeten sich Yam et al. aus einer historischen Perspektive und beschrieben diese. Smarte Labels dienen der indirekten qualitativen Messung sowie Überwachung des Produktzustands mittels dynamischer Informationen über die Umgebung eines Produkts.

Labels mit statischen Daten oder Links wie bspw. erweiterte Produktinformationen sind in Form der zur Identifikation eingesetzten eindimensionalen Symbolcodes weit verbreitet. Auch einfache dynamische Labels finden Anwendung. Ein bekanntes Beispiel dafür sind TTIs, die auf einer Lebensmittelverpackung aufgebracht, die Unterbrechung einer Kühlkette bei Lebensmitteln registrieren und eine dadurch entstandene Haltbarkeitsminderung oder Verzehrbeschränkung des Produktes visualisieren können. Diese Visualisierung der Smart Labels basiert häufig auf dem Prinzip einer Stimuli induzierten chemischen Farbveränderung, die in Form von speziellen Indikatoren oder Farbstoffen auf Substraten den Zustand des Produktes unumkehrbar anzeigen. Diese können jedoch auch durch das Zusammenwirken verschiedener Materialien oder unterschiedlich aufgetragenen Schichtdicken ausgeprägt sein, die sich in Abhängigkeit der Zeit fortlaufend potenzieren und Veränderungen des jeweiligen Stimulus stufenweise anzeigen. Die am weitesten verbreiteten Smart Labels sind neben den TTIs die Labels zur Überwachung der mikrobiellen Belastung für Fleisch- und Fischprodukte. Ihre Verfügbarkeit kann aufgrund der erhöhten Gesundheitsgefahr durch den Verzehr mikrobiell belasteter und somit verdorbener Fleisch- und Fischprodukte erklärt werden.

Die Erfordernisse dieser Labels kann mit der erhöhten Gesundheitsgefahr durch den Verzehr mikrobiell belasteter und somit verdorbener Fleisch- und Fischprodukte erklärt werden. Die Schädigungen sind meist am Produkt selbst nicht erkennbar. Sie entstehen vielfach durch die Unterbrechung der Kühlkette bei unsachgemäßer Lagerung oder beim Transport. Wie bereits im *Kapitel 1.2* einschränkend angemerkt, geben diese qualitativen Labels häufig nur eindimensionale Aussagen ausschließlich über einen Einflussfaktor. Sie können weitere Einflüsse auf den Gesamtzustand des Packguts nicht identifizieren und lassen sich häufig auch nicht maschinell auslesen. Informationen, die zum Zweck der Auswertung und Optimierung einer Lieferkette genutzt werden könnten, werden zudem häufig außer Acht gelassen.

Kritisch gesehen werden qualitative Labels zum einen vom europäischen Lebensmitteleinzelhandel aufgrund der strikten Richtlinien für alle Materialien, die mit Lebensmitteln in Kontakt kommen dürfen und finden unter anderem dadurch keine starke Akzeptanz. Zum anderen haben Studien die Haltung der Verbraucher gegenüber TTIs auf gekühlten Lebensmitteln untersucht (Pennanen et al., 2015). So konnte gezeigt werden, dass die verlässliche Kontrolle der Produktqualität der Lieferkette durch die Verbraucher bevorzugt wird. Andererseits wird jedoch eingewandt, dass diese qualitativen Labels von den Produzenten, Lieferanten und Händlern möglicherweise manipuliert werden oder es zu steigenden Produktkosten kommen könnte (ebd.; Aday & Yener, 2015). So gibt es zudem seitens der Produzenten und Händler Verunsicherungen, die Bedenken bezüglich der Schädigung des Markenimages und des Anstiegs an Warenverlusten haben. So befürchten sie, dass es aufgrund leichter Verfärbungen der qualitativen Labels zu Warenverlusten kommen könnte, denn obwohl die Ware noch unbedenklich ist, könnten Verbraucher allein auf die Kennzeichnung vertrauen und solche Waren prinzipiell meiden. Erste erfolgreiche Ansätze finden sich jedoch auf internationaler Ebene, auf der die qualitativen Labels bereits getestet und angewandt wurden (Ahvenainen).

Bezeichnung	Unternehmen	Land (ISO 3166)	Art	Farbe	Lesegerät notwendig
Ageless Eye	Mitsubishi Gas Chemical	JPN	GS	Pink-Blau	Nein
CheckPoint	Vitsab International	SWE	TTI	Grün-Rot	Nein
CoolVu	Freshpoint	ISR	TTI	Grau-Schw.	Nein
eO	3M	USA	GS	Rot-Grün	Nein
Food Sentinel System	SIRA Technologies	USA	BS	Trans.-Rot	Ja
Fresh Label (Concept)	Kyocera Corporation	JPN	TTI	Weiß-Blau	Nein
Fresh-Check	TempTime	USA	TTI	Rot-Schw.	Nein
Fresh-Tag	COX Technologies	USA	FI	Grün-Rot	Nein
Freshcode	Varcode	USA	TTI	Weiß-Schw.	Ja
Freshcode Label	Kao Chimigraf	ESP	FI	Weiß-Schw.	Nein
Gıda tazelik indikatörü	TeknoKim Kimya	TUR	BS	Gelb-Grün	Nein
Humonitor	Sigmaaldrich	USA	HI	Weiß-Blau	Nein
Kuru Isik Steril. İndik.	TeknoKim Kimya	TUR	TTI	Blau-Schwarz	Nein
Monitor Mark	3M	USA	TTI	Weiß-Blau	Nein
Novas	Insignia Technologys	GBR	FI	Mehrere	Nein
O2xyDot	OxySense	USA	GS	Rotes Licht	Ja
Oli-Tec	Open Life Indicator Technology	GBR	TTI	Ro.-Gr.-Ge.	Nein
OnVu	Biezerba, Ciba S.C., FreshPoint	DEU, CHE, USA	TTI	Blau-Weiß	Nein
Ripesense	Ripesense	NZL	BS	Ro.-Or.-Ge.	Nein
SensorQ	FQSI	USA	BS	Or.-Braun	Nein
Tell-Tab	IMPAK	USA	GS	Pink-Blau	Nein
Thermax	LCR Hallcrest LLC	USA	TTI	Weiß-Schw.	Nein
ThermoTrace	Deltatrek	USA	TTI	Weiß-Schw.	Ja
Timestrip Plus	Timestrip	GBR	TTI	Weiß-Blau	Ja
TopCryo	Cryolog	FRA	TTI	Grün-Rot	Nein
Toxin Guard	Toxin Alert	CAN	BS	Weiß-Blau	Nein
Traceo	Cryolog	FRA	FI	Weiß-Rot	Ja
Warm Mark	Shockwatch	GBR	TTI	Weiß-Rot	Nein
WiseHIC	Wisepac Technology Co.	CHN	HI	Blau-Weiß	Nein
Zaman-Sicaklik-Indik.	TeknoKim Kimya	TUR	TTI	Weiß-Blau	Nein

Tabelle 5 Kommerzielle Smart Labels

Legende:

Ro. (Rot), Ge. (Gelb), Gr. (Grün), Or. (Orange), Schw. (Schwarz), Trans. (Transparent)

BS (Bio sensor), FI (Freshness Indicator), GS (Gas sensor), HI (Humidity sensor), TTI (Time-Temperature-Indicator)

In Tabelle 5 werden kommerzielle Smart Labels vorgestellt. Ihre Bezeichnung, Herkunft (Unternehmen, Land), Einsatzzweck (Art), Farbe (Farbveränderung) und die Notwendigkeit eines Lesegeräts zum Auslesen werden dargestellt. So werden die jeweiligen Funktionen und Eigenschaften ausgewählter Smart Labels, die auf ein Lesegerät zurückgreifen (grau markiert) im Folgenden eingehender betrachtet:

### 1. Food Sentinel System

Das Smart Label mit der Bezeichnung Food Sentinel System (Abbildung 7) basiert auf einem Biosensor, welcher in Form einer Indikatortinte für die Analyse von Immunzellen entwickelt wurde. Dieser kann als Bestandteil eines UPC-Barcodes die Kontamination durch pathogene Bakterien in Lebensmitteln (z. B. Salmonellen, Listerien, E. coli, usw.) kontinuierlich durch eine unumkehrbare Farbveränderung von Transparent nach Rot nachweisen. Das Prinzip (Abb., links) basiert auf einem mehrschichtigen Aufbau, welcher aus einer undurchlässigen Trägerschicht besteht, auf der eine primäre Kapillarschicht mit Membran und drüber eine abdeckende undurchlässige Filmschicht liegt. Die zweite Schicht führt die Produktsäfte aus der Zuflussöffnung zu einer in der Mitte befindlichen Membrane, die sich zwischen zwei absorbierenden Materialien befindet. Diese ist mit Antikörpern für ausgewählte pathogene Organismen imprägniert. Durch das Aufbringen eines Barcodes auf die primäre Kapillarschicht wird der Barcode selbst zum integralen Bestandteil des Systems (Goldsmith et al., 1999). Dieser Barcode reagiert auf einen externen Stimulus mit einem initiierten Farbveränderungsprozess (Abb., rechts). Hierbei wird durch das Anheften unerwünschter Organismen an den Antikörper eine rote Farbe sichtbar, welche die Lesefunktionalität des Barcodes verhindert. Somit ist der Barcode entwickelt worden, um eine bestimmte bakterielle Kontamination zu detektieren und die jeweilige Nutzung des Produktes einzuschränken. Dieses System wurde 1998 unter der Bezeichnung „Food Sentinel System“ zum Patent angemeldet.

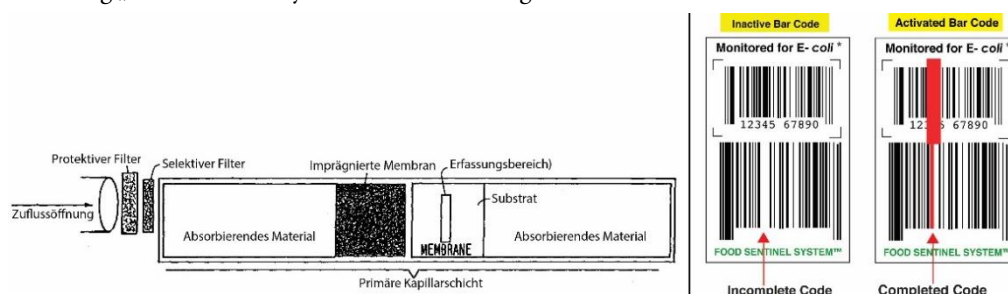


Abbildung 7 Food Sentinel System; Aufbau und Funktion (Carlos, E. et al., 2000)

### 2. Freshcode

Das Smart Label mit der Bezeichnung Freshcode (Abbildung 8) basiert auf einem chemischen Diffusionsprozess, der temperaturbedingt eine Werteänderung eines Datenbereichs innerhalb des Barcodes bewirkt und als TTI der Überwachung einer Kühlkette dient. Die Funktionalität des intelligenten Barcodes ist auf ein Initialisierungsprinzip zurückzuführen, das durch das Entfernen einer Lasche aktiviert wird. Hinter der Lasche befindet sich eine Rückschicht, die ein temperaturempfindliches färbendes Element beinhaltet, das mit einem Farbstoff und Lösungsmittel imprägniert ist. Der aktuelle Zustand des Barcodes wird solange angezeigt, bis die Temperatur für eine vorbestimmte Mindestdauer die vorbestimmte Temperatur nicht überschreitet (z. B. 21 °C für eine Stunde). Solange die Lasche nicht entfernt ist, reagiert das temperaturempfindliche färbende Element samt Lösungsmittel mit dem Qualitätsindikator nicht auf die Temperaturänderungen. Vor der Überschreitung verhindert die Aktivierungsverzögerungsschicht das Diffundieren des Farbmittels. Nach der

Überschreitung des Temperaturbereichs wird diese Schicht jedoch durch ein sich auf der Rückenschicht befindendes Lösungsmittel aufgelöst, diffundiert in das Filterpapier (einzufärbendes Element) und färbt es schwarz ein. Dieses erscheint nun durch den transparenten Bereich (x), lässt einen neuen Barcodeabschnitt erscheinen und macht dadurch den neuen Zustand des Barcodes sichtbar (Nemet & Brand, 2013).

Diese maschinenlesbare Barcodeanzeige des intelligenten Barcodes zeigt mindestens einen von drei Schwellenwerten (x) an, wodurch unterschiedliche Zahlenkombinationen generiert werden. Mittels Barcode-Lesegerät findet ein Abgleich des Barcodes statt und gibt dabei unterschiedliche Aussagen über den jeweiligen Zustand des Produktes wieder. Bei diesem Smart Label wird lediglich ein Parameter erkannt, der die Produktqualität beeinflusst. Dieser kann zwar von jedem gängigen Barcode-Lesegerät ausgelesen, jedoch nicht in eine Datenbank eingetragen oder interpretiert werden. Hierfür wird eine spezielle App des Herstellers benötigt. Über eine Internetverbindung und eine Datenbankverknüpfung können Informationen zum Produkt abgerufen und abgeglichen werden. Durch die Wahl des Zeitpunkts für die Aktivierung des Smart Labels, besteht eine Möglichkeit zur Manipulation. Dieses System repräsentiert eine Gruppe von Smart Labels, die nach dem gleichen Prinzip funktionieren und unter verschiedenen Markennamen vertrieben werden.

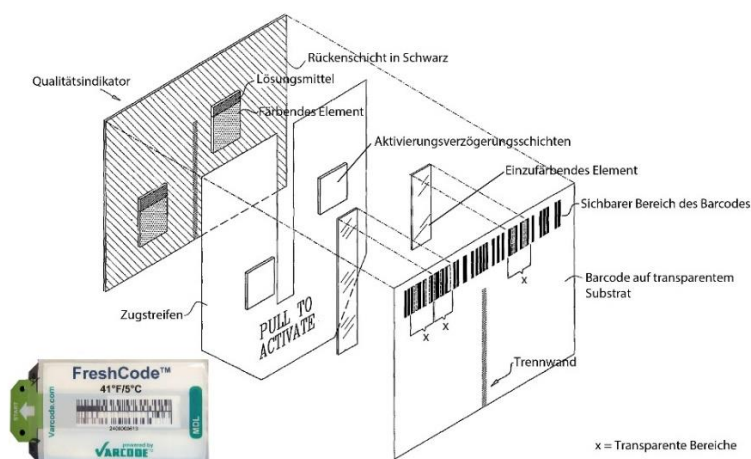


Abbildung 8 Freshcode (Nemet & Brand, 2013)

### 3. O2xyDot (OxyDot)

Das Smart Label mit der Bezeichnung O2xyDot (Abbildung 9) ist im eigentlichen Sinne ein optisches Sensorsystem. Dieses System ist in der Lage, die Veränderung von Sauerstoffkonzentrationen in transparenten, halbtransparenten und transluzenten Verpackungen oder auch in Sauerstoff empfindlichen Flüssigkeiten nicht-invasiv zu messen. Die Sauerstoffmessung basiert auf dem Prinzip eines faseroptischen Lesestifts, in dem sich eine LED mit blauem Licht befindet. Dieser Lesestift wird auf einem O2xyDot positioniert, der zuvor auf die Innenseite des zu untersuchenden Objekts aufgebracht wird. Hierbei wird das Licht im blauen Bereich vom Fluoreszenzfarbstoff (lumineszierende Verbindung) des O2xyDot absorbiert, in einen angeregten Zustand versetzt und wenn die Belichtung des Lichts der LED beendet ist, emittiert die angeregte lumineszierende Verbindung des O2xyDot rotes Licht (fluoresziert). Das rote Licht wird daraufhin von einem optoelektronischen Sensor detektiert und die Eigenschaften der Fluoreszenzlebensdauer gemessen. Die Fluoreszenzlebensdauer wird durch die Anwesenheit von Sauerstoff verkürzt. So weisen unterschiedliche Lebensdauereigenschaften auf unterschiedliche Sauerstoffgehalte innerhalb der Verpackung hin. Hierbei ist

jedoch anzumerken, dass die Anwesenheit von Sauerstoff das Fluoreszenzlicht des Farbstoffs sowie dessen Lebensdauer auslöscht. Der Löschvorgang ist ein rein kollisionsdynamischer Prozess, bei dem die Energie des angeregten Fluoreszenzfarbstoffs während einer Kollision auf das Sauerstoffmolekül übertragen wird, wodurch sowohl die Emissionsintensität als auch die Fluoreszenzlebensdauer (korreliert mit dem Sauerstoffpartialdruck) des Farbstoffs verringert wird (Havens & Barmore, 2010). Nachteilig an diesem Sensorsystem ist, dass das O2xyDot System auf einem externen Messgerät basiert, das über einen faseroptischen Lesestift einen einzelnen Punkt (Dot) misst und spezifisch auf die nicht-invasive Sauerstoffmessung innerhalb eines Mediums ausgelegt und auf eben dieses unhandliche System begrenzt ist.

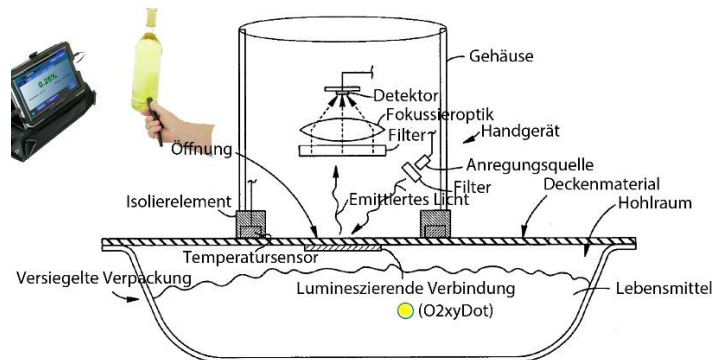


Abbildung 9 O2xyDot (Havens & Barmore, 2010)

#### 4. Timestrip Plus

Das Smart Label mit der Bezeichnung Timestrip Plus (Abbildung 10) basiert auf einem blisterartigen TTI und zeigt die Summe der Zeiten an, in denen eine Schwellenwerttemperatur überschritten wird. Das System basiert auf einer Indikatorflüssigkeit, die unterhalb einer bestimmten Temperatur fest und oberhalb dieser flüssig ist. Vor der Nutzung innerhalb einer Kühlkette muss das Reservoir der Indikatorflüssigkeit manuell durch Druck aktiviert werden, wodurch das Reservoir zerbricht und bis zu einem gewissen Bereich austritt. Bei einer Temperaturüberschreitung wird die im Reservoir enthaltene Indikatorflüssigkeit fließfähig gemacht und migriert daraufhin durch ein mehrlagiges Schichtensystem in ein saugfähiges Medium und kann dieses abschnittsweise in Abhängigkeit der Zeit verfärben. Dieses System zeigt somit abschnittsweise die vergangene Zeit nach der Temperaturüberschreitung visuell über eine Skala an und kann zusätzlich mit einer App des Unternehmens die Zahlenkombination und Teilbereiche der Skala erfassen und wiedergeben (Magen & Isbitsky, 2019). Somit kann jeweils ein Timestrip Plus eine bestimmte Temperaturüberschreitung anzeigen und ist lediglich für einen eindimensionalen Einflussfaktor ausgelegt.

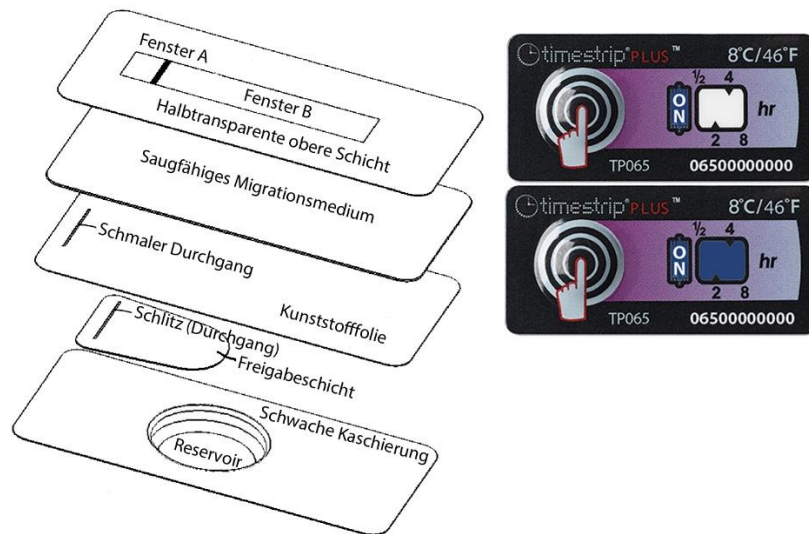


Abbildung 10 Timestrip Plus (Magen & Isbitsky, 2019)

## 5. Traceo

Das Smart Label mit der Bezeichnung Traceo (Abbildung 11) basiert auf einem mikrobiologischen Frischeindikator. Dieser verhindert in Form eines Gels (Kultursubstrat) auf einem transparenten Label das Auslesen eines Barcodes, wenn das Produkt nicht ordnungsgemäß gelagert wurde (Kühlkettenunterbrechung). Das transparente Smart Label wird über einem Barcode angebracht. Aufgrund von Kontamination ändert dieser seine Farbe von Transparent zu Violett-Rot. Durch die Farbänderung wird der darunterliegende Barcode farblich überdeckt und ist nicht mehr lesbar. Das transparente Gel des Smart Codes enthält Mikroorganismen, welche bei Kontamination den Verfall des Packguts simulieren. Dies geschieht durch die Wachstumssteigerung oder alternativ auch durch die Wachstumshemmung von Mikroorganismen. Das Wachstum der Mikroorganismen auf dem Kultursubstrat wird während der Handhabungsphasen durch einen sauren pH-Wert und Inhibitoren in Form von Dehydrierungsmitteln gehemmt. Bei einer temperaturbedingten Schwellenwertüberschreitung wird das Wachstum der Population von Mikroorganismen über den Barcode beeinflusst. Insbesondere geschieht dies durch einen temperaturbedingten Aktivierungsprozess und die Freisetzung von Wasser (Rehydrierungsmittel). Zudem wird ein im Patent nicht erwähnter Stoff zur Einstellung des pH-Wertes hinzugefügt, der durch den Kontakt mit Wasser aktiviert wird und das Wachstum der Kulturen ermöglicht. Der veränderte pH-Wert der Kulturen und der Indikator reagieren miteinander und bewirken eine Farbänderung. Folglich hängt die Zeit, um den Barcode unleserlich zu machen, von der Temperatur und der Dauer der Einwirkung der kritischen Temperaturen ab (Vaillant, 2010).

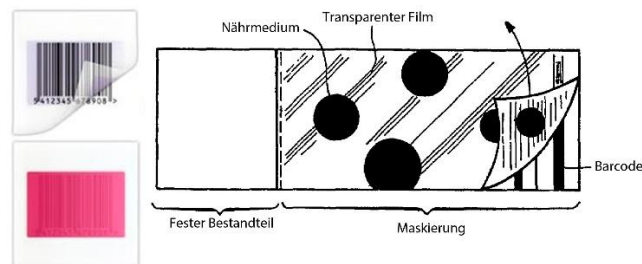


Abbildung 11 Traceo (Vaillant, 2010)

Die vorgestellten Beispiele sind typische Vertreter der in der Tabelle aufgeführten Smart Labels, welche entsprechend nicht einzeln dargestellt werden. Alle aufgelisteten Smart Labels sind irreversibel und zeigen somit unumkehrbar die Belastung des Labels an.

Diese bestehen aus folgenden Komponenten:

- a) Funktionales Element  
Meist ein Farbumschlag, der durch einen äußeren Einfluss herbeigeführt wird. Dabei erfolgt der Farbumschlag durch unumkehrbare chemische oder mikrobiologische Prozesse aber auch die Ausbreitungsgeschwindigkeit einer Fließfront, wie dies beim Timestrip Plus dargestellt wurde.
- b) System (einfach oder komplex)  
Dient der Gewinnung von Informationen des funktionalen Elements.  
Direkt (Skala) oder mit Hilfsgeräten (Barcodeleser, Lesestift unter anderem.)
- c) Unterschiedliche Arten der Dynamik  
Zweipunktsensoren: Sie zeigen nur den Zustand unbelastet oder belastet an.  
Graduelle Sensoren: Sie zeigen die Gesamtintensität der Belastung an (z. B. Freshcode)
- d) Warenkontakt  
Diese sensorischen Farben sind zumeist nicht direkt mit dem Packgut verbunden.  
Eine bedingte Ausnahme bildet OxyDot. Es wird häufig ein Label vorgefertigt und an der Verpackung appliziert. Dies bedeutet, dass die sensorischen Farben nicht den Zustand des Packguts anzeigen, sondern lediglich indirekt auf diesen schließen lassen.
- e) Eindimensionale Einflussfaktoren  
Alle bestehenden Smart Labels indizieren lediglich eine Art der Kontamination, Multifunktionale, ausgenommen TTIs, sind bisher unbekannt.
- f) Vernetzung  
Die meisten sensorischen Farben sind reine Anzeigesensoren, ob direkt oder mit Lese- und Hilfsgeräten. Erste einfache Ansätze, die Sensorinformationen auch datentechnisch zu verarbeiten und auszuwerten, sind zu erkennen (Freshcode).

## Kapitel 3

# Methoden

Das folgende Kapitel stellt die methodische Herangehensweise dar und charakterisiert die verwendeten Materialien und Werkstoffe dieser Dissertation. Zudem werden die Interdependenzen der einzelnen Teilziele hervorgehoben. Im Anhang (A) sind die Parameter und die gewählten Einstellungen der verwendeten Geräte zusammengefasst dargestellt.

### 3.1 Materialien

Im Folgenden werden die im Rahmen der Forschungsarbeiten verwendeten Materialien behandelt, die für weitere Untersuchungen von drei unterschiedlichen sensorischen Farben und auch als sensorische Tinten verwendet. Die eigens für den Laborbetrieb entwickelten sensorischen Tinten werden genauer als photochrome (Tabelle 6), halochrome (Tabelle 7) und hydrochrome Tinten (Tabelle 8) bezeichnet. Für die Entwicklung der jeweiligen sensorischen Farben wurden die in den Tabellen (6 – 8) aufgeführten Materialien herangezogen und auf ihre Eignung und Funktionalität für die jeweilige sensorische Farbe hin untersucht. Die aufgelisteten Materialien selbst bilden dabei nicht die jeweilige Zusammensetzung der sensorischen Farben und Tinten ab, diese werden im jeweiligen Kapitel (*Kapitel 4.7*) behandelt.

Die Entwicklung ist mangels marktgängiger Tinten für den Inkjetdruck zurückzuführen. Die Vorgehensweise, aus marktgängigen hydrochromen und photochromen Siebdruckfarben für den Inkjetdruck geeignete Tinten herzustellen (Bilgin & Backhaus, 2018b, 2018c), erwies sich als bedingt geeignet. Zwar wurde das Ziel durch die Identifizierung eines geeigneten Lösungsmittels und eines entsprechenden Verdünnungs- und Filtrationsprozesses erreicht. Längerfristig sedimentierten jedoch die Partikel und führten zum Verstopfen der Düsen des Druckkopfes. Zudem legen die Farbenhersteller weder die chemische Zusammensetzung der drei Siebdruckfarben noch ihre Funktionsweise offen. Infolgedessen können keine wissenschaftlich belastbaren Aussagen über die Funktionalität, vor allem nicht über die Wechselwirkung der hydrochromen Siebdruckfarbe mit dem Einflussfaktor Wasser, getroffen werden (Bilgin & Backhaus, 2017a, 2019c). Die Entwicklung einer hydrochromen sensorischen Farbe basiert auf den herausgestellten Forschungslücken und -bedarfen zu dem Stand der Forschung im *Kapitel 2.1* mit Ausnahme von Kim et al. (2010) konnten keine weiteren expliziten hydrochromen Inkjettinten ausfindig gemacht werden. Eine hydrochrome sensorische Farbe eignet sich als Indikator für den Feuchtigkeits- und Korrosionsschutz in Verpackungen und Produkten. Die Entwicklung von thermochromen sensorischen Farben wurde aufgrund ihrer chemischen Komplexität und der notwendigen Eignung für unterschiedliche Temperaturklassen verworfen. Eine photochrome sensorische Farbe, speziell für den UV-Bereich, eignet sich unter anderem für den Lebensmittel-, Pharma- und Kosmetiksektor, denn auf den Verderb vieler dieser Produkte hat UV-Licht einen besonders intensiven Einfluss. Halochrome sensorische Farben können beim Kontakt mit einem Lebensmittel über den pH-Wert Änderungen anzeigen. Die Zusammensetzung und die Bestandteile für die Entwicklung der sensorischen Farben bildeten sich innerhalb des Forschungsprozesses und sind an das jeweilige Konzept gebunden, die sensorischen Farben werden in *Kapitel 4.6* ausführlich erläutert.



Die entwickelten sensorischen Farben werden für die Verdruckbarkeit in eine Lösungsmittelmatrix eingebunden. Später wird daraus aus dieser unter Zugabe von Additiven (*Kapitel 3.1.1*) eine inkjetfähige Tinte (*Kapitel 3.2*) formuliert. Die Funktionalität der Tinten und ihre charakteristischen Reaktionsfähigkeiten können aus *Kapitel 4.7* entnommen werden. Zudem können die in den folgenden Tabellen vorgestellten Komponenten, ihre Anteile, Zusammensetzung und Verwendung in eben diesem Kapitel eingesehen werden.

### Photochrome Tinte

Bezeichnung	Summenformel	Kategorie	CAS Nummer
Kaliumhexacyanoferrat(III) (E 536)	$K_3[Fe(CN)_6]$	Wasserfreies Salz	13746-66-2
Ammoniumeisen(III)-citrat	$C_6H_8O_7 \cdot xFe \cdot xH_3N$	Wasserfreies Salz	1185-57-5
Demineralisiertes Wasser	$H_2O$	Lösungsmittel	-

Tabelle 6 Materialien für die Untersuchung einer photochromen Tinte

### Halochrome Tinte

Bezeichnung	Summenformel	Kategorie	CAS Nummer
Diferuloylmethan (E 100)	$C_{21}H_{20}O_6$	natürlicher Farbstoff	458-37-7
Cyanidin (E163)	$C_{15}H_{11}O_6^+ \cdot SO_2$	natürlicher Farbstoff	-
Demineralisiertes Wasser	$H_2O$	Lösungsmittel	-
Ethanol	$C_2H_6O$	Lösungsmittel	64-17-5
Natriumcarbonat (E 500)	$Na_2CO_3$	Wasserfreies Salz	497-19-8
Citronensäure (E 330)	$C_6H_8O_7$	Wasserfreies Salz	77-92-9

Tabelle 7 Materialien für die Untersuchung einer halochromen Tinte

### Hydrochrome Tinte

Bezeichnung	Summenformel	Kategorie	CAS Nummer
Patent Blau V (E 131)	$C_{27}H_{31}N_2NaO_7S_2$	Triphenylmethanfarbstoff	20262-76-4
Natriumcarbonat (E 500)	$Na_2CO_3$	Wasserfreies Salz	497-19-8
Natriumsulfit (E 221)	$Na_2SO_3$	Wasserfreies Salz	7757-83-7
Natriumthiosulfat	$Na_2S_2O$	Wasserfreies Salz	7772-98-7
Natriumbicarbonat E 500 (ii)	$NaHCO_3$	Wasserfreies Salz	144-55-8
Ethanol	$C_2H_6O$	Lösungsmittel	64-17-5
Demineralisiertes Wasser	$H_2O$	Lösungsmittel	-
E24	$H_2O - R$	Lösungsmittel	-
Isopropanol	$C_3H_8O$	Lösungsmittel	67-63-0
Methanol	$CH_4O$	Lösungsmittel	67-56-1

Tabelle 8 Materialien für die Untersuchung einer hydrochromen Tinte

## 3.1.1 Additive

Additive, wie die in Tabelle 9 gelisteten Feuchthaltemittel, Oberflächenspannungsregler und Antischaummittel, können den sensorischen Tinten hinzugefügt werden, um ungünstige Faktoren zu inhibieren, die das Verdrucken der Tinten beeinflussen können. Hierzu wurde bei der hydrochromen Tinte das Feuchthaltemittel Glycerin, das hydratisiert (im flüssigen Zustand) vorhanden ist, durch das dehydratisierte (im trockenen Zustand) Urea (5 mg) ersetzt. Da die hydrochrome Tinte auf Wasser reagieren soll, dürfen Wassermoleküle die Tinte nicht beeinflussen. Des Weiteren wurden bei der halochromen Tinte pH-neutrale Additive wie Glycerin (0,5 ml) verwendet. Zusätzlich könnte auch das Antischaummittel eingesetzt werden. Auf einen Oberflächenspannungsregler wurde aufgrund des negativen pH-verändernden Einflusses verzichtet. Für die photochrome Tinte wurde als Feuchthaltemittel Glycerin

(0,5 ml) eingesetzt. Als Oberflächenspannungsregler kann Decyl Glucoside (0,5 ml) eingesetzt werden. Das Antischaummittel Poly(dimethylsiloxane) fand hier ebenfalls keine Anwendung. Die behandelten Additive in Kombination mit den sensorischen Tinten, wurden in einer Voruntersuchung getestet und beeinflussen nicht die Funktionalität.

Bezeichnung	Summenformel	Kategorie	CAS Nummer
Glycerin	$C_3H_8O_3$	Feuchthaltemittel	56-81-5
Urea	$NH_2CONH_2$	Feuchthaltemittel	57-13-6
Decyl Glucoside	$C_{16}H_{32}O_6$	Oberflächenspannungsregler	68515-73-1
Poly(dimethylsiloxane)	$[Si(CH_3)_2O]_n$	Antischaummittel	63148-62-9

Tabelle 9 Additive für den möglichen Einsatz in den entwickelten sensorischen Tinten

### 3.1.2 Substrate

Ein Haftpapier zur späteren Applikation als Label auf einer Verpackung ist für den intelligenten Code augenscheinlich das am ehesten geeignete Trägermedium. Denn der Direktdruck auf eine Verpackung – sowohl bei der Verpackungsherstellung als auch bei der Abfüllung – erscheint aus Verfahrens-, Geschwindigkeits-, hygienischen und anderen Gründen eher unzweckmäßig zu sein. Der Druck sollte auf den Haftpapieren detailgetreu und kontrastreich wiedergegeben werden. Außerdem darf das Substrat die Funktionalität der sensorischen Farben, die in den intelligenten Code integriert sind, aufgrund einer chemischen oder optischen Wechselwirkung nicht beeinträchtigen. Die Wechselwirkung basiert darauf, dass die chemischen Zusammensetzungen der verwendeten Substrate weitestgehend unbekannt sind und der jeweilige Einfluss der Wechselwirkung somit häufig nicht zu identifizieren ist. Vor diesem Hintergrund ist es notwendig, die in *Kapitel 4.6* entwickelten sensorischen Farben auf ihre wechselseitige Wirkung mit den jeweiligen Substraten hin zu untersuchen und für die sensorische Tinten ein geeignetes Substrat zu identifizieren. Bei den Substraten handelt es sich um Haftpapiere (A – E) und zwei Folien (F und H). So wurden die in Tabelle 10 aufgeführten Substrate mit den in *Kapitel 4.7* aufgeführten sensorischen Tinten bedruckt. Spektralphotometrische Untersuchungen und die entsprechenden Remissionskurven sollen jeweils Auskunft über die Wechselwirkungen geben. In *Kapitel 4.1* wird der erhebliche Einfluss eines Substrats auf die Reaktionsfähigkeit des photochromen Sensors gezeigt.

Nr.	Hersteller / Händler	Produktbezeichnung	Flächengewicht	Dicke	Durchreißwiderstand	Bruchkraft
A	Endisch	ENDI-HAFT (B297420)	70 g/m <sup>2</sup>	90 $\mu$ m	-	Längs: 5.2 kN/m Quer: 2.2 kN/m
B	TP Paperprofi	Digital (8200-320450)	70 g/m <sup>2</sup>	78 $\mu$ m	-	-
C	VPF	Inkjet-Papier (1393264)	100 g/m <sup>2</sup>	120 $\mu$ m	Längs: >600 mN Quer: >600 mN	Längs: >4.0 kN/m Quer: >1.5 kN/m
D	VPF	Inkjet-Papier (1393450)	92 g/m <sup>2</sup>	120 $\mu$ m	Längs: >500 mN Quer: >500 mN	Längs: >4.0 kN/m Quer: >1.5 kN/m
E	VPF	Thermo-Papier, inkjetbedruckbar (1470762)	73 g/m <sup>2</sup>	82 $\mu$ m	Längs: >400 mN Quer: >400 mN	Längs: >4.7 kN/m Quer: >2.2 kN/m
F	VPF	Inkjet-Folie, PP (60241)	85 g/m <sup>2</sup>	58 $\mu$ m	-	-
G	VPF	Inkjet-Folie, PE (71838)	70 g/m <sup>2</sup>	105 $\mu$ m	-	-

Tabelle 10 Untersuchte Substrate

## 3.2 Bestimmung der Tintenanforderungen

Damit sich die sensorischen Tinten für das eingesetzte Drucksystem eignen, wurden Zielparameterfenster definiert. Diese begrenzen den Bereich für das funktionsfähige Verdrucken der Tinten. Für die Bestimmung eines Zielparameterfensters wurden die originalen Tinten des eingesetzten Drucksystems (*Kapitel 4.7*) messtechnisch ermittelt. Leitgrößen waren dabei die Parameter der gerätetypischen Tinten, wie Viskosität, Oberflächenspannung und Partikelgrößenverteilung. Anhand dieser Parameter wurden Bandbreiten mit Unter- (*Minimum*) und Obergrenzen (*Maximum*) definiert und anhand der Empfehlungen der einschlägigen Fachliteratur verifiziert. Im Folgenden werden die Bestimmungsmethoden der Viskosität, Oberflächenspannung und Partikelgrößenverteilung näher beschrieben.

### 3.2.1 Viskosität

Um die rheologischen Eigenschaften der sensorischen Tinten zu ermitteln, insbesondere deren viskoses Fließverhalten an die Tintenanforderungen entsprechend anzupassen, wurde das Analyseverfahren mit einem Physica MCR 101 Rotationsrheometer und einem entsprechenden Kegel- und Plattenmesssystem (CP50-1), mit einem Durchmesser von 49.966 mm und einem Kegelwinkel von  $\alpha = 1^\circ$ , der Firma Anton Paar durchgeführt. Das System erlaubt verschiedene rheologische Untersuchungen, sowohl nach dem Oszillations- als auch dem Rotationsprinzip. Die zu untersuchenden Materialien werden dabei nach dem kontrollierten Schergeschwindigkeitstest (CSR-Test, Controlled shear rate test) durchgeführt, bei dem Prozesse simuliert werden, die von der Strömungsgeschwindigkeit oder dem Volumenstrom abhängig sind (z. B. Strömung durch ein Rohr; Inkjetdruckverfahren). Das entsprechende Messprinzip und die Grundlagen können dabei dem *Kapitel 4.1* entnommen werden.

### 3.2.2 Oberflächenspannung

Für die Bestimmung der dynamischen Oberflächenspannung und der Blasenlebensdauer von Tinten wurde der Blasendruck-Tensiometer der Firma SITA pro line t15 verwendet. Diese Messmethode ist notwendig, da die dynamische Oberflächenspannung für die charakteristischen Merkmale der Tinten, wie Benetzungsvermögen, Tröpfchengröße, Auslaufverhalten und weitere Faktoren relevant sind. In *Kapitel 4.1* finden sich Grundlagen zur dynamischen Oberflächenspannung. Um die Oberflächenspannung der sensorischen Tinten zu ermitteln, wird die Blasendruckmethode zum Messen der dynamischen Oberflächenspannung in Flüssigkeiten eingesetzt.

Bei der Blasendruckmethode wird eine Kapillare in eine Flüssigkeit getaucht und eine Gasblase erzeugt. Das Prinzip der eingesetzten Messmethode basiert darauf, dass durch die innere Anziehungskraft einer Flüssigkeit ebenfalls die in der Flüssigkeit enthaltenden Luftblasen komprimiert werden. Dabei steigt der entstehende Blasendruck mit abnehmendem Blasenradius bis zu einem maximalen Innendruck ( $p_{\max}$ ) an. „Der Druck steigt bis zu einem Maximaldruck an. Bei diesem hat die Blase ihren kleinsten Radius, den Kapillarradius, erreicht und bildet eine Halbkugel.“ (SITA, 2021) Wird dieser Punkt überschritten, vergrößert sich die Gasblase schnell und reißt von der Kapillare ab, wodurch eine neue Blase entsteht. „Den gegenüber der Umgebung der Blase erhöhten Druck macht man sich bei der Blasendruckmethode zu Nutze“ (ebd.), mit der die Oberflächenspannung ( $\sigma$ ) berechnet wird.

Dabei kann die kugelförmige Gasblase ( $r$ ) an der Spitze einer Kapillare mit der Young-Laplace-Gleichung (1) folgend berechnet werden, bei der aufgrund der Oberflächenspannung ( $\sigma$ ) an der Grenzfläche ein um  $\Delta p$  erhöhter Druck herrscht (Krüss, 2020):

$$\Delta p = \frac{2 \cdot \sigma}{r} \quad (1)$$

Der dabei entstehende Druckverlauf wird zur Bestimmung der Oberflächenspannung ausgewertet (ebd.).

### 3.2.3 Partikelgrößenverteilung

Um die Partikelgrößenverteilung in den jeweiligen Proben zu ermitteln, wurde auf ein konfokales 3D Laserscanning Mikroskop der Baureihe VK-X, Fabrikat Keyence, zurückgegriffen. Hierzu wurde die Methode der mikroskopischen Partikelzählung angewandt, wobei jeweils zufällige Stichproben von 100 Partikeln ausgewählt wurden. So ist darauf hinzuweisen, dass diese Untersuchung nicht die Partikelverteilung in der inkjetdruckfähigen Tinte selbst bestimmt, sondern vielmehr qualitativ, unter dem Einsatz eines Konfokalmikroskops, die ungelösten und nach einer Verdunstungstrocknung auf einem Glassubstrat befindlichen Partikel und ihre jeweiligen Größen quantitativ messtechnisch erfasst. Hierzu wurden jeweils 2 ml der unterschiedlichen Konzentrationen, der jeweiligen sensorischen Tinten unverdünnt auf ein Glassubstrat getropft und unter einem Konfokalmikroskop mit einer 50-fachen Vergrößerung gelegt und mit einer Referenzskala von 100  $\mu\text{m}$  markiert. Darüber hinaus wurde ein Rasterelektronenmikroskop (REM) der Modellreihe FEI Quanta 450 mit einem BSED (Backscattered Electron Detector) eingesetzt, um die Partikel genauer zu untersuchen.

### 3.2.4 Filtrationsprozess

Bevor die entwickelten sensorischen Tinten in das *Inkjetsystem* (Kapitel 3.3) befüllt werden, müssen diese einen mehrstufigen Filtrationsprozess durchlaufen (Abbildung 12). Erst dann können sie in den jeweils dafür vorgesehenen Tintenkanal gefüllt werden, um daraufhin den entwickelten intelligenten Code zu drucken. Ansonsten können ungelöste Feststoffe, die eine zu große Partikelgrößenverteilung aufweisen und über ein definiertes Zielparameterfenster hinausgehen, zu zahlreichen technischen Störungen führen (Otterstätter, 2007, S. 70). Zudem können unkontrollierte Bedingungen wie bspw. Agglomeration oder Sedimentation zu Farbtonveränderungen oder zu einer Verstopfung der Tintenstrahldüse und der zugehörigen Leitungen führen (Magdassi, 2010, S. 34).

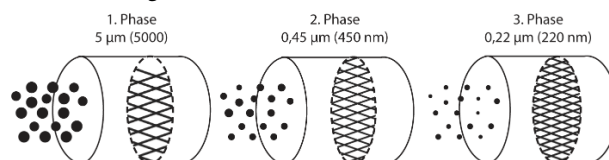


Abbildung 12 Phasen der Filtration

In diesem Zusammenhang wurde die Verteilung der ungelösten Partikel in der Lösung des jeweiligen sensorischen Tinten eingehend untersucht und ein Partikelfiltrationsprozess mit Hilfe der in Tabelle 11 aufgeführten Filtrationsausrüstung durchgeführt. Zuerst wurden die Partikel mit einer Millex SV (SLSV025LS) Filtermembran mit einer Porengröße von 5,0  $\mu\text{m}$  gefiltert. Somit konnten durchschnittlich Partikel größer als 5,0  $\mu\text{m}$  herausgefiltert werden. Anschließend wurden die restlichen Partikel mit

einer Millex HPF HV (SLHVM25NS) Filtermembran mit einer Porengröße von 0,45 µm gefiltert, sodass durchschnittlich Partikel kleiner als 0,45 µm (450 nm) in der Tinte vorhanden waren. Schließlich wurde der dritte Filtrationsprozess mit dem Express Membranfilter angewendet, bei dem durchschnittlich alle Partikelgrößen über 0,22 µm (220 nm) herausgefiltert wurden.

Die genutzten Filtermembranen filtern im Durchschnitt die angegebenen Porengrößen. Der Filtrationsprozess pro Filtermembrane wurde fünffach wiederholt durchgeführt, um einen Näherungswert zur angegebenen Filtereinheit (bspw. 5,0 µm) der Filtermembrane zu erhalten. Durch den Filtrationsprozess weisen die entwickelten sensorischen Tinten den empfohlenen Partikelgrößenbereich zwischen 200 nm bis 300 nm auf (ebd., S.34). Durch diesen aufwendigen Filtrationsprozess wird erreicht, dass große Partikel nicht zur Verstopfung der feinen Tintenstrahlrohre und Düsen führen. Die Viskosität und Oberflächenspannung der drei sensorischen Tinten haben sich nach dem mehrstufigen Filtrationsprozess nicht signifikant verändert.

Bezeichnung	Porengröße	Material	Filterfläche	Max. Eingangsdruck	Benetzbarkeit
Millex-SV (SLSV025LS)	5,0 µm	Hydrophiles (PVDF)	3,9 cm <sup>2</sup>	5.2 bar (75 psi)	Hydrophil
Millex-HPF HV (SLHVM25NS)	0,45 µm	Hydrophiles (PVDF)	3.9 cm <sup>2</sup>	7 bar (100 psi)	Hydrophil
Millipore Express (GPWP04700)	0,22 µm	Polyethersulfone	1,7 cm <sup>2</sup>	690 mbar (10 psi)	Hydrophil

Table 11 Materialien für den Filtrationsprozess  
PVDF (Hydrophilic Polyvinylidene Fluoride)

### 3.3 Inkjet Drucksystem

Für das Verdrucken der drei sensorischen Tinten wurde exemplarisch der Epson EcoTank ET-2650 mit der MicroPiezo TFP (Thin-Film-Piezoelectric) Druckkopftechnologie eingesetzt (Abbildung 13). Dieser wird im Bürogeräte-segment genutzt, um Ausdrücke im Format DIN A4 zu realisieren. Dieses Drucksystem verfügt über ein integriertes Tintensystem. Bei der Düsenkonfiguration wurde der schwarze Farbkanal für das Drucken des statischen Bereichs des intelligenten Codes verwendet und die jeweiligen weiteren Farbkanäle durch die drei entwickelten sensorischen Tinten ersetzt. Hierzu wurde die photochrome Tinte in den Farbkanal Cyan (C), die hydrochrome Tinte in den Farbkanal Magenta (M) und die halochrome Tinte in den Farbkanal Yellow (Y) befüllt. Somit wurden alle Farbkanäle des Druckers voll ausgeschöpft.



Abbildung 13 Drucksystem Epson EcoTank ET-2650 mit Tintentanksystem (Epson, 2020)

Beim mehrfarbigen Druck generiert das Farbmanagementsystem des Druckers aus den Kanälen C, M und Y Mischfarben entsprechend dem vom Benutzer vorgegebenen Farbprofil. Beim Druck der sensorischen Tinten darf keine Mischfarbe entstehen, da diese den einzelnen Sensor verunreinigen würde. Da der Zugriff auf den Mikrokontroller und auf die softwareseitige Steuerungseinheit verwehrt ist, mussten

die verfügbaren Parameter und Einstellungen (Rendering Intent und Eingabeprofil) experimentell identifiziert werden, um weitestgehend reine Flächendeckungen, ohne die Kontamination von weiteren Farbkanälen, zu gewährleisten. Beim Rendering Intent legt der Benutzer des Druckers fest, welche Umrechnungsmethode eines Colormanagementsystems angewendet werden soll (Wahrnehmung, Sättigung, relativ farbmétrisch, absolut farbmétrisch), wobei die Eingabeprojile (sRGB IEC61966-2-1, Adobe RGB 1998, Color Match RGB) zugewiesen sind. Hierzu wurden mittels einer Testform (Abbildung 14) (Abstufung von 10% bis 100%) zum einen Voruntersuchungen der jeweils integrierten Druckerprofile vorgenommen. Zum anderen wurden in der Druckvorstufe bei der Erzeugung eines Testcharts statt dem CMYK-Farbraum der RGB-Farbraum mit Volltonfarben gewählt. Nach dem Verdrucken der Testcharts wurden die Proben mit dem bereits oben erwähnten 3D-Laser-Scanning-Konfokalmikroskop der Baureihe VK-X auf Farbverunreinigungen untersucht (20-fache Vergrößerung). Das beste Ergebnis wurde mit dem Rendering Intent „Wahrnehmung“ in der Kombination mit dem Eingabeprofile „Adobe RGB (1998)“ erzielt. Dadurch gab der Y-Farbkanal eine 100% Flächendeckung, der Farbkanal M eine 98% (2% Y) und der Farbkanal C eine 97 % (3% Y) Flächendeckung aus. In den treiberbasierten Druckereinstellungen wurde die maximale Druckauflösung von 5.760 x 1.440 dpi gewählt, die laut Hersteller eine Tröpfchengröße von jeweils 3 pl bis 6 pl aufweist.

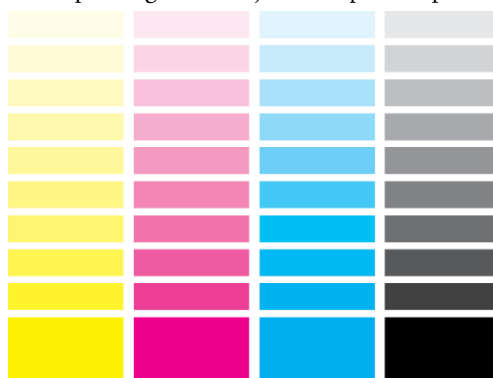


Abbildung 14 Testform CMYK Farben dargestellt in RGB Volltonfarben

### 3.4 Farbmetrische Untersuchung und Datenerhebungen

Auf den mit sensorischen Tinten verdruckten Testcharts finden aufgrund spezifischer Stimuli Farbveränderungsprozesse statt. Diese wurden mit einem Spektral-Densitometer (TECHKON SpektroDens, A507031) gemessen. Für die Messungen wurde eine (DIN ISO 13655) konforme Unterlage verwendet. Die farbmetrischen Grundlagen finden sich hierzu im *Kapitel 4.1*. Bei diesem Messverfahren wurden die Farbwerte im RGB und  $L^*a^*b^*$  Farbraum gemessen, um die stufenweise Farbveränderung der sensorischen Farben unter einem euklidischen Abstand (euklidische Distanz) zu vergleichen. Spezifischer erfolgt die Interpretation der Stimuli bedingten Farbveränderungen der sensorischen Farben zum einen gemäß der Farbabstandsformel  $\Delta E_{76}$  und zum anderem gemäß der Farbabstandsformel  $\Delta E_{00}$ . Zur vereinfachten Darstellung und zum besseren Verständnis der Ergebnisse wird das kartesische Koordinatensystem statt des üblichen polaren zugrunde gelegt. Die kumulierten durchschnittlichen Farbabstände, die im Labormaßstab referenziert wurden, dienen der Bestimmung von Grenzwerten. Diese Grenzwerte sind wichtig für die Bewertung der unterschiedlichen Reaktionsverläufe unter kontrollierten Einflussfaktoren. Zur besseren Darstellung der spektralen Veränderungen der sensorischen Farben wurden zusätzlich die Remissionskurven vor und nach einer Aktivierung der sensorischen Farben messtechnisch erfasst.

## 3.5 Lichtarten

Die Lichtarten TL84, D65 und eine UV-Lichtquelle (Abbildung 15) wurden für die Untersuchungen der photochromen Sensoren verwendet (Kapitel 4.1). Die ausgewählten Lichtarten weisen unterschiedliche UV-Anteile (grau) in ihrer spektralen Verteilung auf, die einen Einfluss auf die Belichtungscharakteristika der photochromen Sensoren haben. Außerdem wurde eine zusätzliche UV-Lichtquelle (Abbildung 16) zur Untersuchung der Lichtechtheit der hydrochromen sensorischen Farbe herangezogen (Kapitel 4.6). Die spektrale Verteilung dieser UV-Lichtquelle wurde mit einem Spektralradiometer erfasst, um ihre jeweilige Bestrahlungsstärke zu bestimmen. Zum Einsatz kam das Gerät UVpad E des Herstellers Opsytec Dr. Gröbel GmbH. Dieses kann die spektralen Bereiche UVA, UVB, UVC, VIS mittels 512 Fotodioden erfassen. Der Spektralbereich des Messgeräts reicht von 240 nm bis 480 nm  $\pm 5$  nm, wobei eine Bestrahlungsstärke von 0 bis 5000 mW/cm<sup>2</sup> oder einer Bestrahlungsdosis von 1 mJ/cm<sup>2</sup> bis 600 J/cm<sup>2</sup> aufgebracht werden kann.

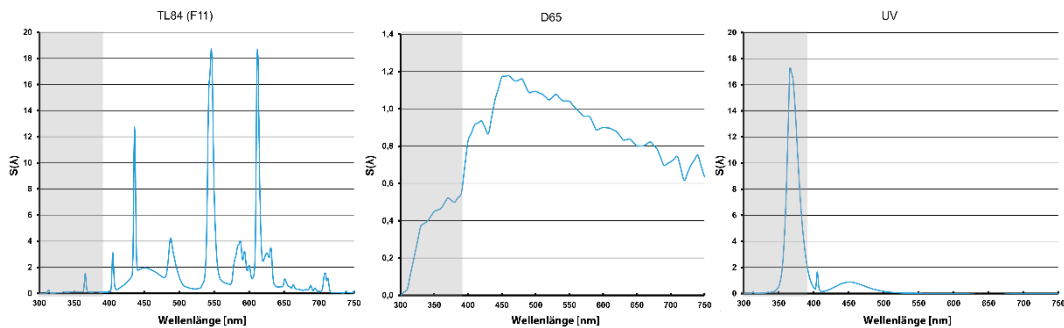


Abbildung 15 Lichtarten + UV-Lichtquelle

Die Lichtquelle TL84 (F11) hat eine Farbtemperatur von 4100 K und weist im Wellenlängenbereich von  $\lambda = 300$  nm bis 750 nm ( $1 \text{ nm} = 10^{-9} \text{ m}$ ) einen kleinen UV-Anteil bei 370 nm auf. Die maximale Intensität liegt bei etwa 370 nm. Diese Lichtquelle wird vermehrt in Fabrikhallen und Supermärkten eingesetzt. Die CIE Lichtquelle D65 (6500 K) entspricht der (DIN 5033-7). Sie basiert auf experimentellen Messungen von Tageslicht (D für Daylight) bei einem Wellenlängenbereich von  $\lambda = 300$  nm bis 750 nm (UV-VIS bis Nah-IR) mit einem UV-A-Anteil von etwa 300 nm bis 370 nm. Die ultraviolette Lichtquelle zeigt eine große Intensität bei einer Wellenlänge von etwa 365 nm.

### 3.5.1 Untersuchung der Lichtechtheit

Im Kapitel 4.6 werden zwei verschiedene Zusammensetzungen einer hydrochromen Tinte hinsichtlich ihrer Lichtechtheit untersucht. Hierbei wird die Lichtechtheit nachfolgend als Beständigkeit gegen die Einwirkung von UV-Licht ohne direkten Einfluss der Witterung bei konstanter Temperatur, Feuchtigkeit, Schichtdicke (Farbaufrag) sowie konstantem Substrat (Bedruckstoff), Lichteinfallswinkel und konstanter Lichtintensität definiert. Des Weiteren wurde die Testmethode der Norm (DIN ISO 12040) aufgrund ihrer realen oder künstlich simulierten Tageslichteinflüsse, welche das gesamte Spektrum des ultravioletten (UV) und des sichtbaren Lichts (VIS) abdecken, nicht berücksichtigt. Präferiert wurde stattdessen der kurzwellige und somit energiereichere UVA Wellenlängenbereich (315 nm – 400 nm). Zusätzlich lässt die Norm den Einflussfaktor der variierenden Witterungsbedingungen außer Acht, welche in der folgenden Untersuchung unter kontrollierten Bedingungen erfolgt. So wird darüber hinaus

die quantitative farbmimetrische Bewertung von Farbunterschieden (*Kapitel 3.4*), statt der visuell qualitativen Messung mittels Wollskala, bevorzugt. Das experimentelle Setting bestand aus streifenförmigen Proben (40 mm x 100 mm), die alle bis auf einen Teilbereich mit einer lichtundurchlässigen Schicht (Kartonschablone) bedeckt und für eine definierte Zeitdauer und in einem definierten Abstand dem UVA-Licht ausgesetzt wurde. Anschließend wurde ein weiterer Teilbereich für das Belichten freigelegt. Für die Beschichtung der Streifen wurde die Stoffzusammensetzung 1.1 (Tabelle 19) verwendet. Die Belichtung der Proben wurde mittels UV-Lichtquelle (LE Lighting EVER) mit einem Wellenlängenbereich von 315 nm bis 400 nm (UVA) und einer erhöhten maximalen Intensität bei einer Wellenlänge von etwa 370 nm belichtet (Abbildung 16). Um die Leistungsunterschiede der Lichtquelle zu definieren, wurde diese in verschiedenen Höhenabständen (Abstand der eingesetzten Messeinheit zur Lichtquelle) gemessen.

So ermittelte die Messeinheit an einer nahezu direkten Nähe zur Lichtquelle (ca. 0 cm) eine Leistung von  $0.3 \text{ mW/cm}^2$ . Diese Einstellung wurde aufgrund der höheren Bestrahlungsstärke für die UV Expositionen der Materialien verwendet.

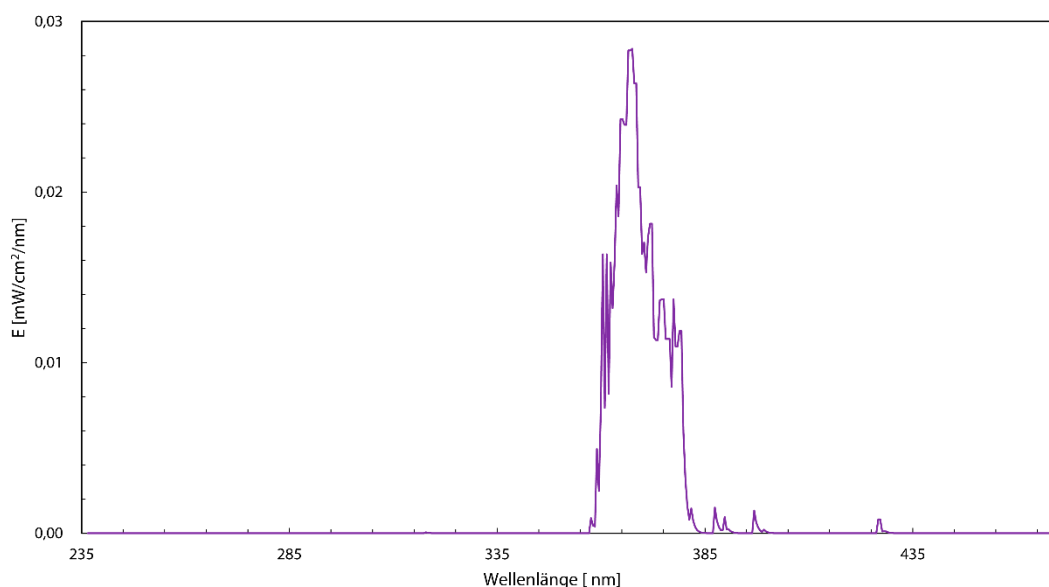


Abbildung 16 Spektrale Verteilung UV-Lichtquelle

### 3.6 Standardisierung

Alle Untersuchungen wurden unter kontrollierten Laborbedingungen durchgeführt. Um die Reproduzierbarkeit der Experimente zu gewährleisten, wurde auf ein Klimatisierungssystem zurückgegriffen, welches das Klima bei den Laborbedingungen konstant hielt. So ist eine Standardisierung des experimentellen Settings gewährleistet. Alle Einflussparameter wurden kontinuierlich aufgezeichnet und in Protokollen festgehalten. Die Temperatur während der Experimente betrug  $20 \text{ }^\circ\text{C}$  ( $\pm 1 \text{ }^\circ\text{C}$ ) und die relative Luftfeuchtigkeit betrug  $55\%$  ( $\pm 1\%$ ). Es wurden zu jeder Messung jeweils fünf Messpunkte herangezogen, aus denen ein Mittelwert bestimmt wurde.



## 3.7 Programmiertechnische Methoden

In diesem Abschnitt werden die programmiertechnischen Methoden, die für die Realisierung der zu entwickelnden Leseapplikation dienen, vorgestellt. Zudem werden die Methoden der unterstützenden Algorithmen näher behandelt.

### 3.7.1 Geräteneutrale Auswertung

Eine möglichst geräteneutrale Auswertung setzt voraus, dass geräteabhängige Einflussfaktoren der Smart Device Kameras nach einer Aufnahme durch eine nachfolgende Verarbeitung des Operating System (OS) nicht ausgeschlossen werden. Häufig sind die gleichen Kameramodule in Smartphones verschiedener Hersteller verbaut. Bei der Aufnahme desselben Motivs unterscheiden sich die Farben der wiedergegebenen Bilder aufgrund der herstellereigenen Software für die Bildwiedergabe (Alpeyev & Furukawa, 2019). Die von der Kameraeinheit gelieferten Rohdaten (durch Algorithmen unbeeinflusste Bildinformationen) der Bildaufnahmen werden von der herstellereigenen Bildverarbeitungssoftware beeinflusst. Die Einflussfaktoren auf die Rohdaten der Kamera können bspw. integrierte Algorithmen zur Farboptimierung der Aufnahmen sein. Auf diese Weise erfahren (von den Algorithmen segmentierte) Bildbestandteile, wie bspw. der blaue Himmel oder das Gras einer grünen Wiese – abweichend von den Rohdaten – eine erhöhte Farbsättigung. Eine solche Manipulation der Rohdaten ist für eine messtechnische Auswertung von Bilddaten unzulässig. Diese Einflussfaktoren würden die Analyse der durch unterschiedliche Smart Devices aufgenommenen intelligenten Codes negativ beeinträchtigen. Aus diesem Grund müssen diese Einflussfaktoren jedoch minimiert werden. Dies gilt ebenfalls für die Rohdaten der Kameras, auf die bspw. über die Kameraapplikation des OS zugegriffen und Einfluss genommen wird. Aus diesem Grund wird methodisch direkt über eine Schnittstelle auf die Rohdaten der Kameras zurückgegriffen.

### 3.7.2 Colormanagement

Zentraler Bestandteil vieler Smartphones und anderer Smart Devices sind integrierte Displays und Kameras. Je nach gerätespezifischen Eigenschaften unterscheiden sich die auf dem Display angezeigten Bilder oder das von der Kamera erfasste Bild erheblich in ihrer Farbgebung. Wenn dasselbe Bild auf verschiedenen Anzeigeräten betrachtet wird, kann ein nicht kalibrierter Bildschirm zu abweichenden Farbwahrnehmungen durch denselben Betrachter führen. Dasselbe gilt für die meisten Komponenten zur Verarbeitung von Farbinformationen, insbesondere für Kameras. So unterscheiden sich die Farben eines von der Kamera aufgenommenen Objektes häufig von Gerät zu Gerät erheblich. Die charakteristischen Farbeinstellungen der Geräte können durch eine Farbprofilierung weitestgehend neutralisiert werden, um ein originalgetreues Bild zu erhalten. Die farbgetreue Aufnahme ist ebenfalls wichtig, um die Farbgenauigkeit für die Bewertung von sensorischen Farben zu gewährleisten. Bevor das Bild eines intelligenten Codes ausgewertet werden kann, sollten die Bilder verschiedener Smart Device Kameras, wie bspw. in Smartphones mit einem Colormanagementsystem (CMS) weitestgehend neutralisiert werden, um eine möglichst einheitliche Farbgenauigkeit für verschiedene Smartphones zu erreichen.

Zur Charakterisierung der verschiedenen Kameras werden genormte ICC-Profile verwendet. Diese definieren, wie die Farbinformationen der Bilder relativ zu dessen Referenzfarbraum zu interpretieren sind. Zu diesem Zweck werden ICC-Farbprofile (International Color Consortium) verwendet, um Farbabweichungen in Smartphone Kameras anzunähern. Auf diese Weise kann ein ICC-Konvertierungsprofil erstellt werden, das Konvertierungseinstellungen zeigt, die zu einer möglichst natürlichen Bilddatei führen sollen. Die ICC-Farbprofile beschreiben den Farbraum eines Farbeingabe- oder Farbausgabegeräts (Green, 2010, S.19 ff.). Weitere Informationen können im Colormangement Workflowprozess (*Kapitel 6.6.4*) entnommen werden. Um das methodische Vorgehen zu verstehen, wird nachfolgend die Rendering Intent basierte Profile-to-Profile Transformation näher erläutert. Bei der Transformation von einem Quellfarbraum in einen Zielfarbraum (Gamut Mapping) wird die dafür notwendige Umrechnungsmethode als Rendering Intent (RI) bezeichnet. Diese Methode wurde für das Colormangement angewandt. Das RI bestimmt, was mit den Farben außerhalb des Zielfarbraumes bei einer Berechnung geschehen soll. Die bei Fotografien am häufigsten eingesetzten Umrechnungsmethoden (RIs) sind perceptiv (wahrnehmungsorientiert) und relativ farbmétrisch. Wobei die erste Methode (perceptiv) innerhalb dieser Arbeit eingesetzt wurde. So wurden seitens der ICC-Spezifikation vier unterschiedliche RIs definiert (McHugh, 2005):

0. Wahrnehmungsorientiert (rezeptiv)

In diesem RI werden alle Farben aus dem größeren Quellfarbraum in den kleineren Zielfarbraum proportional zueinander überführt, in dem sie gestaucht werden und ihre Beziehung zueinander unverändert bleibt. Dieser RI ist für die Zwecke dieser Dissertation unzureichend.

1. Relativ farbmétrisch

Bei der relativ farbmétrischen Berechnung werden diejenigen Farben, die innerhalb des Farbraums liegen, beibehalten. Die Farben, die außerhalb liegen, werden hingegen beschnitten und relativ zur nächsten verfügbaren Farbe verschoben. Dabei wird der alte Weißpunkt mit angepasst (Weißpunktanpassung), damit dieser mit dem neuen Farbraum übereinstimmt. Ohne die berücksichtigte Weißpunktanpassung wird eine leichte Farbverschiebung ausgelöst.

2. Absolut farbmétrisch

Bei der absolut farbmétrischen Berechnung bleiben, ähnlich wie bei der relativen Farbmétrik, die Farben im Farbraum erhalten, nur die Farben außerhalb des Farbraums werden beschnitten. Bei diesem Vorgehen kommt der Weißpunkt ebenfalls hinzu, der innerhalb dieser Berechnung bewahrt wird. Aus diesem Grund wird dieser RI nicht berücksichtigt.

3. Sättigungserhaltend

Bei dem letzten RI bleiben die ausgeprägten und somit intensiven Farben erhalten. Diese werden zusätzlich über eine Farbsättigungssteigerung maximiert, sodass dieser RI für den Zweck dieser Dissertation ebenfalls unzureichend ist.

### 3.7.3 Auswahl und Bewertung geeigneter Smart Devices

Für den Colormanagementprozess ist es vorgesehen, ICC-Profile der gängigsten Smartphones bereitzustellen, um die Mindestanforderungen an eine geeignete hochauflösende Kamera für die Codeanalyse zu erfüllen. Zu diesem Zweck wurden diejenigen Smartphone-Marken und -Typen ausgewählt, die auf dem Markt große Verbraucherzahlen aufweisen (Tenzer, 2020) und sich für das Colormanagement eignen (Tabelle 12):

Marke	Typ	Anzahl der effektiven Pixel	Bildsensor	Hintere Kamerablende
Apple	iPhone 6S	12.2 MP, 4032 x 3024 Pixel	CMOS	$f/2.2^b$
Apple	iPhone 7 Plus	12.2 MP, 4032 x 3024 Pixel	CMOS	$f/1.8^b, f/2.8^c$
Apple	iPhone 8 Plus	12.2 MP, 4032 x 3024 Pixel	CMOS	$f/1.8^b, f/2.8^c$
Apple	iPhone X	12.2 MP, 4032 x 3024 Pixel	CMOS	$f/1.8^b, f/2.4^c$
Apple	iPhone 11	12.2 MP, 4032 x 3024 Pixel	CMOS	$f/2.4^a, f/1.8^b, f/2.0^c$
Samsung	Galaxy S7 <sup>d</sup>	12.2 MP, 4032 x 3024 Pixel	CMOS	$f/1.7^b$
Samsung	Galaxy S8 <sup>d</sup>	12.2 MP, 4032 x 3024 Pixel	CMOS	$f/1.7^b$
Samsung	Galaxy S9 <sup>d</sup>	12.2 MP, 4032 x 3024 Pixel	CMOS	$f/2.4^b$
Huawei	Honor 6X	12.2 MP, 4032 x 3024 Pixel	CMOS	$f/0.95\text{-}f/16$
Huawei	Honor7i	13 MP, 4208 x 3120 Pixel	CMOS	$f/2.0^b$
Huawei	P20 Lite <sup>d</sup>	16 MP, 4608 x 3456 Pixel	CMOS	$f/0.95\text{-}f/16$
Huawei	P30 Pro <sup>d</sup>	40 MP <sup>b</sup> , 20 MP <sup>a</sup> , 8 MP <sup>c</sup>	CMOS	$f/2.2^a, f/1.6^b, f/3.4^c$

Tabelle 12 Vergleich verschiedener Smartphone-Marken

Legende:

a: Ultra-Weitwinkel, b: Weitwinkel, c: Teleobjektiv | d: Mindestens ausgestattet mit Android 8.0 (mit CMS) oder höher

Die in Tabelle 12 aufgeführten Smartphones wurden nachfolgend referenziert, indem ein standardisiertes Target (Colorchecker classic) des Herstellers X-Rite erfasst wurde. Dieses Target dient zur visuellen Farbbeurteilung auf Grundlage eines physikalischen Standards mit 24 Farbfeldern (Abbildung 17). Das Target wurde unter einer CIE Standardlichtart D65 aufgenommen, die einem grauen behangenen Wolkenhimmel mit einer Farbtemperatur um 6.500 Kelvin (DIN 5033-7, S. 9) entspricht. Die Lichtquelle D65 bildet die Grundlage für den Vergleich von Bildern oder Farben. Während der Aufnahme betrug der Abstand zwischen den Smartphones und dem Target 20 cm. Zu diesem Zweck wurde das Target zentral in der Lichtkabine positioniert.

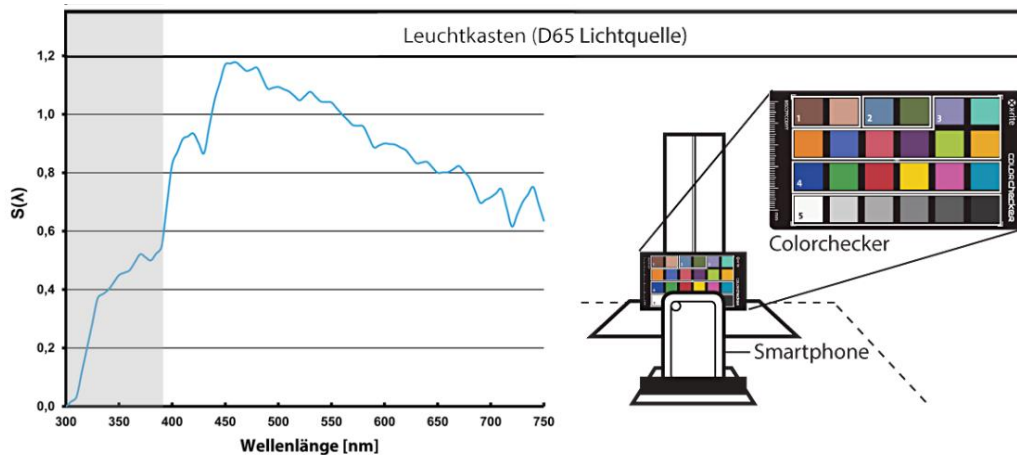


Abbildung 17 Farbfelder des Macbeth Colorchecker classic und die CIE-Normlichtart D65

Legende:

1: Hauttöne; 2: Blau- und Grüntöne; 3: Sekundärfarben; 4: RGB- und CMY-Farben; 5: Grautöne

Im nächsten Schritt wurden die mit den Smartphone Kameras aufgenommenen Testcharts verwendet, um ICC-Farbprofile für jedes Gerät zu generieren. Dies wurde gemeinsam mit Xie (2020) realisiert. Zu diesem Zweck wird in *Kapitel 6.6.4* der Vorgang des programmiertechnischen Implementierungsprozesses vorgestellt, bei dem das folgende Softwaresystem *i1 Colormanagement-Software* von X-Rite und die *ColorChecker Kamera-Profilierungssoftware* von X-Rite zum Tragen kommen.

### 3.7.4 Helligkeits- und Kontrastanpassung

Im Folgenden wird die in der programmiertechnischen Umsetzung angewandte Methode der Helligkeits- und Kontrastanpassung mittels Bildverarbeitungsoperatoren näher behandelt. Im *Kapitel 6.6.2* werden die programmiertechnischen Schritte des schnellen und effizienten Bildverarbeitungsoperators behandelt. Dieser dient der Vorverarbeitung der nachfolgend zu korrigierenden Bildinformationen. Hierzu wird auf die Funktion eines allgemeinen Punktoperators zurückgegriffen. Dieser nimmt durch eine Bildtransformation ( $T$ ) die Pixelwerte ( $i, j$ ) von Eingabebildern  $f(i, j)$  auf, manipuliert diese unabhängig von den Nachbarpixeln (Pixeltransformationen) und erzeugt ein Ausgabebild  $f'(i, j)$ . Hierbei ist jedoch der Wert jedes Ausgangspixels vom entsprechenden Wert des Eingangspixels inklusive spezieller Parameter abhängig (Crane, 1997, S. 59 ff.); (Szeliski, 2011, S. 103 ff.).

Die Funktion des Bildverarbeitungsoperators ( 2 ) kann wie folgt beschrieben werden:

$$f'(i, j) = T_{i, j}(f(i, j)) \quad (2)$$

Folgend werden den Konstanten die Verstärkungsparameter ( $\alpha > 0$ ) und Verzerrungsparameter ( $\beta$ ) hinzugefügt, welche ebenfalls den Kontrast und die Helligkeit steuern ( 3 ):

$$f'(i, j) = \alpha \cdot f(i, j) + \beta \quad (3)$$

### 3.7.5 Farbkorrektur und -transformation

Wie auch zuvor werden bei der Farbkorrektur und -transformation ebenfalls Bildverarbeitungsoperatoren verwendet. Farbkorrektur und -transformation werden mittels einer Bildtransformation durchgeführt, indem die Pixelwerte der Eingabebilder entnommen und unabhängig von den Nachbarpixeln durch hinterlegte Referenzpixelwerte korrigiert werden (Pixeltransformation). Somit wird ein korrigiertes Ausgabebild erzeugt, das nachfolgend analysiert (*Kapitel 6*) wird. Abbildung 18 veranschaulicht den Einfluss verschiedener (Norm)lichtarten (D65, TL84, A und das UV-Licht) auf den intelligenten Code. Hierbei ist ersichtlich, dass das Umgebungslicht eine wichtige Rolle bei der Codeerfassung und einer anschließenden Codeanalyse spielt. Das spezifische Farbspektrum des Lichtes kann einen erheblichen Einfluss auf die spätere Auswertung der Farbinformationen der gedruckten Sensoren haben und die Farbinformationen Analysedaten verfälschen. Zu diesem Zweck wurde der intelligente Code unter verschiedene Normlichtarten gelegt und aufgenommen.

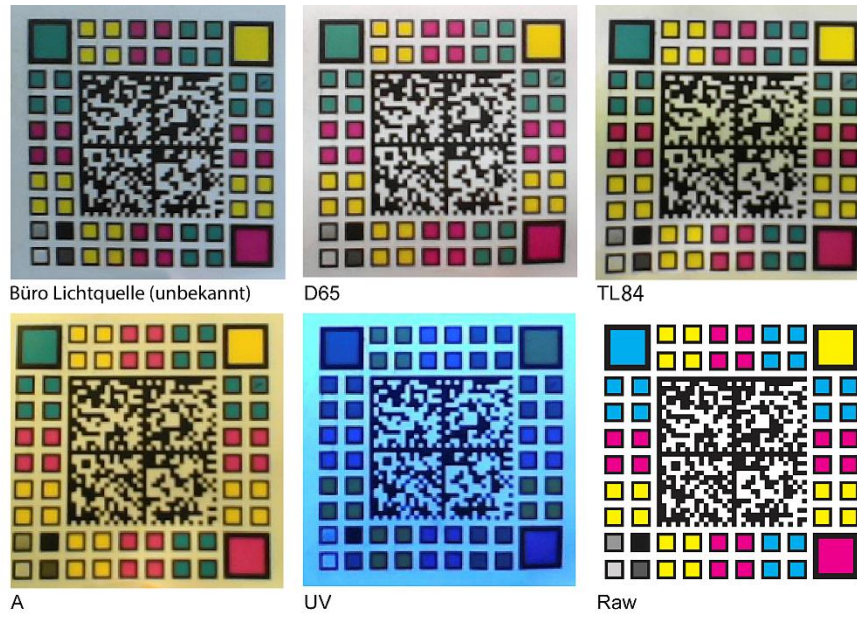


Abbildung 18 Intelligenter Code unter verschiedenen Lichtquellen

Der Effekt des Umgebungslichts wird besonders bei einem sehr gelblichen Glühbirnenlicht (A<sub>i</sub>) deutlich, das eine Farbtemperatur von ca. 2856 K aufweist und die Farbinformationen des Codes sichtlich verfälscht. Auch die TL84 Lichtart weist einen erheblichen gelben Anteil im Spektrum auf. Das Normlicht D65 eignet sich aufgrund des Tageslichtspektrums und der neutralen Färbung besser für Referenzpixelwerte und für eine Pixeltransformation. Somit können diese Informationen in den im *Kapitel 6.6.1* behandelten Algorithmus zur Farbkorrektur und -transformation integriert werden.

### 3.8 Soft- und Hardwarearchitektur

In dem vorliegenden Kapitel werden die wesentlichen Bausteine des im *Kapitel 1.3* behandelten *Konzepts* methodisch näher erörtert. Ein wesentlicher Baustein dieses Konzepts ist die entwickelte Softwarearchitektur (*Kapitel 6*). Im Folgenden ist das zu behandelnde methodische Vorgehen des zugrundeliegenden Modells ergänzend aufgeführt. In Tabelle 13 sind darüber hinaus die für die Realisierung der Softwarearchitektur notwendigen Softwarekomponenten aufgelistet, auf die im *Kapitel 6.2* näher eingegangen wird. Bei den ausgewählten Programmibliotheken handelt es sich um lizenzierte Entwickler. Vor allem die OpenCV Programmibliotheken wurden unter anderem von Intel entwickelt und auch die Py-libdmtx Programmibliothek ist vom MIT (Massachusetts Institute of Technology) lizenziert.

Die Architektur baut auf dem Client-Server-Modell (Abbildung 19) auf, das eine zentrale Netzwerktopologie in Sternstruktur aufweist. Bekannte Webanwendungen, wie unter anderem Websites, FTP (File Transfer Protocol) und E-Mail bauen auf diesem Modell auf (Kurose & Ross, 2013, S. 86). So findet sich innerhalb des Modells ein ständig aktiver Server bzw. Webserver, mit permanenter Zugriffsmöglichkeit (Host), der die Anfragen von mehreren Clients (Endgeräte) verarbeitet. Somit können mehrere Clients, mit dem vorinstallierten Browser auf dem Endgerät oder auch auf Smart Devices, gleichzeitig Inhaltsanfragen (Request) an den über die IP-Adresse zu adressierenden Webserver senden. Dies geschieht, indem ein Request mit einem Paket an die IP-Adresse des Servers gesendet wird. So wird nach einer entsprechenden Verarbeitungszeit / Antwort (Response) des Webserver auf den angeforderten Inhalt der Leseapplikation der Zugriff gewährt. Außerdem ist darauf hinzuweisen, dass bei einem Client-Server-Modell, die Clients in einer Webanwendung, in der Regel nicht direkt miteinander kommunizieren, was bei einer Peer-to-Peer Verbindung üblich ist. Der Webserver weist zudem eine feste und bekannte IP-Adresse auf.

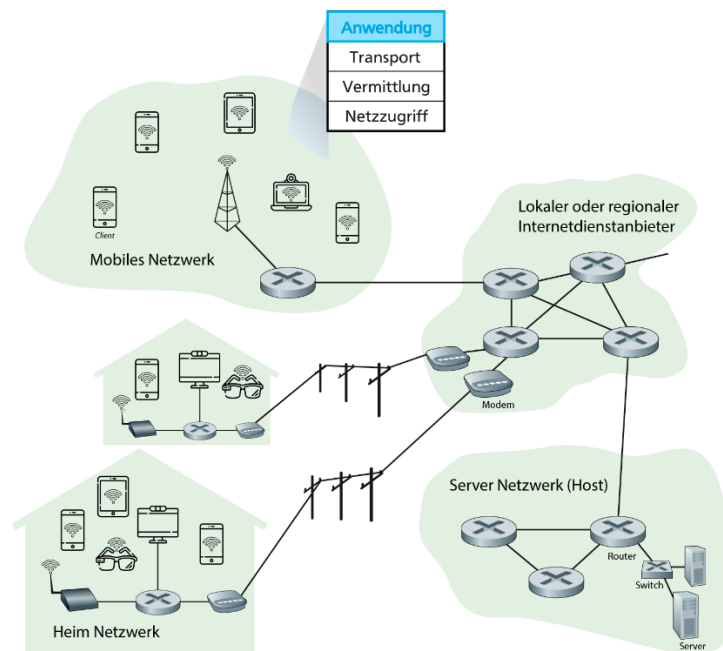


Abbildung 19 Client-Server-Architektur, angelehnt an Kurose und Ross (2013, S. 85)

Ein wesentlicher Nachteil der Sternstruktur ist es, dass bei einem Ausfall des zentralen Servers, auch die gesamte Kommunikation und der Zugriff auf den Server unterbrochen wird. Diesem kann anhand eines bereitstehenden und gespiegelten Servers (Backup) entgegengewirkt werden, indem ein Algorithmus die zyklisch wiederkehrende Verfügbarkeit des Servers überprüft.

Bezeichnung	Funktion	Kategorie
Apache	HTTP-Server	Software
WebRTC	Webbasierte-Echtzeitkommunikation	Standard
MySQL	Datenbankmanagementsystem	Datenbank
OpenCV	Algorithmen für Bildverarbeitung und Computer Vision	Programmibibliothek
OpenCV JS	Algorithmen für Bildverarbeitung und Computer Vision	Programmibibliothek
Pillow (PIL)	Python Imaging Library, Bildverarbeitungsprozesse	Programmibibliothek
Numpy	Numeric Python, Wissenschaftliche Berechnungen mit Python	Programmibibliothek
PyCMS	ICC Colormangement Interface	Programmibibliothek
Pylibdmtx	Lesen und Schreiben von DataMatrix 2D-Codes	Programmibibliothek
Scikit-image color	Farbwissenschaften Modul (Color science)	Programmibibliothek
Python	Höhere Programmiersprache	Programmiersprache
JS	JavaScript, Benutzerinteraktionen	Skriptsprache
PHP	Hypertext Preprocessor, Entwicklung dyn. Webanwendungen	Skriptsprache
HTML	Hypertext Markup Language, Entwicklung elektr. Dokumente	Auszeichnungssprache
CSS	Cascading Style Sheets, Gestaltungsanweisungen für HTML	Stylesheet-Sprache

Tabelle 13 Komponenten der Softwarearchitektur

Web Real-Time Communication (WebRTC) ist ein offener Echtzeit-Kommunikationsstandard, bei dem die Übertragung über (S)RTP ((Secure)Real-Time Transport Protocol) erfolgt. Dieser wurde vom World Wide Web Consortium (W3C) und der Internet Engineering Task Force (IETF) als ein Bündel an Kommunikationsprotokollen und API-Anwendungsprogrammierschnittstelle (Application Programming Interfaces) definiert. WebRTC basiert auf JavaScript und HTML5. So war es möglich, dass Medien (Kameras und Mikrofone) über eine Peer-to-Peer Verbindung, seitens der Browser, also mehrere Clients, miteinander zu kommunizieren. Entsprechend war es mit WebRTC das erste Mal möglich, Telekommunikation und Webentwicklung miteinander zu verbinden, so dass auf Basis einer Peer-to-Peer Kommunikation, mit anderen Browsern Videoübertragungen ermöglicht werden konnten, ohne dass dafür zwischengeschaltete Server oder herstellerspezifische Plug-Ins erforderlich sind (Loreto & Romano, 2014, S. 2). WebRTC wiederum ermöglicht auf Video-Stream-Daten und auch auf die Rohdaten der Endgerät-Kameras zurückzugreifen. Die in dieser Arbeit eigens entwickelte Webanwendung der Leseapplikation macht sich vom WebRTC Standard Gebrauch und integriert dieses in das Gesamtsystem der Soft- und Hardwarearchitektur. Der Einsatz von WebRTC hat den Hintergrund, dass vor der Einführung des Standards, ohne Plug-Ins von Drittanbietern (bspw. Adobe Flash Player) keine browserseitige Echtzeitkommunikation möglich war. Die Architektur von WebRTC basiert nahezu auf dem zuvor vorgestellten Client-Server-Modell. Das Modell wird wiederum durch eine Peer-to-Peer Verbindung ergänzt, wobei die Verbindung über eine WebRTC-API zwischen den Client-Browsern aufgebaut wird (Abbildung 20, grün). Hierzu sendet bspw. der clientseitige Browser eine HTTP-Anfrage (Hypertext Transfer Protocol) bezüglich des Inhalts, der sich auf dem adressierten Webserver befindet und der daraufhin mit einem Respons antwortet und den angeforderten Inhalt enthält (ebd., S. 1) (Abbr., rot). Loreto und Romano (2014, S. 3) verweisen dabei auf ein allgemeines WebRTC-Szenario hin, bei dem zwei clientseitige Browser, eine heruntergeladene Webanwendung ausführen, die von derselben Webseite heruntergeladen wird. Das Aufbauen und Beenden einer Kommunikation erfolgt dabei über das Standard Signalisierungsprotokoll, bspw. SIP (Session Initiation Protocol) oder Jingle (XEP-0166).

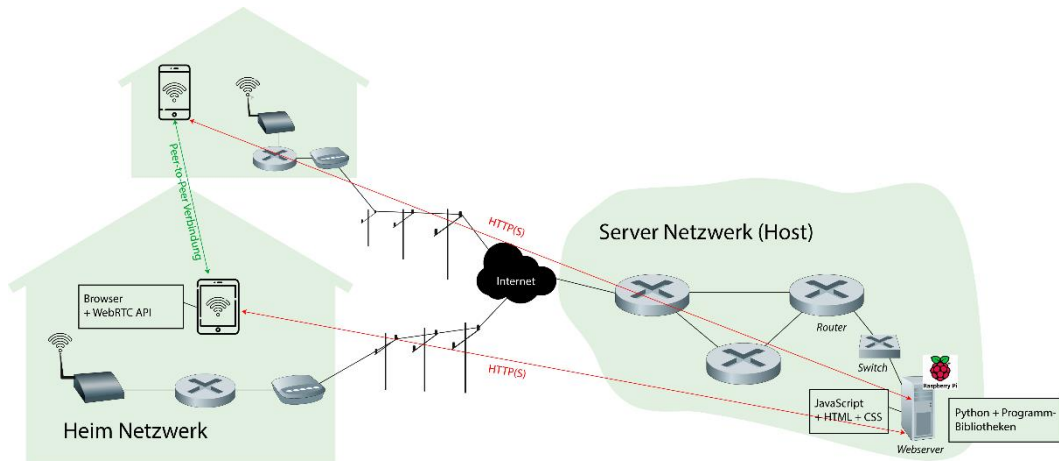


Abbildung 20 Client-Server-Model Peer-to-Peer, angelehnt an Kurose und Ross (2013, S. 85)

Die entwickelte Webanwendung der Leseapplikation (*Kapitel 6*) für das Detektieren des intelligenten Codes weist eine WebRTC Struktur auf, die HTML5, CSS und JavaScript integriert. Dabei können clientseitige Browser über eine standardisierte WebRTC-API und eine entsprechende HTTP(S)-Anfrage eine Echtzeitkommunikation aufbauen. Durch das Extrahieren einzelner Frames aus dem Datenstrom der Kameras können die Bilder (Frames) im nächsten Schritt mit Computer Vision weiterverarbeitet werden. Somit kann über die clientseitige Endgerät-Kamera auf die Leseapplikation zugegriffen werden und bei Bedarf der intelligente Code ausgelesen und analysiert werden.



## Experimentelle Umsetzung

Im Folgenden wird die experimentelle und prototypische Umsetzung der Ziele, die im *Kapitel 1.1* beschrieben wurden, erläutert. Hierbei kommt das *Konzept* von *Kapitel 1.3* zum Tragen. Innerhalb des vorgesehenen Konzepts werden die nachfolgenden sensorischen Tinten vorgestellt, welche im späteren Verlauf in Form eines intelligenten Codes mittels Inkjetdruckverfahren auf geeigneten Haftpapieren verdruckt und auf ihre Eigenschaften und Funktionalitäten hin untersucht werden. Für die Umsetzung des Konzepts werden die im *Kapitel 3* behandelten *Methoden* herangezogen. Die daraus resultierenden Ergebnisse werden vorgestellt und den im *Kapitel 2* behandelten *Forschungsstand* gegenübergestellt.

## Kapitel 4

# Sensorische Tinten

Sensorische Tinten werden unter anderem mit einem Auftragsverfahren, bspw. durch das Versprühen von feinen Aerosolen oder auch durch klassische mechanische Druckverfahren, wie dem Sieb-, Flexo-, Offset- und Tiefdruckverfahren, gefertigt. Der Fokus dieser Dissertation liegt, wie im *Kapitel 3.3* bereits erwähnt, auf digitalen Drucktechnologien wie dem kontaktlosen *Inkjetdruckverfahren*.

Unter dem Terminus *Sensoren* werden vorrangig elektronische Sensoren und Bauteile wie Kondensatoren, Leiterbahnen, Transistoren, Widerstände und weitere elektronische Komponenten verstanden. Diese basieren auf verschiedenen elektrisch leitfähigen Materialien, auf sogenannten *conductive inks* wie Carbon, Graphen, Kupfer, Silber etc. und können auf unterschiedlichen Substraten verdruckt werden. Der Fokus dieser Dissertation liegt, im Sinne der Materialforschung, ebenfalls auf der Entwicklung von funktionellen Materialien. Vielmehr werden jedoch funktionelle Materialien entwickelt, die angeregt durch einen spezifischen Stimulus aus der unmittelbaren Umwelt reagieren, in dem sie ihre chemische Struktur verändern und als Reaktion eine Farbveränderung induzieren (Bamfield & Hutchings, 2018, S. 3). Für diesen Reiz-Reaktionsprozess (*stimuli-responsive*) sind weder elektronische Komponenten, eine elektrische Leitfähigkeit noch eine externe Energieversorgung notwendig. Diese funktionellen Materialien sind auch als *chromotrope Materialien* bekannt (Fukuda, 2007, S. 1). Sie registrieren physikalische oder (bio-)chemische Einflussfaktoren, wie bspw. Wasser (*Hydrochrome*), Licht (*Photochrome*), Temperatur (*Thermochrome*), Druck (*Piezochrome*), Polarität eines Lösungsmittels (*Solvatochrome*), Anlegen einer elektrischen Spannung (*Elektrochrome*), pH-Wert (*Halochrome*), Gase (*Gaschrome*). Darüber hinaus existieren zahlreiche weitere Formen. Das Suffix „-chrome“ wird hierbei mit weiteren funktionellen Stoffnamen verbunden. Im *Kapitel 2.1* wurde bereits der Terminus *sensorische Farben* als einheitlicher Oberbegriff für die Vielzahl der verschiedenen funktionellen Materialien eingeführt. Der Begriff *sensorische Tinten* bezeichnet Tinten für den Inkjetdruck, die auf Basis von sensorischen Farben formuliert werden.

Im Kontext dieser Dissertation setzt die Verwendung der sensorischen Farben als sensorische Tinten voraus, dass diese nach einer induzierten Farbreaktion unumkehrbar in ihrem neuen farblichen Zustand verbleiben (*irreversibel*) und sich nicht in ihren ursprünglichen farblichen Zustand zurückführen lassen (*reversibel*). Somit besitzen sie die Eigenschaft, die Auswirkungen eines externen Stimulus in ihrem farblichen Zustand zu speichern und sind aufgrund dieser Eigenschaft als Sicherheitsmerkmal oder druckbare Sensorsysteme geeignet.

Für die prototypische Umsetzung eines intelligenten Codes sind geeignete sensorische Farben notwendig. Wie bereits im *Kapitel 2.1* erwähnt, gibt es einige wenige marktgängige irreversible sensorische Farben, jedoch keine, die für das Inkjetdruckverfahren geeignet sind. Dies veranlasste – im Labormaßstab – die Entwicklung von drei exemplarischen sensorischen Tinten auf Basis sensorischer Farben (Hydrochrome, Photochrome, Halochrome). Die sensorischen Farben weisen eine Irreversibilität auf und eignen sich für das verwendete kontaktlose Inkjetdruckverfahren. Die für diesen Einsatzzweck eigens entwickelten sensorischen Farben detektieren exemplarisch folgende Einflussfaktoren:

- a) Einfluss von Wasser (*Hydrochrome*)
- b) Einfluss von UVA-Licht (*Photochrome*)
- c) Einfluss saurer oder basischer Substanzen, angezeigt über den pH-Wert (*Halochrome*).

In den nachfolgenden Kapiteln werden die zuvor aufgeführten sensorischen Farben genauer betrachtet, die notwendigen Grundlagen zum Verständnis dieser vermittelt und neben der Untersuchung der Eigenschaften und der Funktionalität die notwendige Vorgehensweise für die Entwicklung sensorischen Tinten vorgestellt.

## 4.1 Grundlagen sensorischer Farben

Zum Verständnis der nachfolgenden Kapitel, insbesondere der entwickelten und eingesetzten sensorischen Farben und ihrer Komplexität im Themenfeld der Farbmittel, werden nachfolgend Grundlagen aus der analytischen, anorganischen und organischen Chemie vorausgesetzt. Außerdem werden druckverfahrenstechnische Grundlagen aus der Druck- und Medientechnologie behandelt.

## 4.2 Farbmetrische Grundlagen

Nach DIN 5033-1 (S. 4), Grundbegriffe der Farbmetrik, wird Farbe als ein „durch das Auge vermittelter Sinneseindruck, durch den sich zwei aneinandergrenzende, strukturlose Teile des Gesichtsfeldes bei einäugiger Beobachtung mit unbewegtem Auge allein unterscheiden lassen“ beschrieben. Im physikalischen Sinne ist die Farbmetrik „die Lehre von den Maßbeziehungen der Farben untereinander. Sie setzt demgemäß voraus, daß man die Farben durch Maß und Zahl eindeutig beschreiben, und dies wiederum, daß man sie messen kann.“ (Richter, 2011, S. 7)

### 4.2.1 Farbreiz

Ein Farbreiz wird als „Strahlung, die durch unmittelbare Reizung der Netzhaut eine Farberscheinung hervorruft“, beschrieben (DIN 5033-1, S. 5). Spezifischer reagieren die lichtempfindlichen Rezeptortypen der Netzhaut, die sogenannten Stäbchen, in denen das Pigment Rhodopsin eingelagert ist, und die Zapfen, in denen das Pigment Iodopsin eingelagert ist, auf einen Reiz in Form von absorbierter elektromagnetischer Strahlung (Abbildung 21) des sichtbaren Lichts (VIS) (Khorana, 1992, S. 1; Bear et al., 2018, S. 324). Das für das menschliche Auge wahrnehmbare, sichtbare Licht ist nur ein Teilbereich des elektromagnetischen Spektrums und umfasst den Wellenlängenbereich von ca.  $\lambda = 380 \text{ nm}$  bis  $780 \text{ nm}$  ( $1 \text{ nm} = 10^{-9} \text{ m}$ ). Es ist zwischen der kurzwelligen Ultravioletten Strahlung (UV) und der Infrarotstrahlung (IR) verortet (Lübbe, 2013, S. 2).

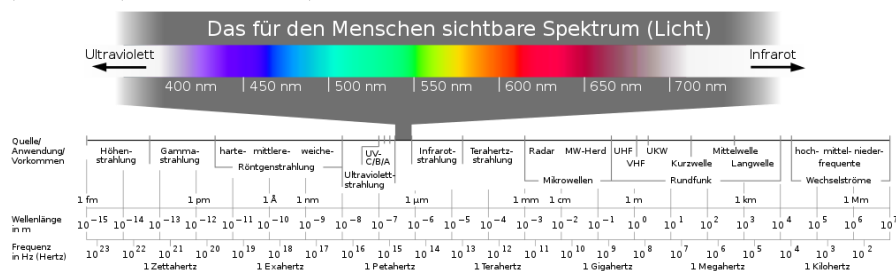


Abbildung 21 Elektromagnetisches Spektrum (Frank et al., 2005)

Die physikalisch messbare Größe eines Farbreizes ist durch die Strahlung  $\varphi(\lambda)$  des für den Menschen sichtbaren Wellenlängenbereichs zu definieren, bei der durch die Reizung der Netzhaut die Reaktion der Farbwahrnehmung angestoßen wird. Demnach wird der Farbreiz bei undurchsichtigen Körpern (Körperfarben), wie bspw. Farbfleichen, durch die nachfolgende Farbreizfunktion (4) und somit innerhalb eines skalaren Feldes durch die beleuchtende Lichtart  $S(\lambda)$  und durch den spektralen Remissionsgrad  $\beta(\lambda)$  der Oberfläche bestimmt (ebd., S.25; Richter, 1976, S. 6):

$$\varphi(\lambda) = S(\lambda) \cdot \beta(\lambda). \quad (4)$$

## 4.2.2 Farbvalenzen

Die Farbvalenz als ein integraler Bestandteil gehört zur Farbreizfunktion  $\varphi(\lambda)$ , wobei die „zahlenmäßige Kennzeichnung einer Farbvalenz [...] das primäre Ziel der Farbmotrik“ ist (Richter, 1976, S. 107). Dies geschieht durch die Multiplikation der Farbreizfunktion  $\varphi(\lambda)$  mit international festgelegten Normspektralwertfunktionen  $\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$ ,  $\bar{z}(\lambda)$ , die durch den farbmestechischen 2° und 10° Normal-Beobachter CIE (Commission Internationale de l'Éclairage) 1931 definiert wurden (DIN EN ISO/CIE 11664-1, S. 12). Diese lassen sich auf die CIE X, Y, Z Normfarbwerte übertragen (DIN EN ISO/CIE 11664-3, S. 8) (5):

$$\begin{aligned} X &= k \cdot \sum_{\lambda} \varphi_{\lambda}(\lambda) \cdot \bar{x}(\lambda) \cdot \Delta\lambda \\ Y &= k \cdot \sum_{\lambda} \varphi_{\lambda}(\lambda) \cdot \bar{y}(\lambda) \cdot \Delta\lambda \\ Z &= k \cdot \sum_{\lambda} \varphi_{\lambda}(\lambda) \cdot \bar{z}(\lambda) \cdot \Delta\lambda \\ k &= \frac{100}{\sum S(\lambda) \cdot \bar{y}(\lambda) \cdot \Delta\lambda} \end{aligned} \quad (5)$$

Die Normfarbwertanteile X, Y können als Koordinaten eines Farbortes in einer Ebene (Farbtafel) zur graphischen Darstellung der Farbarten verwendet werden (ebd.). Da die euklidischen Abstände sehr stark von den empfundenen Farbabständen abweichen und nicht beurteilt werden können, hat die CIE unter anderem einen aus dem Normvalenz-System transformierten L\*a\*b\*-Farbraum (1976) empfohlen (6). Hierbei stehen die Normvalenzen  $X_n$ ,  $Y_n$  und  $Z_n$  für die entsprechende Lichtart S( $\lambda$ ) (DIN EN ISO/CIE 11664-4, S.7; Richter, 1976, S. 224):

$$\begin{aligned} L^* &= 116 \cdot \sqrt[3]{\frac{Y}{Y_n}} - 16 \\ a^* &= 500 \cdot \left( \sqrt[3]{\frac{X}{X_n}} - \sqrt[3]{\frac{Y}{Y_n}} \right) \\ b^* &= 200 \cdot \left( \sqrt[3]{\frac{Y}{Y_n}} - \sqrt[3]{\frac{Z}{Z_n}} \right) \end{aligned} \quad (6)$$

## 4.2.3 Farbabstand

Der euklidische Farbabstand  $\Delta E_{76}$  zweier Farborte in einem L\*a\*b\* Farbraum wird durch die Farbmotrik beschrieben. Dieser Farbabstand (7) lässt sich wie folgt berechnen:

$$\Delta E_{ab}^* = \sqrt{(L_2^* - L_1^*)^2 + (a_2^* - a_1^*)^2 + (b_2^* - b_1^*)^2} \quad (7)$$

Eine Neuerung des Farbabstands ist im Jahr 2014 durch  $\Delta E_{00}$  erfolgt (8), die die Farbabstandsgenauigkeit verbessert und Berechnungsgrundlage der vorliegenden Arbeit ist (DIN EN ISO 11664-6, S. 9):

$$\Delta E_{00}^* = \sqrt{\left( \frac{\Delta L'}{k_L \cdot S_L} \right)^2 + \left( \frac{\Delta C'}{k_C \cdot S_C} \right)^2 + \left( \frac{\Delta H'}{k_H \cdot S_H} \right)^2} + R_T \cdot \frac{\Delta C'}{k_C \cdot S_C} \cdot \frac{\Delta H'}{k_H \cdot S_H} \quad (8)$$

Die Bewertung der Farbabstände  $\Delta E_{76}$  und  $\Delta E_{00}$  erfolgt durch die Bewertungsskala in Tabelle 14, welche die Wahrnehmbarkeitsschwelle widerspiegelt.

$\Delta E$ - Wert	Wahrnehmungsschwelle
< 1	Nicht wahrnehmbar
1 – 3	Schwer wahrnehmbar
3 – 6	Akzeptable Wahrnehmbarkeit
> 6	Nicht Akzeptabel

Tabelle 14 Wahrnehmungsschwellen im  $L^*a^*b^*$ -Farbraum (Bühler et al., 2018, S.16)

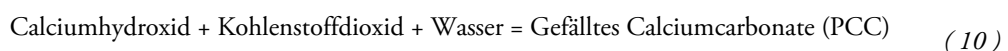
### 4.3 Substrate

Ein Substrat, insbesondere auch ein Haftpapier, kann einen Einfluss auf die chemische Verbindung der sensorischen Tinten haben und eine Wechselwirkung eingehen (Abbildung 22). Häufig ist die chemische Beschaffenheit eines Haftpapiers ein nahezu unbekannter Faktor, über die nur wenige Informationen von Seiten der Hersteller zur Verfügung stehen. Zudem ist der Einsatz von Primern problematisch. Einerseits sind fast alle Substrate für den Inkjetdruck mit Primern vorbehandelt, um die Adhäsion und Absorption der Inkjettinten zu verbessern. Andererseits kann speziell eine von einem Primer unerwünschte Wirkung auf die sensorische Tinte (bspw. auf Halochrome) ausgehen werden. In der Abbildung ist die Funktionseinbuße aufgrund der Wechselwirkung mit dem photochromen Sensor zu erkennen, bei der unmittelbar nach dem Verdrucken mit dem Substrat F (Folie beschichtet mit Primer) (Tabelle 10) eine Reaktion (Blaufärbung) stattfand und der photochrome Sensor dadurch aktiviert und unbrauchbar wurde. In der Abbildung ist exemplarisch die unmittelbare Reaktion nach dem Verdrucken auf dem Substrat F zu sehen.



Abbildung 22 Wechselwirkung zwischen Substrat (F) und Photochromen Sensor

Vor allem halochrome sensorische Farben mit pH-indikativen Funktionalitäten sollten auf ein geeignetes Substrat aufgetragen werden, denn der pH-Wert eines Papiers, Primer oder die Streichsuspension auf der Oberfläche können Einfluss auf die Funktionalität nehmen. Papier selbst ist säureempfindlich, da das chemisch saure Papier die Zellulose mit der Zeit in kleine Moleküleinheiten spaltet. Dadurch resultierend verliert das Papier an Widerstandsfähigkeit und Reißfestigkeit. Papiere weisen unterschiedliche pH-Werte auf. Diese sind oft auf die Pigmente und Füllstoffe des Papiers zurückzuführen. Streichsuspensionen für die Oberfläche des Papiers bestehen in der Regel aus weißen Pigmenten und Bindemitteln. Dabei verleihen Pigmente aus Kaolin und Kalziumkarbonat dem Papier eine geschlossene und gleichmäßige Oberfläche. Kalziumkarbonat hat einen pH-Wert von 7 bis 9 und Kaolin einen pH-Wert von 4 bis 9 (gesättigte Lösung bei 20 °C). Ausgefälltes Kalziumkarbonat wird in der Papiermasse als Füllstoff und zur Säureregulation verwendet ( 9 ), ( 10 ):



Calciumcarbonat verleiht dem Papier Weiße und Dichte, erhöht das Volumen und verbessert die Bedruckbarkeit sowie die Lichtechtheit des Papiers. Kasein oder Stärke wird üblicherweise als Bindemittel für die Haftung der Pigmente und zur Erhöhung der Wasserbeständigkeit der Beschichtung verwendet.

Unter Gestrichen versteht man einen mineralischen Oberflächenauftrag von mindestens 5 g/m<sup>2</sup> ein- oder beidseitig mit einer Streichmasse auf eine Papierbahn (DIN 6730, S. 29). Zur Kontrolle des pH-Wertes für halochrome pH-Indikatoren verwendete Papiere sollte der pH-Wert nahe dem Neutralpunkt liegen (Siebel et al., 1953, S. 101). So müssen vor der Verwendung sensorischer Tinten stets Säurebestimmungen durchgeführt werden. Säurefreie Papiere haben einen pH-Wert nahe 7 und können aus jedem Zellstoff hergestellt werden, solange der aktive saure Zellstoff bei der Verarbeitung neutralisiert wird (Bilgin, 2021, S. 7).

## 4.4 Inkjetdruck

Das Inkjetdruckverfahren wird in verschiedenen Industriezweigen eingesetzt und gehört zur Kategorie der kontaktlosen digitalen Druckverfahren (Non Impact Printing, NIP). Die im industriellen Einsatzbereich häufig verwendeten Druckverfahren (Abbildung 23) sind Continuous Inkjet (CIJ) und Drop-On-Demand (DOD). Hierbei unterteilt sich das DOD noch einmal in den thermischen DOD und piezoelektrischen DOD. Im Rahmen dieser Arbeit wird das piezoelektrische DOD-Verfahren verwendet, bei dem durch einen entsprechenden Druckimpuls bedarfsgerecht Tropfen, im Volumenbereich von ca. 10 µm bis 100 µm ausgestoßen werden (Hoath, 2016, S.1). Das Druckprinzip basiert darauf, dass eine Inkjettinte auf Wasser- oder Lösungsmittelbasis zunächst durch Unterdruck vor dem Auslaufen gehindert wird. Dabei wird der piezoelektrische Effekt genutzt, bei dem ein spannungsabhängiger Piezoaktuator, der sich unter elektrischer Spannung verformen kann und so die Tinte durch die jeweilige Düse impulsartig herausdrückt.

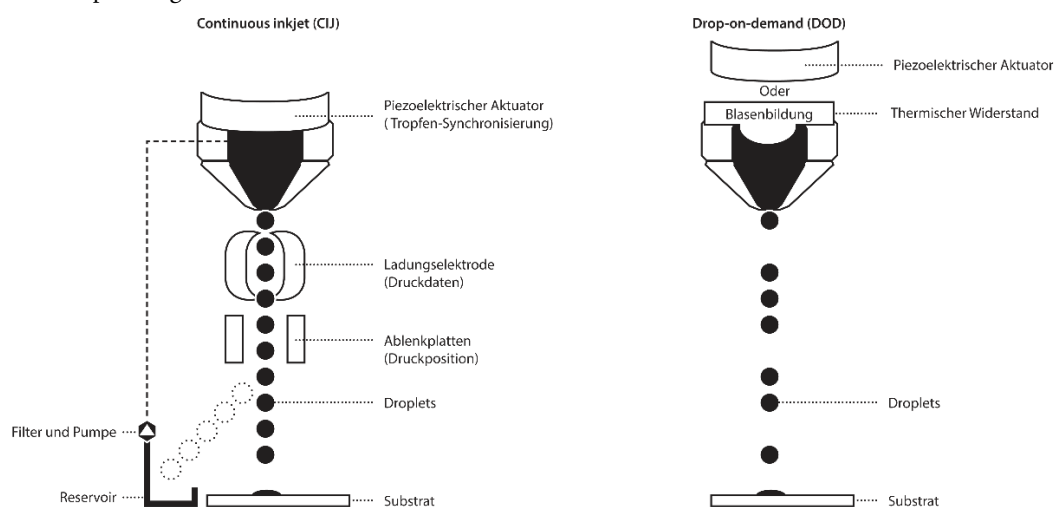


Abbildung 23 Inkjetdruckverfahren (Varianten)

Für die Entwicklung einer industriellen inkjetfähigen Tinte, die eine dauerhaft stabile Tinte für das kontaktlose Druckverfahren darstellt, müssen mehrere Faktoren berücksichtigt werden. Störende zeitliche Veränderungen, wie unter anderem Partikelablagerung, Agglomeration, Phasentrennung und andere negative Faktoren, müssen dabei längerfristig kontrolliert werden (Zapka, 2018, S. 25). So sind bei der Entwicklung von Inkjettinten unter anderem Faktoren wie Viskosität, Oberflächenspannung, pH-Wert, Farbstoff-/ Pigmentkonzentration und Schaumbildung zu berücksichtigen. Dabei sind nicht alle Faktoren gleichermaßen für das eingesetzte Inkjetdruckverfahren relevant. Aus diesem Grund liegt der Fokus hier auf den Faktoren Löslichkeit, Partikelgrößenverteilung, Viskosität und Oberflächenspannung.

#### 4.4.1 Grundlagen der Viskosität

Laut Hutchings und Martin (2013, S. 23) bestimmen die zwei physikalischen Eigenschaften Oberflächenspannung und Viskosität das Verhalten der beim Tintenstrahl Druck entstehenden Strahlen und Tropfen. Die Viskosität beschreibt dabei das Fließverhalten eines Stoffes. Zapka (2018, S.33) beschreibt, dass die Mehrheit der gängigen Inkjet Tinten ein newtonsches Verhalten zeigt und somit ein linear viskoses Fließverhalten aufweist. Die Stoffdaten der zu entwickelnden Inkjet Tinten können durch den Einsatz eines Rheometers ermittelt und iterativ angepasst werden, um eine Druckfähigkeit der Tinten zu erzielen. Insbesondere ist eine Untersuchung der rheologischen Eigenschaften sinnvoll, da die Inkjet Tinte während des Druckvorgangs hochfrequenten Druckschwankungen im Kanal und hoher Scherspannung an der Düsenwand ausgesetzt ist und diesen standhalten muss (Hoath, 2016, S. 341). Die Scherbewegung bildet eine Grundlage für die Definition der Viskosität. Durch ein Zwei-Platten-Modell lässt sich dabei das viskose Fließverhalten einer Tinte vereinfacht beschreiben (Abbildung 24). Hierbei besteht das Modell aus einer unteren statischen ( $v = 0$ ) und einer oberen konstant bewegenden dynamischen Platte (A). Auf letztere wirkt die Scherkraft ( $F$ ) resultierende Geschwindigkeit ( $v$ ) ein, die zur Mitte hin abnimmt. Im Abstand ( $h$ ) zwischen den beiden Flächen wird ein Flüssigkeitsbereich (bspw. Tinte) definiert. Die Flüssigkeit wird geschert und weist an beiden Platten eine Wandhaftung auf, bei der eine laminare Fließbedingung (Schichtenströmung) herrscht (Mezger, 2016, S. 23). Dabei findet ein linearer Geschwindigkeitsabfall statt, der senkrecht zur oberen und unteren Oberfläche verläuft und durch die Flüssigkeit erzeugt wird. Diese ist durch die folgende Schergeschwindigkeit  $\dot{\gamma}$  gegeben ( 11 ):

$$\dot{\gamma} = \frac{v}{h} \quad (11)$$

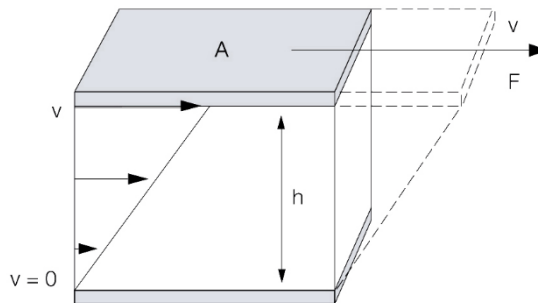


Abbildung 24 Zwei-Platten-Modell ( Mezger, 2016, S. 23)

Ein Aspekt bei der Tintencharakterisierung ist die lineare Beziehung einer newtonschen Flüssigkeit. Dabei ist die Scherspannung  $\tau$  ( 12 ) proportional zur Schergeschwindigkeit  $\dot{\gamma}$ . Dies wird in der folgenden Gleichung beschrieben, bei der  $\eta$  für die dynamische Viskosität steht:

$$\tau = \eta \cdot \dot{\gamma} \quad (12)$$

Ist die Viskosität  $\eta$  unabhängig von der Schergeschwindigkeit  $\dot{\gamma}$ , wird dieses Verhalten als linear oder newtonsches Verhalten bezeichnet (Hoath, 2016, S. 7). Die SI-Einheit für die dynamische Viskosität ist die Pascal-Sekunde (Pa·s). Wasser weist eine newtonsche Viskosität (idealviskos) auf und weist bei einer Temperatur von ca. 20 °C eine Viskosität von etwa 1 mPa·s auf. Ein erheblicher Anteil von wasserbasierten Tinten weist ein newtonsches Verhalten, ähnlich zu Wasser, auf (Hoath, 2016, S. 342). Das methodische Vorgehen zum Ermitteln der rheologischen Eigenschaften der sensorischen Tinten kann in Kapitel 3.2.1 eingesehen werden.



Pigmente können in einer Tintenmatrix aufgrund von physikalischen Wechselwirkungen in der Strömung (eines Fluids mit zwei Phasen) zu einer Abweichung des newtonschen Verhaltens der Tinte führen. Insbesondere die Wechselwirkung und Konzentration von unterschiedlichen Partikelgrößen haben einen erheblichen Einfluss auf die Viskosität, welche eine Hauptursache für Energieverluste darstellen (Hoath, 2016, S. 341). Die Zugabe von Pigmenten (Partikeln) in eine Tintenmatrix führt zu einer Viskositätssteigerung und stellt einen nützlichen Anhaltspunkt für eine Viskositätssteigerung dar. Die Viskositätssteigerung kann in erster Näherung mit Hilfe des Einstein'schen Viskositätsgesetzes wie folgt ermittelt werden ( 13 ):

$$\eta = \eta_0(1 + 2,5\phi) \quad (13)$$

Hierbei wird  $\eta_0$  durch die Viskosität des Grundlösungsmittels und  $\phi$  durch den Volumenanteil der zugesetzten Partikel beschrieben. Die Einsteingleichung gilt für niedrige Konzentrationen nicht wechselwirkender Kugeln in newtonschen Lösungsmitteln (ebd., S. 343). Kräfte entstehen aus den Wechselwirkungen zwischen den Molekülen der Flüssigkeit und wirken zwischen Flüssigkeitsbereichen, die sich relativ zueinander bewegen (Hutchings & Martin, 2013, S. 24).

#### 4.4.2 Grundlagen der Oberflächenspannung

Engel und Reid (2006, S. 216) beschreiben die Oberflächenspannung mit der Form eines runden Flüssigkeitstropfens in der Schwerelosigkeit, bei der die Geometrie des Tropfens von „der maximale[n] Anzahl von Molekülen von benachbarten Molekülen umgeben ist“. Dabei ist die Wechselwirkung der Moleküle (Isotropie) in einer homogenen Phase (Flüssigkeit) so anziehend, dass die energetisch günstigere Form der größeren Oberfläche eines kugelförmigen Tropfens mit konstantem Volumen geschaffen wird. Die physikalische Eigenschaft der dabei aufzuwendenden Energie der daraus resultierenden Oberflächenspannung ( $\sigma$ ) lässt sich dabei durch die nachfolgende thermodynamische Definition der Grenzflächen beschreiben ( 14 ):

$$\sigma = \frac{\Delta E_{GF}}{\Delta A} = \frac{\Delta W_{GF}}{\Delta A}, \quad [\sigma] = \frac{N}{m} \quad (14)$$

Hierbei steht  $\Delta E_{GF}$  für die Zunahme der Grenzflächenenergie im Verhältnis zur neugeschaffenen Grenzfläche  $\Delta A$ . Ein Grenzflächenmolekül  $\Delta W_{GF}$  hat dabei den Betrag der aufzuwendenden Energie (Binnering, 2016). Die SI-Einheit für die Oberflächenspannung ist dabei Newton pro Meter (N/m).

Die für die Ermittlung der dynamischen Oberflächenspannung eingesetzte Blasendruckmethode kann im *Kapitel 3.2.2* eingesehen werden. Hier findet sich auch die Young-Laplace-Gleichung. Diese beschreibt den Zusammenhang zwischen der dynamischen Oberflächenspannung, dem Druck und der Oberflächenkrümmung einer kugelförmigen Gasblase in einer Flüssigkeit.

#### 4.4.3 Reynolds-, Weber- und Ohnesorge-Zahl

Nach Hoath (2016, S. 8) sind die Oberflächenspannung, Trägheit und Viskosität von entscheidender Bedeutung bei der Bildung und dem Verhalten von Flüssigkeitsstrahlen und -tropfen. So können „zur Charakterisierung ihrer zwei dimensionslose Kennzahlen die Reynolds- und die Weber-Zahlen verwendet werden“ (ebd.). Dabei stellt die Reynolds-Zahl (Re) das Verhältnis zwischen Trägheits- und Zähigkeitskräften in einem bewegten Fluid dar und ist durch die folgende Formel definiert ( 15 ):

$$Re = \frac{\rho \cdot v \cdot d}{\eta} \quad (15)$$

Hierbei beschreibt  $\rho$  die Dichte,  $v$  eine Strömungsgeschwindigkeit,  $\eta$  die Viskosität des Fluids und  $d$  die charakteristische Länge (typischerweise der Durchmesser des Strahls, der Düse oder des Tropfens). Als typische Düsenaustrittsgeschwindigkeit wird eine Spannweite von 10 m/s bis 30 m/s angegeben (Hutchings & Martin, 2013, S. 27; Cummins & Desmulliez, 2012, S. 198). Hoath (2016, S. 83) wiederum gibt einen variierenden Wertebereich von 4 m/s bis 11 m/s an. Innerhalb dieser Dissertation wird der Micro-Piezo-Druckkopf von Epson eingesetzt, bei dem Tintentropfen mit einer Frequenz von 10 kHz mit einer Tropfengröße von etwa 5 pl und einer Geschwindigkeit ( $v$ ) von ca. 5 m/s ausgestoßen werden (Okumura & Takahashi, 2007, S. 317). In Abbildung 25 ist der Micro-Piezo-Druckkopf (ca. 33 mm x 11 mm) samt Tintenflusskanäle und Ausschnitt der exemplarischen Düsenflusskanäle (ca. 161  $\mu\text{m}$  x 118  $\mu\text{m}$ ) zu erkennen. In der Veröffentlichung von Okumura und Takahashi (2007, S. 316) wird im professionellen Segment ein kleinerer Düsendurchmesser ( $d$ ) von ca. 19  $\mu\text{m}$  angegeben. Der messtechnisch ermittelte Düsendurchmesser des eingesetzten nicht professionellen Drucksystems (*Kapitel 3.3*) ist in Abbildung 26 dargestellt und beträgt ca. 61  $\mu\text{m}$ .

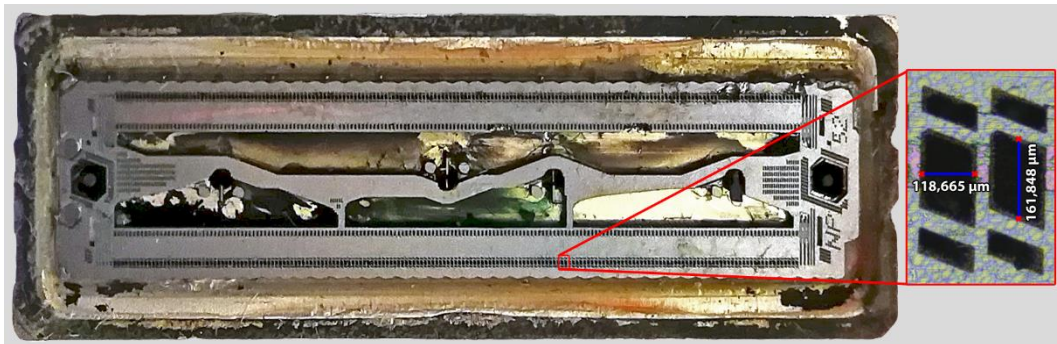


Abbildung 25 Micro-Piezo-Druckkopf und Tintenflusskanäle

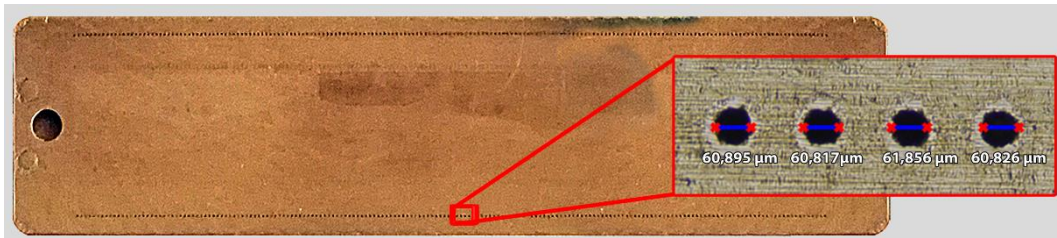


Abbildung 26 Piezodüsenplatte samt Düsenausschnitt

Die Weber-Zahl ( $We$ ) hingegen hängt vom Verhältnis zwischen Trägheit und Oberflächenspannung ab, bei der  $\sigma$  für die Oberflächenspannung steht. Die Formel ( 16 ) ist insbesondere wichtig für die Bildung eines kugelförmigen Tropfens (ebd.):

$$We = \frac{\rho \cdot v^2 \cdot d}{\sigma} \quad (16)$$

Der Einfluss der Geschwindigkeit auf die dimensionslosen Kennzahlen kann durch die Bildung einer weiteren Kennzahl, die durch die Ohnesorge-Zahl ( $Oh$ ) definiert ist, aufgelöst werden (ebd.) ( 17 ):

$$Oh = \frac{\sqrt{We}}{Re} = \frac{\eta}{\sqrt{\sigma \cdot \rho \cdot d}} \quad (17)$$

Die Ohnesorge-Zahl ist eine Kennzahl zur Charakterisierung des Zerfalls von Flüssigkeitsstrahlen bzw. zur Beurteilung von Tropfendehformationen. Bei einer hohen Ohnesorge-Zahl ( $Oh > \sim 1$ ) verhindern viskositätsbedingte Kräfte das Generieren eines Tropfens, während bei einer zu niedrigen Ohnesorge-Zahl ( $Oh < \sim 0.1$ ) der Tintenstrahl vermehrt Satellitentropfen bildet. So fasst Hoath (2016, S. 8 f.) zusammen, dass „eine zufriedenstellende Leistung einer Flüssigkeit beim DOD-Inkjetdruckverfahren eine geeignete Kombination physikalischer Eigenschaften, die auch von der Tröpfchengröße und -geschwindigkeit abhängen“, erfordert. Darüber hinaus wird darauf hingewiesen, dass die Ohnesorge-Zahl einige Einflussfaktoren wie bspw. Antriebsbedingungen oder die jeweils notwendige kinetische Energie, um einen Strahl aus der Düse auszusondern, oder die Partikelgrößenverteilung einschließlich von Geometrieschwankungen, nicht berücksichtigt (Hoath, 2016, S. 9).

Mithilfe der Re- und Oh-Zahl lässt sich ein Zielparameterfenster für bestimmte Bandbreiten, für eine stabile Verdruckbarkeit abschätzen (Abbildung 27). Anhand dem eingesetzten Drucksystem nahekommenden Parametern, wie Düsendurchmesser ( $d$ ), Tropfengeschwindigkeit ( $v$ ), der im *Kapitel 4.7* ermittelten durchschnittlichen Viskosität ( $\eta$ ) und Oberflächenspannung ( $\sigma$ ), lassen sich die Re- und Oh-Zahlen abschätzen. Im Anhang (B) finden sich hierzu Berechnungen der entwickelten sensorischen Tinten, dessen Resultate in der Abbildung mit farbigen Punkten eingezeichnet sind und auf eine Wahrscheinlichkeit der vermehrten Satellitentropfenbildung, jedoch immer noch nahe dem druckbaren Bereich, hinweisen.

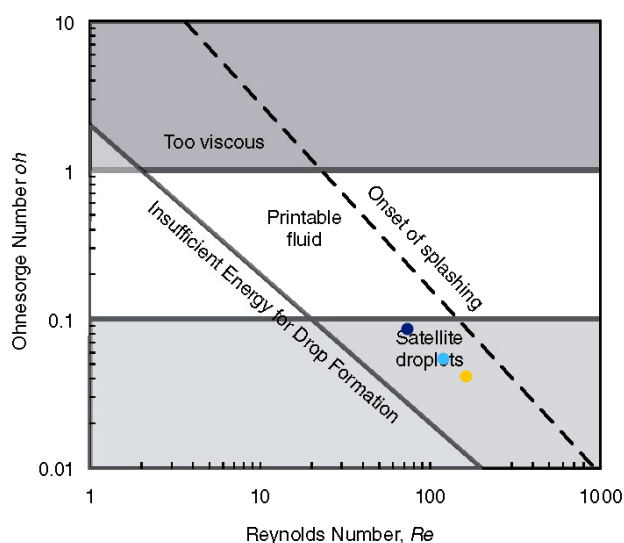


Abbildung 27 Zielparameterfenster für den stabilen Betrieb des DOD-Inkjetdruck (Hoath, 2016, S. 9)

Legende:

Dunkelblauer Punkt (photochrom), hellblauer Punkt (hydrochrom), gelber Punkt (halochrom)

In *Kapitel 4.7* wurde die Oberflächenspannung, Viskosität und Partikelgrößenverteilung (Löslichkeit) unter der Entwicklung eines Zielparameterfensters auf Basis der für das eingesetzte Inkjetdruckverfahren entwickelten sensorischen Tinten angepasst und verifiziert. Es konnte einerseits sichergestellt werden, dass durch die dabei angewandte Vorgehensweise ein annäherndes Zielparameterfenster für die Verdruckbarkeit entwickelt wurde. Andererseits konnte sichergestellt werden, dass die sensorischen Tinten dem DOD-Inkjetdruckverfahren entsprechen.

So ist es das Ziel dieser vorliegenden Arbeit, prototypische Inkjettinten als sensorische Tinten zu entwickeln. Um den Funktionsumfang der sensorischen Tinten zu analysieren. Hierzu, wurden Additive aus

der Tabelle 9, soweit es für die Verdruckbarkeit notwendig war, eingesetzt. Dies hat den Hintergrund, dass Additive wie Harze und Polymere eingesetzt werden, um bestimmte Komponenten innerhalb der Tinte (z. B. Pigmente, ungelöste Feststoffe) zu binden. Diese Additive beeinflussen unter anderem die Viskosität der Flüssigkeit und auch die Elastizität aufgrund ihres potentiellen polymeren Charakters (ebd. S. 341).

## 4.5 Chemische Grundlagen

Das folgende Kapitel setzt sich mit den notwendigen chemischen Grundlagen der sensorischen Farben auseinander und gibt einen ersten Überblick über die chemisch-physikalischen Mechanismen der entwickelten sensorischen Farben.

### 4.5.1 Anregung von Atomen

Atome bestehen aus einem Kern – nahezu demselben Anteil an positiv geladenen Protonen ( $p^+$ ) und (neutralen) Neutronen ( $n$ ) sowie einer Hülle aus Elektronen ( $e^-$ ). Die Anzahl der Elektronen hängt dabei von der Anzahl von Protonen ab, wodurch ein Atom ungeladen und somit ausgeglichen ist. Dabei gilt, dass das Atom durch die Anzahl der Protonen bestimmt werden kann (Ordnungszahl im Periodensystem). Bei chemischen Reaktionen innerhalb eines Moleküls erfolgen die Veränderungen an der Atomhülle, jedoch nicht am Kern selbst. Dabei werden die Abstände zwischen den Bahnen, auf denen sich die Elektronen bewegen, als Energieniveaus (Energiezustände) bezeichnet. Der Grundzustand entspricht dem geringstmöglichen Energiezustand und ist am stabilsten. Ein Gesamtsystem möchte dabei immer den stabilsten Grundzustand erhalten oder erreichen. Die Elektronenbewegung auf den Bahnen ist strahlungsfrei. Bei einem Bahnwechsel eines Energiezustands näher zum Kern, gibt ein Elektron Energie ab (Emission), entgegengesetzt wird beim Entfernen vom Kern Energie aufgenommen. Anzumerken ist, dass die Energie auch in Form von Licht, das absorbiert wird, zugeführt werden kann. So kann in einigen Fällen einerseits ein Ungleichgewicht vorherrschen, bei dem mehr Elektronen als Protonen vorhanden sind, weshalb das Atom schließlich negativ (Elektronenüberschuss) oder positiv geladen (Elektronenmangel) ist. Bei einem angeregten Zustand wechselt das Elektron (Quantensprung) in einen höheren Zustand ( $n > 1$ ) zwischen den Bahnen (unterschiedliche Energieniveaus). Beim Zurückkehren des Elektrons in seinen Grundzustand ( $n = 1$ ) wird die Energiedifferenz in Form von Photonen (Emission) einer bestimmten Wellenlänge ausgestrahlt.

Die Frequenz der Strahlung hängt mit der Energiedifferenz zwischen dem Ausgangs- und Endzustand zusammen. Dabei ergibt das Planck'sche Wirkungsquantum ( $h$ ) mit der Frequenz ( $f$ ) die Energiedifferenz ( $\Delta E_e$ ) ( 18 ):

$$h \cdot f = \Delta E_e \quad (18)$$

Für Photonen steht nach der Plank-Einstein'schen Beziehung (Welle-Teilchen-Dualismus) für  $E$  für Energie und  $\nu$  für die Frequenz der elektromagnetischen Strahlung ( 19 ). Die Planck'sches Wirkungsquantum  $h$  beträgt dabei  $4,135\ 6692\ (12)\ 10^{-15}$  eV und  $c$  für Lichtgeschwindigkeit im Vakuum beträgt dabei  $299,792,458\ \frac{m}{s}$  (Hädener & Kaufmann, 2006, S. 187):

$$E = h \cdot \nu = h \cdot \frac{c}{\lambda} \quad (19)$$

## 4.5.2 Orbitalmodell

Das Orbitalmodell beschreibt, wie sich Elektronen um ein Moleküllern in verschiedenen dreidimensionalen Räumen, den Orbitalen bewegen und wo sie sich nach der Heisenberg'schen Unschärferelation mit hoher Wahrscheinlichkeit verorten. Die Darstellung der Orbitale wird dabei aus der Schrödinger Gleichung abgeleitet, aus der die Form des Orbitals für die Elektronen bestimmt werden kann. Die Formel der Nebenquantenzahl ( $l = n - 1$ ) beschreibt dabei die Formen der Aufenthaltsbereiche der Elektronen in den einzelnen Orbitalen (s, p, d, f, g). Bspw. weist ein  $1s^1$ -Orbital eine kugelförmige Form auf und kann maximal  $2n^2$ -Elektronen aufnehmen. Die räumliche Anordnung wird durch die magnetische Quantenzahl  $m_l$  ( $\pm 1$ ) und die Spinquantenzahl  $m_s$  ( $+\frac{1}{2}$  oder  $-\frac{1}{2}$ ) definiert (Abbildung 28) (Lechner, 2017, S. 70).

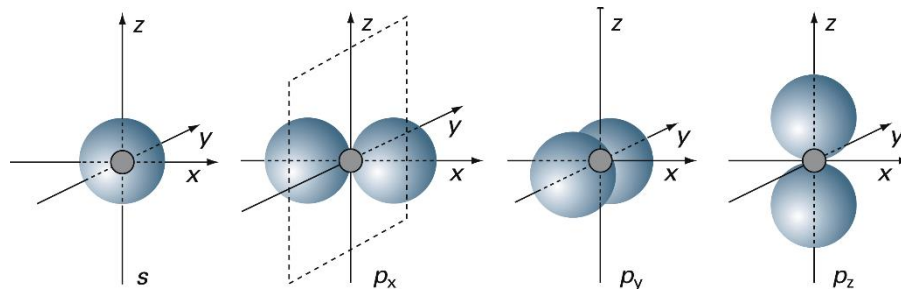


Abbildung 28 Orbitalmodell (Hädener und Kaufmann, 2006, S. 5)

Die Gesamtübersicht der Quantenzahlen kann aus der Tabelle 15 entnommen werden. Die Besetzung der Schalen (Abbildung 29) folgt dabei zum einen nach dem Pauli-Prinzip, bei dem die Elektronen nicht in allen Quantenzahlen übereinstimmen dürfen und jede Schale eines Elements maximal mit  $2n^2$  Elektronen besetzt werden kann. Zum anderen werden Orbitale nach dem Hund'schen Prinzip mit gleicher Energie nacheinander mit Elektronen besetzt. Dies bedeutet, dass in einem Element, die Elektronen die Zustände nach der energetischen Reihenfolge der Orbitale periodenweise nach einem Aufbauprinzip besetzen (Lechner, 2017, S. 36).

Schalen	Unterschale	Orbitale	Bezeichnung	e <sup>-</sup> Anzahl
1	0	0	1s	2
2	0	0	2s	2
	1	-1, 0, +1	2p	6
3	0	0	3s	2
	1	-1, 0, +1	3p	6
	2	-2, -1, 0, +1, +2	3d	10
...	...	...	...	...

Tabelle 15 Quantenzahlen in der Übersicht

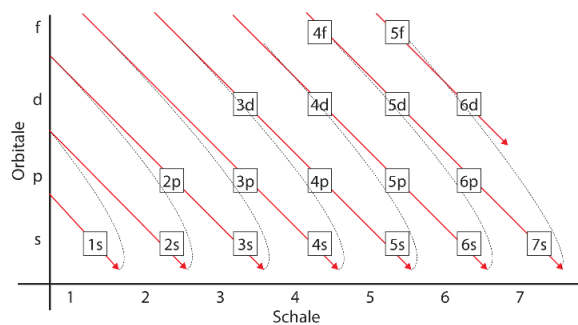


Abbildung 29 Aufbauregel

### 4.5.3 Anregung von Molekülen

Innerhalb der Moleküle kann es zwischen den Valenzelektronen der Atome zu Wechselwirkungen kommen. Dieser Aspekt ist besonders relevant für Farbstoffmoleküle. Eine Bindung zwischen zwei Atomen entsteht dort, wo sich zwei Orbitale überlappen und als kovalente Bindungen bezeichnet werden. Bei  $\sigma$ -Bindungen überlappen sich die Orbitale rotationssymmetrisch um die Bindungsachse. Dies ist bspw. bei der Überlappung von zwei s-s oder s-p Orbitalbindungen der Fall, in seltenen Fällen können jedoch auch sich überlappende p-p Orbitalbindungen vorkommen. Hier sind die  $\sigma$ -Bindungen stärker und stabiler, jedoch weniger reaktiv, und ihre Elektronen sind lokalisierbar. Bei einfachen  $\sigma$ -Bindungen ist die benötigte Energie (kurzwelliger UV-Bereich), um die entsprechenden  $\sigma$ -Elektronen auf ein höheres energetisches Niveau anzuheben, sehr viel größer. So reicht die Niveauanhebung durch das sichtbare elektromagnetische Spektrum (UV/VIS) nicht aus. Bei den sogenannten  $\pi$ -Bindungen überlappen sich die Orbitale nicht rotationssymmetrisch. Dies ist bspw. bei der Überlappung von zwei p-p Orbitalen zu erkennen. Hier sind die  $\pi$ -Bindungen schwächer und instabiler, jedoch ist das System reaktiver und ihre Elektronen delokalisiert, d.h. innerhalb der Struktur beweglich. Bei einer  $\pi$ -Bindung in ungesättigten Bindungen (z. B. C=C Bindungen) mit mindestens einer Mehrfachbindung ist die benötigte Energie, um die  $\pi$ -Elektronen auf ein energetisch höheres Niveau anzuheben, sehr viel geringer. So kann bereits Licht im langwelligen UV-Bereich absorbiert werden. Nach dem Valenz-Bindungs-Modell (Heitler & London, 1927) wird der Farbigkeit einer chemischen Struktur eine konjugierte Doppelbindung mit freien  $\pi$ -Elektronen vorausgesetzt, die leichter in ein höheres Energieniveau angeregt werden. Bei einem stetig ausgedehnten  $\pi$ -Elektronensystem mit konjugierten Doppelbindungen, können sich die delokalisierten Elektronen durch Mesomerie (Resonanz) freier bewegen und benötigen dadurch weniger Energie zwischen dem Grund- und Anregungszustand. Die folgende Abbildung 30 zeigt die Molekülorbitale und ihre charakteristischen Elektronenübergänge:

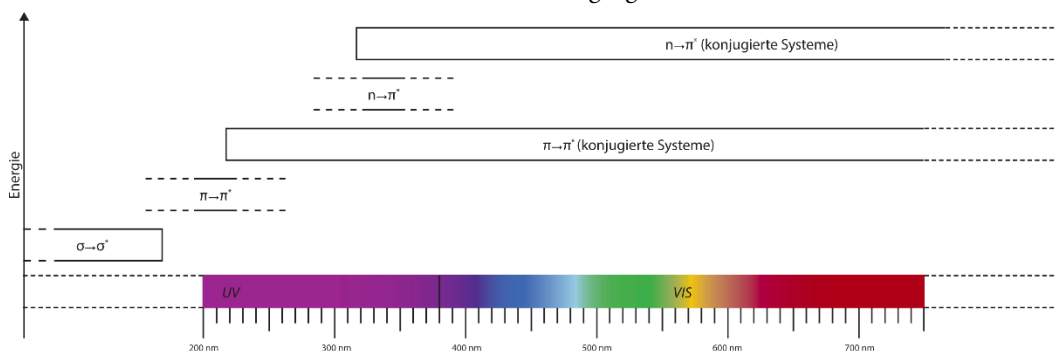


Abbildung 30 Molekülorbitale und Elektronenübergänge

Chromophore (Witt, 1876) – vom alt. griech. χρῶμα (chroma) für Farbe und φερός (phorós) für tragend (Gottwein, 2020) – sind der farbgebende Teil von Farbstoffmolekülen. Diese lassen sich in anorganische und organische Chromophore einteilen. Erstere beruhen unter anderem auf den Charge-Transfer-Übergängen, bei denen eine Wechselwirkung zwischen Elektronendonatoren und Akzeptor stattfinden. Hierbei wird durch die Absorption des Lichts ein Übergangsmetallkomplex ausgelöst, durch den ein Elektron zwischen Liganden und oder zwischen Metallen übertragen wird (Klöckl, 2015, S. 63). Hierzu findet sich das Beispiel vom Charge Transfer in Preußisch Blau (*Kapitel 4.6.2*). Zweitere beruhen, wie bspw. das Diferuloylmethan (*Kapitel 4.6.3*), auf einem System konjugierter Doppelbindungen oder auf mehreren aromatischen Ringen mit mesomeren Bereichen. Triphenylmethanfarbstoffe wie bspw. Patentblau V können durch die Einführung elektronenliefernder Substituenten (Auxochrome) an mindestens zwei der aromatischen Ringe ein chinoides System als Chromophor bilden, das die Farbgebung der Verbindungen bestimmt (*Kapitel 4.6.1*).

#### *Auxochrome*

Auxochrome werden als Substituenten (-NR<sub>2</sub>, -NH<sub>2</sub>, -OCH<sub>3</sub>, -Halogen, -OR) bezeichnet, welche mit freien Elektronenpaaren als Elektronendonatoren (+M-Effekt) die chromophore Struktur verändern. Dabei wird eine Ausdehnung des  $\pi$ -Elektronensystems bewirkt und damit einhergehend die Absorption in den längerwelligen Bereich verschoben (ebd., S. 127). Antiauxochrome Substituenten (-NO<sub>2</sub>, -SO<sub>3</sub>H, -NH<sub>3</sub><sup>+</sup>) weisen dabei einen elektronenentziehenden Charakter (-M-Effekt) auf.

#### *Bathochromie*

Bei einer Bathochromie wird durch bathochrome Gruppen (-OH, -NH<sub>2</sub>, -NR<sub>2</sub> und -OCH<sub>3</sub>) eine Verschiebung des Absorptionsspektrums in den längerwelligen und somit energieärmeren Bereich des elektromagnetischen Spektrums bewirkt. Jede bathochrome Verschiebung von  $\pi \rightarrow \pi^*$  und  $n \rightarrow \pi^*$  Elektronenübergängen wird bspw. durch die Zufuhr von konjugierten Doppelbindungen erreicht (ebd., S. 128). Diese Erweiterung der Chromophore ist vor allem für eine Absorption im VIS-Bereich relevant (ebd., S. 130).

So lässt sich zusammenfassen, dass insbesondere p-Orbitale für Farbstoffmoleküle wesentlich sind, diese werden auch als  $\pi$ -Bindungen bezeichnet. So führen mehrere  $\pi$ -Bindungen in einem Molekül zu Überlappungen und somit zu einer Delokalisierung der Elektronen über mehrere Elemente. Diese  $\pi$ -Elektronensysteme können durch geringere Energien angeregt werden und erscheinen mit farbgebenden Molekülen, den sogenannten Chromophoren farbig. Durch die Absorption elektromagnetischer Schwingungen im ultravioletten (UV) und dem sichtbaren (VIS) Bereich des Spektrums wird die Farbigkeit eines Moleküls durch eine Elektronenanregung angestoßen. Dabei wird durch den Übergang der Elektronen aus dem Grundzustand in den angeregten Zustand Energie im Bereich des sichtbaren Lichtes absorbiert und die daraus resultierende reflektierte Komplementärfarbe erscheint als Farbe des Moleküls. Dabei wird jede Elektronenanregung von unterschiedlichen Rotations- und Schwingungsübergängen begleitet, woraus längere oder kürzere Absorptionsbanden resultieren (Herbst & Hunger, 1995, S. 12).

## 4.6 Untersuchung sensorischer Farben

Im Folgenden werden die drei eigens entwickelten sensorischen Farben, die als sensorische Tinten Bestandteil des in der vorliegenden Dissertation entwickelten intelligenten Codes sind, näher erörtert. Hierbei werden die chemischen Strukturen, Zusammensetzungen und Funktionalitäten genauer beschrieben.

### 4.6.1 Hydrochrom

Hydrochrome Farbsysteme – Hydro, vom alt. griech. ὕδωρ (*hydōr*) für Wasser (Gottwein, 2020) – verändern oder verlieren ihre Farbigkeit unter dem Einfluss von Wasser ( $H_2O$ ). Da sich Wasser negativ auf unterschiedliche Prozesse und Werkstoffe auswirken kann, werden hydrochrome Farbsysteme eingesetzt, um das Eindringen von Wasser oder Feuchtigkeit bspw. in der Hardware von Smartphones anzuzeigen. Ebenfalls werden diese Farbsysteme für den Nachweis der Unterbrechung einer Kühlkette herangezogen, um den Tau- und erneuten Gefrierprozess von gekühlten Waren zu kennzeichnen. Sie können außerdem als Indikator mit einem Trockenmittel, wie bspw. Kieselgel kombiniert werden, um den Feuchtigkeitsgehalt und somit die Übersättigung des Trockenmittels anzuzeigen (z. B. Superabsorber mit Indikator in Windeln). Natürliche hydrochrome Phänomene finden sich auch in der Umwelt, wie bspw. die Skelettblume (*Diphylleia grayi*). Diese Blume weist weiße Blütenblätter auf, die bei Kontakt mit Wasser durchsichtig werden. Nach einer Trocknungsphase werden die Blütenblätter erneut weiß (Yong et al., 2015). Ebenfalls weisen tropische Käfer, wie bspw. der männliche Käfer *Dynastes Hercules*, hydrochrome Eigenschaften auf. Dieser ist normalerweise grünlich gefärbt, wenn jedoch die relative Luftfeuchtigkeit 80 % übersteigt, verändert sich seine Farbgebung in Schwarz (Rassart et al., 2008). Des Weiteren weisen einige hydrochrome Materialien neben ihrer Reversibilität auch toxische und spezifischer ökotoxische Eigenschaften auf, sodass sie für den Lebensmittel- oder Gesundheitsbereich ungeeignet sind. Ein beispielhaftes toxisches Material mit hydrochromen Eigenschaften ist Kupfer(II)-sulfat (ZVG Nr.:1760), das im dehydrierten Zustand farblos ist und im hydrierten Zustand eine blaue Farbe aufweist. Zusätzlich zu nennen ist Cobalt(II)-chlorid (ZVG Nr.: 2590), das im dehydrierten Zustand eine blaue Farbe und im hydrierten Zustand eine violette Farbe aufweist. Cobalt(II)-chlorid hat zudem kanzerogene Eigenschaften<sup>2</sup>. Dahingehend war es das Ziel, eine irreversible, hydrochrome sensorische Farbe zu entwickeln, die auf umweltgefährliche Stoffe verzichtet und sich als sensorische Tinte eignet. – Diese Vorgabe wurde ebenfalls für alle anderen verwendeten Materialien umgesetzt.

Um eine irreversible hydrochrome sensorische Farbe zu entwickeln, wurden unterschiedliche Materialien gesichtet und auf ihre möglichen Eigenschaften hin untersucht. Als Ergebnis dieses iterativen Prozesses lag der Fokus, ausgehend des dazustellenden konzeptionellen Entwurfs, auf Patentblau V (Abbildung 31, links), einem Triphenylmethanfarbstoff. Ein solcher Farbstoff ist in seiner Triphenylmethan-Teilstruktur farblos (Abb., links, orange hervorgehoben). Die chemische Struktur verändert sich durch die Zuführung von elektronenliefernden Substituenten (*Auxochrome*), sodass mindestens zwei aromatische Ringe in ortho-Stellung an ein zentrales ungesättigtes Kohlenstoffatom gebunden werden. Somit

---

<sup>2</sup> Stoffverbindungen sind unter der ZVG Nummer in der Datenbank des Gefahrstoffinformationssystems (GESTIS) aufrufbar.



ist die chemische Struktur in der Lage, ein Chromophor zu bilden, der die Absorption von nicht sichtbaren in den langwelligen Bereich verschiebt. Durch das Hinzufügen einer zweiten auxochromen  $(\text{CH}_3)_2\text{N}$ -Gruppe (Abb. links, blau) an die Struktur, erhöht sich die konjugative Wechselwirkung. Die Delokalisierung der  $\pi$ -Elektronen in der Struktur führen zum Farbstoff Malachitgrün, der entsprechend eine grüne Farbgebung aufweist. Entsprechend weist die Farbstoffform Kristallviolett (Abb. rechts) ebenfalls eine strukturelle Veränderung, auf die jedoch an allen drei Stellungen Auxochrome  $(\text{CH}_3)_2\text{N}$ -Gruppen aufweist und somit in violetter Farbgebung erscheint, wohingegen Patentblau V statt der dritten eine zusätzliche Sulfonsäuregruppe in ortho-Stellung aufweist. So finden sich Triphenylmethanfarbstoffe wie bspw. Patentblau V in Lebensmittelfarbstoffen (wie E 131), bei der Herstellung von Druckfarben und Indikatoren, in intelligenten Beschichtungen, in Durchschlagpapieren, beim Einfärben von Textilien und in anderen Bereichen (Klöckl, 2015, S. 312). Hinzuzufügen ist, dass Patentblau V nach dem Globally Harmonised System (GHS) als ungefährlicher Stoff einzustufen ist (ZVG Nr.: 115098).

Nach der Verordnung EU Nr. 231/2012 der Europäische Kommission, 2012 besteht Patentblau V in erster Linie aus Calcium-, Natrium- und auch Kaliumverbindungen, die sich aus „[4-( $\alpha$ -(4-Diethylaminophenyl)-5-hydroxy-2,4-disulphophenyl-methyliden)-2,5-cyclohexadien-1-yliden] Diethylammoniumhydroxid“ in Form eines inneren Salzes zusammensetzt (ebd.). Um eine Verbindung in Leukoform (reduzierte Form; farbloser Farbstoff, wird oxidiert; erscheint farbig) herzustellen, wird Patentblau V durch „Kondensation und Sulfonierung von N,N-Diethylanilin und 3-Hydroxy-benzaldehyd unter sauren Bedingungen (*Schwefelsäure*) in einer einstufigen Reaktion“ hergestellt (ebd.). Die Leukoverbindung wird darauffolgend „in Ammoniak aufgelöst und mit Mangandioxid in Phosphorsäure oxidiert und mit Natriumhydroxid neutralisiert“ (ebd.). Darauffolgend wird entweder (I) „Salzsäure und Calciumchlorid“ oder (II) „Salzsäure und Natriumchlorid“ ausgefällt, der Niederschlag gepresst, getrocknet und in einem anschließenden Prozess zur jeweiligen Ausgangskorngröße vermahlen (ebd.). In Abbildung 31 ist ersichtlich, dass sich die Patentblau-Variante VF durch die fehlende Hydroxygruppe an fünfter Stelle des Benzolrings auszeichnet, die bei Patentblau mit der Variante V durch Sulfonierung mit 3-Hydroxybenzaldehyd hinzugefügt wurde. So sind diese Unterschiede ebenfalls im Vergleich zwischen den systemischen Namen (II) und (III) zu erkennen. Patentblau V wird aufgrund seiner Eignung als Lebensmittelfarbstoff (E 131) und seiner Umweltfreundlichkeit für die nachfolgenden Untersuchungen bevorzugt.

- I. Calciumsalz (CAS-Nr.: 3536-49-0,  $\text{C}_{27}\text{H}_{31}\text{N}_2\text{O}_7\text{S}_2 \cdot 0,5\text{Ca}$ )  
Calciumbis(4-[[4-(diethylamino)phenyl][4-(diethyliminio)-2,5-cyclohexadien-1-yliden]methyl]-6-hydroxy-1,3-benzoldisulfonat)
- II. Natriumsalz (CAS-Nr.: 20262-76-4,  $\text{C}_{27}\text{H}_{31}\text{N}_2\text{NaO}_7\text{S}_2$ )  
Natrium-4-[[4-(diethylamino)phenyl][4-(diethyliminio)-2,5-cyclohexadien-1-yliden]methyl]-6-hydroxy-1,3-benzoldisulfonat
- III. Natriumsalz (CAS-Nr.: 3536-49-0,  $\text{C}_{27}\text{H}_{31}\text{N}_2\text{NaO}_6\text{S}_2$ )  
Natrium-4-[[4-(diethylamino)phenyl][4-(diethyliminio)-2,5-cyclohexadien-1-yliden]methyl]-1,3-benzoldisulfonat

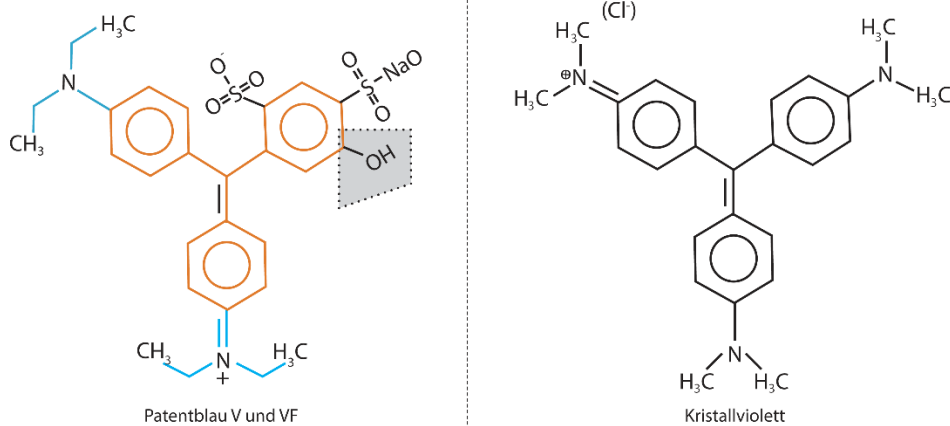


Abbildung 31 Strukturformel Patentblau V und Patentblau VF, ohne Hydroxygruppe (-OH, grau)

#### 4.6.1.1 Ideenansatz

Auf Grundlage des chemischen Prinzips eines Tintenlöschers wurde eine irreversible hydrochrome sensorische Farbe entwickelt. Der chemische Mechanismus des Tintenlöschers (Abbildung 32) war bereits in den 1930er Jahren von der Firma Pelikan unter der Bezeichnung „Radierwasser“ auf den Markt gebracht worden (Pelikan, 2020). Der vollständige Tintenlöschstift besteht aus einem farblosen, mit einem Reduktionsmittel durchsetzten Faservlies. Dieses dient zum einen dem Löschen der Tinte und zum anderen zum Schreiben auf der korrigierten Stelle mit einer löschresistenten Tinte. Die chemischen Bestandteile eines Tintenlöschers zur Entfernung der Tinte bestehen aus gereinigtem Wasser, Reduktionsmitteln und weiteren sekundären Zusätzen. Entsprechend weist die Schreibtinte die dazu passenden chemischen Bestandteile auf, die auf einem Triphenylmethanfarbstoff, Wasser, Konservier- sowie Feuchthaltemittel und weiteren Additiven basiert (ebd.).

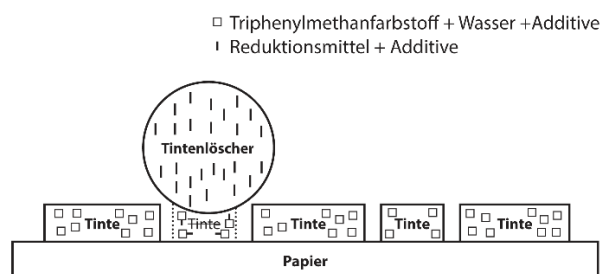


Abbildung 32 Konzept eines Tintenlöschers

#### 4.6.1.2 Konzeptioneller Ansatz

Das Prinzip der folgenden irreversiblen hydrochromen sensorischen Farbe (Abbildung 33) basiert auf der Zusammensetzung von Patentblau V, Lösungsmittel auf Alkoholbasis, Reduktionsmittel und weiteren Additiven. Als Matrix für die gelöste und eingeschlossene Stoffzusammensetzung wurden in einem nachfolgenden Untersuchungsprozess ein geeignetes Lösungsmittel und ein geeignetes Reduktionsmittel identifiziert. Die Matrix auf Alkoholbasis war notwendig, um die Wechselwirkung und somit die Reaktivität für den Nachweis von Wasser zu gewährleisten. Bei Kontakt mit Wasser wird die Reaktion mit dem Reduktionsmittel angeregt, dadurch gelangt der Triphenylmethanfarbstoff in eine Wechselwirkung mit dem Reduktionsmittel. Dabei wird die Farbigkeit aufgrund der damit veränderten Geometrie, der chemischen Struktur in den nicht sichtbaren Wellenlängenbereich verschoben und ist somit für das menschliche Auge unsichtbar (Bilgin & Backhaus, 2020c). Der Farbighkeits- und Entfärbungsprozess von Patentblau V wird im Folgenden beschrieben.

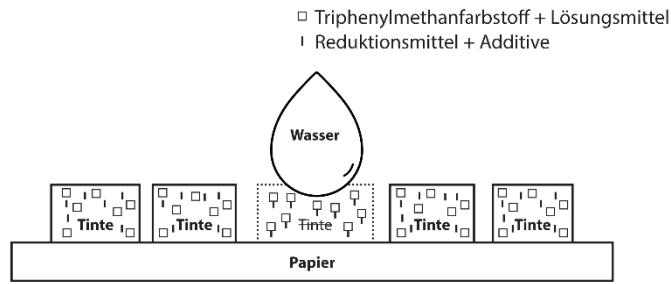


Abbildung 33 Konzept einer hydrochromen sensorischen Farbe

#### 4.6.1.3 Farbigkeit von Patentblau V

Aufgrund der  $sp^2$ -Hybridisierung des zentralen ungesättigten Kohlenstoffatoms an den drei aromatischen Verbindungen ist die Farbigkeit der Triphenylmethanfarbstoffe auf die völlige planare Anordnung der Moleküle zurückzuführen. In diesem Fall sind die  $\pi$ -Elektronen der Doppelbindungen und die nichtbindenden Elektronen der Substituenten über das gesamte Molekül delokalisiert. Dadurch können die Elektronen leichter angeregt und niederenergetische Strahlung effizienter aus dem sichtbaren Spektrum absorbiert werden. Nicht absorbierte, aber reflektierte Spektralanteile führen zum wahrnehmbaren komplementären Farbeindruck (Blume, 2003).

#### 4.6.1.4 Entfärbungsprozess

Reduktionsmittel wie das negativ mit Wasser geladene Anion Hydroxid-Ion ( $OH^-$ ) oder das Hydrogensulfit-Ion ( $HSO_3^-$ ) greifen das zentrale Kohlenstoffatom des Triphenylmethan-Farbstoffmoleküls an und werden miteinander addiert. Hierbei wird die Geometrie des Farbstoffmoleküls Patentblau V verändert (tetraedisch ausgerichtet) und das zentrale Kohlenstoffatom wird  $sp^3$ -hybridisiert und ist somit gesättigt. Folglich wird durch diese Reduktion die Delokalisierung (Resonanz) der Elektronen unterbrochen, wodurch sichtbares Licht nicht mehr mit den Elektronen innerhalb der Struktur wechselwirken kann (ebd.). Das Patentblau V absorbiert nun nur noch UV-Strahlung anstelle von sichtbarem Licht, liegt nicht mehr im sichtbaren Wellenlängenbereich und erscheint folglich farblos (Abbildung 34) (Bilgin & Backhaus, 2020a, 2020b).



Abbildung 34 Farbveränderung von blau zu transparent

#### 4.6.1.5 Interpretation

Die Interpretation der hydrochromen sensorischen Farbe kann im  $L^*a^*b^*$  Farbraum erfolgen. Hierzu kann die  $L^*$ -Achse, die für Helligkeit steht, für die Messbarmachung des Entfärbungsprozesses herangezogen werden. Mit der folgenden Farbabstandsformel  $\Delta L^*_{76}$  kann hierzu die Helligkeitsdifferenz berechnet werden (20):

$$\Delta L^*_{76} = L^*_{M2} - L^*_{M1} \quad (20)$$

Eine weitere Möglichkeit der Interpretation besteht in der Betrachtung der  $b^*$ -Achse, die den blauen bis gelben Farbbereich darstellt und mit der folgenden Farbabstandsformel  $\Delta b^*_{76}$  der Gelb-Blau-Buntheit berechnet und interpretiert werden kann (21):

$$\Delta b^*_{76} = b^*_{M2} - b^*_{M1} \quad (21)$$

So kann zudem mittels beider Farbabstandsformeln  $\Delta E_{76}$  und  $\Delta E_{00}$  (Kapitel 4.1) der Abstand der gesamten Farbveränderung des hydrochromen Sensors ebenfalls interpretiert werden.

#### 4.6.1.6 Lösungsmittel Identifizierung

Um den Anforderungen entsprechend eine hydrochrome sensorische Farbe zu entwickeln, war die Identifizierung eines geeigneten Lösungsmittels von Relevanz (Tabelle 17). Der Farbstoff Patentblau V und das Lösungsmittel sollten eine homogene Phase in einer homogenen Lösung bilden. Die Agglomeration und auch Sedimentation von Partikeln in der Lösungsmittelmatrix sollte vermieden werden, da diese zu einer Verstopfung der Tintenstrahldüse und der Leitungen führen und den Druckprozess erheblich beeinträchtigen oder unterbrechen können (Magdassi, 2010, S. 34). Nach dem Skalenniveau der Ordinalskala wurden ausgewählte Lösungsmittel qualitativ nach ihren Ausprägungen in den Kategorien Löslichkeit, Homogenität und Sedimentation bewertet (Tabelle 16).

Skala	Löslichkeit im Lösungsmittel	Farbauftrag Homogenität	Sedimentation
-- schwach	Viele Feststoffe festzustellen	Pastos unterbrochener Farbauftrag	Kaum bis kein Niederschlag
- mittel	Vereinzelt Feststoffe festzustellen	Phasenweise Halbdeckend	Leichter Niederschlag
+ stark	Kaum Feststoffe festzustellen	Wolkiger oder fleckiger Farbauftrag	Starker Niederschlag
++ sehr stark	(Nahezu) Keine Feststoffe	Gleichmäßiger Farbauftrag	Fast vollständig abgelagert

Tabelle 16 Qualitative Bewertung der Ausprägungen

Lösungsmittel auf Wasserbasis, wie das demineralisierte Wasser oder E24, eine speziell für den Epson-Drucker entwickelte wasserbasierte Basistinte, konnten nicht zusammen mit der irreversiblen hydrochromen sensorischen Farbe verwendet werden, da diese ansonsten sofort reagieren. Des Weiteren fanden sich Literaturhinweise zur Löslichkeit von Patentblau V, wobei es in Wasser am besten, in Ethanol mäßig löslich und in Öl unlöslich ist (RÖMPP, 2020). Entsprechend wurden kombinierte Untersuchungen mit den drei Lösungsmitteln Isopropanol, Methanol, Ethanol und dem Farbstoff Patentblau V bei einer konstanten Raumtemperatur (Kapitel 3.6) durchgeführt. Bei der Untersuchung wurde jeweils 0,10 g Patentblau V in 10 ml der zuvor aufgeführten Lösungsmittel gelöst. Hinzuzufügen ist, dass Patentblau V eine Löslichkeit in Wasser bei 20 °C von 20 g/L und in Ethanol bei 90 °C von 100 g/L aufweist (Otterstätter, 1999, S. 51). Bei Farbstoffgemischen, wie Patentblau V, das aus Komponenten unterschiedlicher Löslichkeit aufgebaut ist, ist auf eine vollständige Lösung des Farbstoffpulvers zu achten (Otterstätter, 2007, S. 70). So weist Otterstätter (ebd.) darauf hin, in einem empfohlenen Konzentrationsbereich der Löslichkeit zu arbeiten, „in denen nicht die Gefahr des Auskristallisierens der Farbstoffe besteht“. Dahingehend erwies sich das Lösungsmittel Isopropanol als wenig geeignet, da der Farbstoff nicht vollständig gelöst werden konnte, sondern auf dem Substrat Kristallisationserscheinungen zeigte und sich dadurch nicht homogen aufgetragen ließ. Die Kristallisation ist auf die Übersättigung der unterschiedlichen Konzentrationen im Zusammenhang mit der gegebenen Temperatur zurückzuführen. Die restlichen Kristalle wurden im späteren Verlauf ausgefiltert (Kapitel 3.2.4). Ein weiteres Ausschlusskriterium war die Untersuchung der Sedimentation des Farbstoff-Lösungsmittel-Komplexes. Nach einer Aufbewahrungszeit von 24 Stunden waren die Farbstoffpartikel im Lösungsmittel Isopropanol sedimentiert, was zu einem Ausschlusskriterium von Isopropanol führte: Diese lagern sich ab und die Zugänge zu den Düsen können verstopfen. Ethanol und Methanol erwiesen sich als geeignete Lösungsmittel (*grau*) für Patentblau V, da kaum eine Ablagerung bzw. eine schwache Ablagerung stattfindet.

Lösungsmittel	Löslichkeit	Homogenität	Sedimentation
Demineralisiertes Wasser	++	++	-
E24	+	-	+
Isopropanol	+	--	++
Methanol	+	+	-
Ethanol	+	+	-

Tabelle 17 Identifizierung eines geeigneten Lösungsmittels für Patentblau V

#### 4.6.1.7 Identifizierung eines geeigneten Reduktionsmittels

Die Lösungsmittel Methanol und Ethanol werden als eine Lösungsmittelmatrix für das Einbetten eines Reduktionsmittels herangezogen (Tabelle 18). Hierbei wird die Löslichkeit anhand der folgenden Zusammensetzung von Patentblau V und 0,10 g des jeweiligen Reduktionsmittels in jeweils 10 ml Methanol oder Ethanol gelöst und die Entfärbung der Stoffzusammensetzung qualitativ unter der Bewertung der beobachteten Ausprägungen untersucht. So werden Natriumcarbonat, Natriumsulfit, Natriumthiosulfat und Natriumbicarbonat als mögliches Reduktionsmittel herangezogen. Die aussagekräftigsten Ergebnisse einer vollständigen Entfärbung (++) wurden mit Natriumcarbonat und Natriumbicarbonat in Ethanol beobachtet. In Ethanol zeigten Natriumthiosulfat eine mäßige (-) und Natriumsulfit eine sehr schlechte, bis kaum sichtbare (--) Entfärbung. In Methanol waren Natriumsulfit und Natriumbicarbonat gut (+) löslich, während Natriumcarbonat eine mäßige (-) und Natriumthiosulfat eine schlechte (--) Löslichkeit aufwies. Für die nachfolgenden Untersuchungen wurde die Stoffkombination Patentblau V mit Natriumbicarbonat und Ethanol herangezogen. So zeigt Natriumbicarbonat eine mäßige Löslichkeit in Wasser (93,4 g/L, 20 °C) und ist nahezu unlöslich in Ethanol (Carl Roth, 2016; Otterstätter, 1999, S. 5). Aus diesem Grund eignet sich Natriumbicarbonat gut (+) als Reduktionsmittel, da es erst mit Kontakt zu Wasser reagiert und erst dann das Patentblau V entfärbt.

Lösungsmittel	Natriumcarbonat	Natriumsulfit	Natriumthiosulfat	Natriumbicarbonat
Methanol	-	+	--	+
Ethanol	++	--	+	++

Tabelle 18 Reduktionsmittel für die Entfärbung von Patentblau V

#### 4.6.1.8 Stoffkonzentrationen

Bei Voruntersuchungen für die Entwicklung sensorischer Farben wurden Kombinationen der Mengenteile in der Lösung aus Farbstoff, Reduktions- und Lösungsmittel ermittelt, welche sich durch hohe Farbgebung, Kontrastumfang und geringeres Kristallisationspotential auszeichnen (Tabelle 19).

Nr.	Farbstoffgehalt	Reduktionsmittelgehalt	Lösemittelgehalt
1.1	0,10 g Patentblau V	0,10 g Natriumbicarbonat	10 ml Ethanol
1.2	0,08 g Patentblau V	0,08 g Natriumbicarbonat	10 ml Ethanol
1.3	0,06 g Patentblau V	0,06 g Natriumbicarbonat	10 ml Ethanol
1.4	0,04 g Patentblau V	0,04 g Natriumbicarbonat	10 ml Ethanol
1.5	0,02 g Patentblau V	0,02 g Natriumbicarbonat	10 ml Ethanol

Tabelle 19 Hydrochrome Stoffkonzentrationen

#### 4.6.1.9 Lichtechtheit

Lichtechtheit „löst an den Farbmolekülen photochemische Prozesse aus, die zu Farbveränderungen – dem sogenannten »Ausbleichen« – führen“ (Fogra, 2001). Entsprechend wurde der Farbstoff Patentblau V und die hydrochrome sensorische Farbe hinsichtlich ihrer Lichtechtheit untersucht. Darüber hinaus gibt es weitere Hinweise, dass Patentblau V mäßig lichtecht ist und unter dem Einfluss von Licht schnell

verblasst. Unter anderem beschreibt Scotter (2015, S. 220) Patentblau V ebenfalls als mäßig lichtstabil und auch als verblasend in Verbindung mit Schwefeldioxid, Ascorbinsäure, Fruchtsäuren und alkalischen Medien.

Im Folgenden werden zwei verschiedene Zusammensetzungen von Patentblau V hinsichtlich ihrer Lichtechtheit untersucht:  $V_a$  (Patentblau V + Ethanol) besteht aus dem Farbstoff und dem Lösungsmittel.  $V_b$  (Patentblau V + Ethanol + Natriumbicarbonat) repräsentiert die Grundzusammensetzung der hydrochromen sensorischen Farbe. Das methodische Vorgehen zum experimentellen Setting kann in *Kapitel 3.5.1* nachgelesen werden.

Die Farbveränderungen wurden im  $L^*a^*b^*$  Farbraum bewertet. Das Kriterium ist dabei der  $L^*$ -Wert (Helligkeit), welcher sich für die Darstellung einer Helligkeitszu- oder -abnahme eignet. Die Abbildung 35 zeigt den zeitlichen Verlauf von  $L^*$  bei einer UVA Lichtexposition mit einer Leistung von  $0.3 \text{ mW/cm}^2$  (Abbildung 16). Beide Proben,  $V_a$  und  $V_b$  verdunkeln dabei über den zeitlichen Verlauf kontinuierlich. Ebenfalls zu erkennen ist, dass die Formulierung  $V_a$  ohne Reduktionsmittel vor der Exposition im  $L^*$ -Wert geringfügig heller ( $L^*$ : 54,53) war, als  $V_b$  mit Reduktionsmittel ( $L^*$ : 51,09). Prägnant ist der bei steigender Expositionsdauer abnehmende  $L^*$ -Wert und somit der messbare Verdunklungsprozess beider Zusammensetzungen. Ebenfalls zu erkennen ist, dass die Formulierung  $V_a$  ohne Reduktionsmittel bereits zu Beginn geringfügig heller ( $L^*$ : 54,53) war, als  $V_b$  mit Reduktionsmittel ( $L^*$ : 51,09). Nach 20 Stunden und längerer Expositionsdauer waren zwischen beiden Proben keine signifikanten Unterschiede in der Helligkeit mehr nachzuweisen.

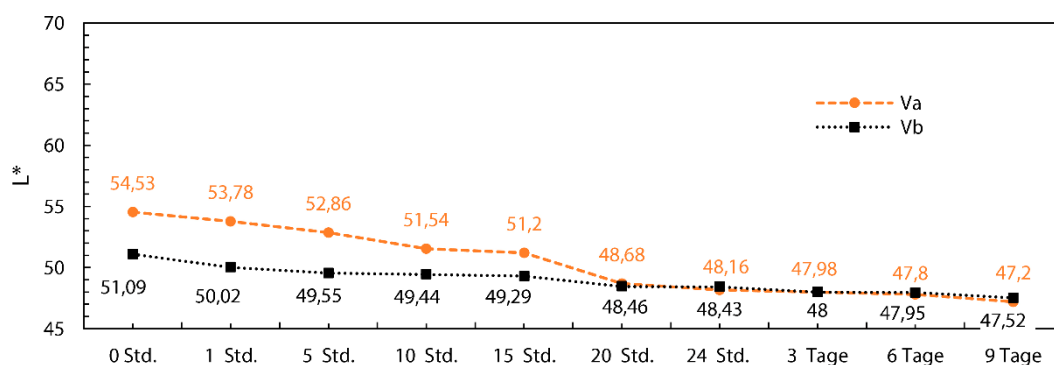


Abbildung 35  $L^*$ -Wert (Patentblau V, hydrochromer Sensor) bei andauernder UV-Licht-Exposition

Der zu Beginn gemessene Unterschied beider Zusammensetzungen wurde somit im zeitlichen Verlauf kompensiert. Im Folgenden lässt sich zusammenfassen, dass  $V_a$  im unbelichteten Zustand ein  $L^*$ -Wert von 54,53 aufzuweisen ist, der sich jedoch nach 24 Stunden auf einen Wert von 48,16 verringerte. Derselbe Effekt ist ebenfalls bei  $V_b$  messbar. Zu Beginn wurde ein  $L^*$ -Wert von 51,09 gemessen und erreichte nach 24 Stunden einen  $L^*$ -Wert von 48,43. Nach mehr als einer Woche (9 Tage) kontinuierlicher Exposition erreichte  $V_a$  einen  $L^*$ -Wert von 47,2 und  $V_b$  einen  $L^*$ -Wert von 47,52. Infolgedessen dunkelten beide Zusammensetzungen (Abbildung 36) nach einer kontinuierlichen UVA-Expositionsdauer signifikant um eine Gesamtfarbabweichung von  $V_a$ ,  $\Delta E_{00}$  8,53 und  $V_b$ ,  $\Delta E_{00}$  13,32. Patentblau V ist als Material für einen Sensor einsetzbar, da sich der Kontrast, welcher sich aus der zeitlich intensivierte Verdunklung durch eine UVA Exposition bei einer Entfärbung in Kontakt mit Wasser deutlich zur vorherigen Farbigkeit abhebt.

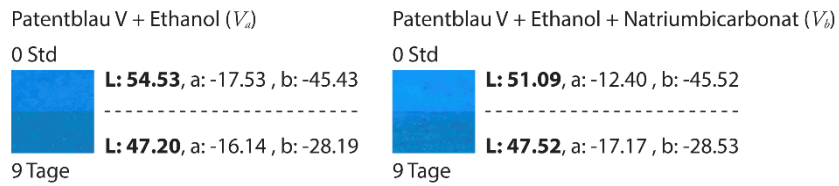


Abbildung 36 Lichtechtheit vor und nach der Exposition mit UV-Licht

Anhand der in der Abbildung 37 dargestellten Remissionskurve ist keine, bis kaum eine geringfügige Abweichung der beiden Zusammensetzungen  $V_a$  und  $V_b$  im unbelichteten Zustand zu erkennen. Diese sind als nahezu überlappte Kurven dargestellt. Die charakteristische maximale Intensität von Patentblau V ist im Bereich der Wellenlängen von 425 nm bis ca. 560 nm zu erkennen. Der Mittelpunkt der maximalen Intensität verortet sich bei 465 nm.

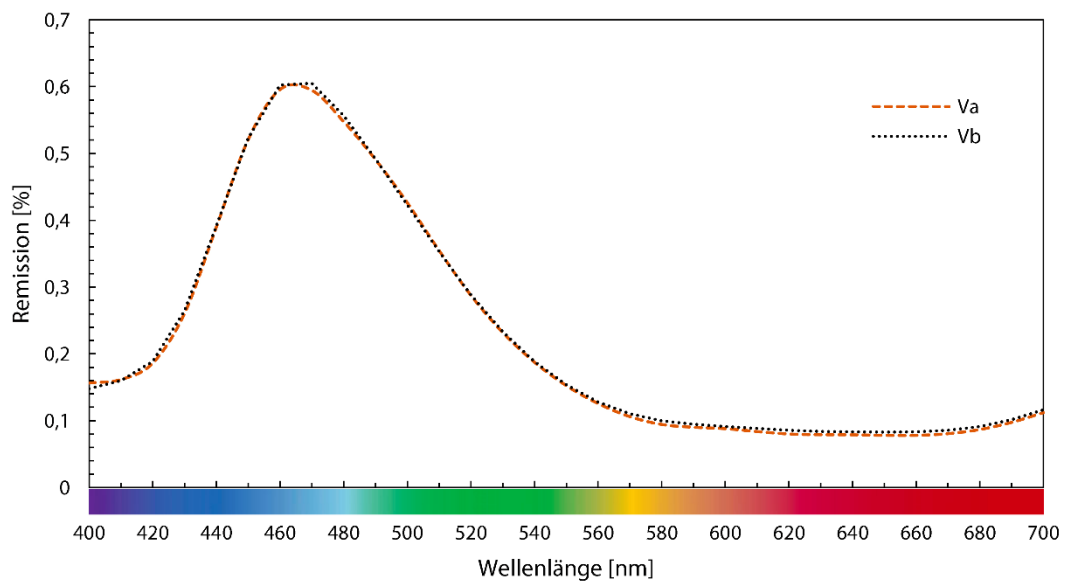


Abbildung 37 Remissionskurven Patentblau

## 4.6.2 Photochrom

Photochrome Farbsysteme – Photo, vom alt. griech.  $\varphi\acute{\omega}\varsigma$  (*phōs*) für Licht (Gottwein, 2020) – verändern ihre Farbigkeit unter dem Einfluss von Licht einer spezifischen Wellenlänge. Hierbei wird durch den Stimulus elektromagnetische Strahlung, spezifischer durch energetisch geladene Lichtquanten, sogenannte Photonen ( $\gamma$ ), die chemische Struktur der bestrahlten Materialien verändert. Daraus resultierend wird ebenfalls eine Änderung des Absorptionsspektrums angestoßen, wobei der dabei reflektierte spektrale Anteil die wahrnehmbare Farbveränderung bewirkt. Abhängig vom Stoff bzw. seiner chemischen Struktur sind die lichtinduzierten Strukturveränderungen reversibel oder irreversibel. Ebenso kann die Farbveränderung eine so bezeichnete negative (*Entfärbung*) oder positive (*Färbung*) Photochromie bewirken. Dabei kann bei den reversiblen Verbindungen die Strukturveränderung thermisch oder photochemisch zurückgeschaltet werden. Die Funktionalität der photochromen Verbindungen basieren bspw. auf der Ringöffnungsreaktion von Spiropyran oder Spirooxazine. Weitere photochrome Verbindungen basieren unter anderem auf der photoinduzierten Ringschlussreaktion – bedingt durch eine cis-trans-Isomerisierung – des Diarylethene, eines Stilbenderivats (Irie et al., 2014, S. 2). So kann eine photochrome Verbindung auch durch eine Heterolyse, bei der das elektronegativere Element die negative Ladung aufnimmt, von Triphenylmethanfarbstoffen, die eine reversible und negative Photochromie aufweisen können, erfolgen (MacNair, 1967, S. 780). Historisch ist der Terminus Photochromie auf (Hirshberg, 1950) zurückzuführen. Schriftlich beschrieben wurde der photochrome Effekt allerdings bereits von Fritzsche (1867, S. 336). Dieser „setzt die heisse Mischung dem directen Sonnenlichte aus [sic!]“ (ebd.) und beobachtete eben dieses Phänomen. Hirshberg und Fischer (1954) haben gemeinsam Spiropyran wissenschaftlich untersucht, wobei auch Hirshberg (1956) die Idee der Verwendbarkeit, als photochemischer Speicher formuliert hatte.

Heute finden sich einige Forschungen zu molekularen Schaltern und optischen 3-D-Speichern, die auf photochromen Verbindungen basieren und mittels eines Lasers Daten dreidimensional beschreib-, lös- und wiederbeschreibbar verarbeiten lassen (Patel et al., 2017). Um den definierten Zielerfordernissen entsprechend eine irreversible photochrome sensorische Farbe zu entwickeln, wurden auch hier zuvor verschiedene Materialien gesichtet und auf ihre möglichen Eigenschaften untersucht. Als Ergebnis dieses iterativen Prozesses lag der Fokus auf dem von Diesbach um ca. 1706<sup>3</sup> entwickelten synthetisch anorganischen Pigment Preußisch Blau ( $\text{Fe}_4^{\text{III}}[\text{Fe}^{\text{II}}(\text{CN})_6]_3$ ), das auch unter der Bezeichnung Berliner Blau bekannt ist (ZVG Nr.: 5440) (Kraft, 2012, S. 6). Diesbachs Entdeckung war ein Zufallsereignis, da er versucht hatte ein rotes Pigment auf der Basis von Cochenillerot zu gewinnen. Versehentlich hatte er dabei ein Additiv mit Eisenhexacyanoferrat verunreinigt, sodass daraus eine lichtechte blaue Farbe hervorging (Kraft, 2008, S. 61). Der Einsatzbereich erstreckt sich vom Färbemittel der Uniformen der königlich preußischen Armee um 1700, bis in den heutigen Anwendungsbereichen der Elektrochemie (Wagner & Oetken, 2016, S. 1), der Sensortechnik (Vámos & Gábor, 1973) und in der Medizin (Ruprecht, 2001).

---

<sup>3</sup> „In [...] der Berliner Staatsbibliothek findet man [...] das Jahr 1706 als das Jahr der Erfindung des Berliner Blau angegeben.“ (Kraft, 2008, S. 61)



#### 4.6.2.1 Herstellung von Preußisch Blau

Preußisch Blau (Abbildung 38) wird hergestellt, indem Eisen(III)-chlorid, ( $\text{FeCl}_3$ ) oder Eisen(III)-nitrat, ( $\text{Fe}(\text{NO}_3)_3$ ) mit Kaliumhexacyanoferrat(II), ( $\text{K}_4[\text{Fe}^{\text{III}}(\text{CN})_6]$ ), auch gelbes Blutlaugensalz genannt, in einer wässrigen Lösung vermischt werden, wobei hier Preußisch Blau schließlich ausfällt. Diese Cyano-Verbindung enthält sowohl  $\text{Fe}^{\text{III}}$  als auch  $\text{Fe}^{\text{II}}$  im selben Molekül (Klöckl, 2015, S. 233 f.).

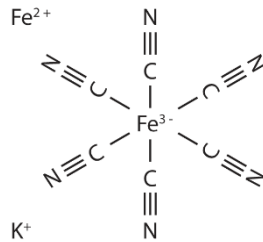
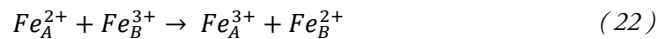


Abbildung 38 Strukturformel Preußisch Blau

#### 4.6.2.2 Farbigekeit von Preußisch Blau

Die preußisch blaue Farbe wird infolge eines Prozesses gebildet, der Intervalenz Charge Transfer (*IVCT*) genannt wird. Hierbei werden durch einen photoinduzierten Elektronentransfer Kationen zwischen  $\text{Fe}^{\text{II}}$ , einer niedrigeren Oxidationsstufe (*Donator*), jedoch in eine höhere Oxidationsstufe (*Akzeptor*) desselben Elements ( $\text{Fe}^{\text{III}}$ ), übertragen. Hierzu müssen sich die Kationen in unterschiedlichen chemischen Umgebungen (*A und B*) befinden und räumlich in oktaedrischer Koordination dicht beieinander sein, so dass sie sich gegenseitig wechselwirken können (Klöckl, 2015, S. 119 ff.) (22):



Dementsprechend ist das  $\text{Fe}^{\text{II}}$ -Kation von einer sechsfachen Kohlenstoffumgebung (*Cyanid*) umgeben und bildet einen Low-Spin-Komplex, während das  $\text{Fe}^{\text{III}}$ -Kation von Stickstoff (*Cyanid*) und Sauerstoff (*Kristallwasser*) umgeben ist. Diese Umgebung „führt zu einem schwächeren Ligandenfeld, einer geringeren Aufspaltung und zu einem High-Spin-Komplex“ und somit zur sichtbaren blauen Farbe (Klöckl, 2015, S. 123).

#### 4.6.2.3 Photochrome Zusammensetzung

Die folgende irreversible photochrome sensorische Farbe (Abbildung 39) bildet als Endprodukt Preußisch Blau. Ihre Funktionalität war bereits um 1842 bekannt (Herschel, 1842) und wurde in der Reproduktionstechnik und den Widerstandsaktionen von Anna Beyer 1933 verwendet (Bromberger & Mausbach, 1987, S. 90). Damit wurde die hier eingesetzte photochrome Funktionalität von Preußisch Blau bereits zu frühen Zeiten eingesetzt.

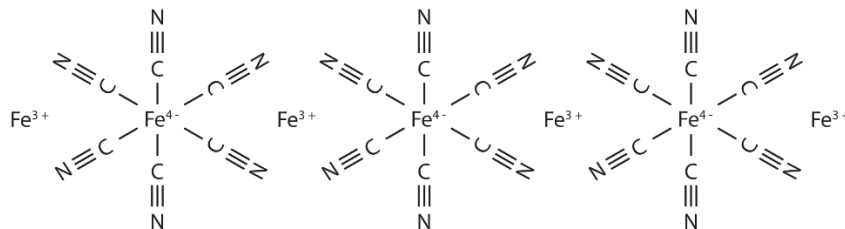
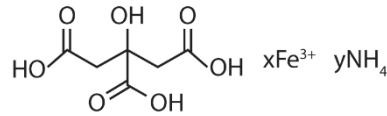


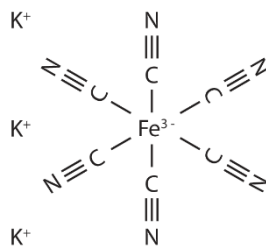
Abbildung 39 Strukturformel Preußisch Blau

Um eine irreversible photochrome sensorische Farbe zu entwickeln, kann unter anderem Ammoniumtrioxalatoferrat(III)  $((\text{NH}_4)_3[\text{Fe}(\text{C}_2\text{O}_4)_3] \cdot 3\text{H}_2\text{O})$  oder Ammoniumeisen(III)-citrat  $(\text{C}_6\text{H}_8\text{O}_7 \cdot n\text{Fe} \cdot n\text{H}_3\text{N})$  als Reduktionsmittel mit einer Löslichkeit von 1200 g/l bei 20 °C (ZVG Nr.: 108991) verwendet werden. Aufgrund der bewährten Eigenschaft im Einsatzbereich der Fotosensibilisierung wird Ammoniumeisen(III)-citrat *Abbildung 40* gewählt.



*Abbildung 40 Strukturformel Ammoniumeisen(III)-citrat*

Als Oxidationsmittel wird Kaliumhexacyanidoferrat(III) (*Abbildung 41*)  $(\text{K}_3[\text{Fe}(\text{CN})_6])$  nach der GESTIS Stoffdatenbank (ZVG Nr.: 4120) mit einer Löslichkeit von 464 g/l bei 20 °C angegeben.



*Abbildung 41 Strukturformel Kaliumhexacyanidoferrat(III)*

Die erwähnten Reduktions- und Oxidationsmittel werden gemeinsam in einem Lösungsmittel gelöst, wobei daraus resultierend eine gelblichgrüne photochrome Lösung entsteht.

#### 4.6.2.4 Belichtungsprozess

Die entwickelte photochrome sensorische Farbe ist irreversibel, unidirektional schaltbar und greift die Funktionalität des lichtempfindlichen Preußisch Blau auf. Der photochrome Effekt beruht auf einer photoinduzierten Redoxreaktion. In Folge einer Belichtung mit UVA-Licht wird das Citrat-Anion oxidiert, wobei das dreiwertige Kaliumhexacyanidoferrat (III) in das zweiwertige Kaliumhexacyanidoferrat (II) reduziert und reagiert mit dem restlich gelösten dreiwertigen Fe (III) zu Preußisch Blau:  $\text{Fe}^{3+} + e^- \rightarrow \text{Fe}^{2+}$ . Infolgedessen entsteht, basierend auf den verschiedenen Oxidationsstufen von Eisen, schrittweise eine dunkelblaue Färbung, das Preußisch Blau (Roesky et al., 1996, S. 57).

#### 4.6.2.5 Interpretation

Die Interpretation der photochromen sensorischen Farbe kann wie in *Kapitel 4.7.2* beschrieben im  $L^*a^*b^*$  Farbraum, unter der Berechnung der Farbdifferenz  $\Delta E_{76}$  oder  $\Delta E_{00}$  erfolgen. Hierbei kann ebenfalls die  $L^*$ -Achse für die Quantifizierung des Färbungsprozesses von Gelblichgrün nach Dunkelblau herangezogen und mit der Farbabstandsformel  $\Delta L^*_{76}$  hierzu die Helligkeitsdifferenz berechnet werden.

#### 4.6.2.6 Lösungsmittel Identifizierung

Die chemische Zusammensetzung der photochromen sensorischen Farbe bedingt, dass demineralisiertes Wasser als ein integraler Bestandteil zu berücksichtigen ist. Das zu verwendende Wasser, in Form eines Kristallwassers (Hydrat), ist wesentlich für die Bildung einer Kristallgitterstruktur. Spezifischer ausgedrückt wird das Kristallwasser benötigt, um für das  $\text{Fe}^{\text{III}}$ -Kation für einen Low-Spin-Komplex die notwendige Umgebung zu bilden.

Für die notwendige Umgebung mit Stickstoff und Sauerstoff liefert das Kristallwasser den Sauerstoff, wobei das Ammoniumion das notwendige Gegenion bildet, das für die Farbtonung notwendig ist (ebd., S.234). Somit kann als Lösungsmittel demineralisiertes Wasser eingesetzt werden.

#### 4.6.2.7 Stoffkonzentrationen

Im Folgenden werden verschiedene Konzentrationen der photochromen sensorischen Farbe aufgelistet (Tabelle 20). Die Zusammensetzungen wurden in Voruntersuchungen identifiziert, welche die beste Funktionalität des photochromen Stoffes fokussierten.

Nr.	Oxidationsmittelgehalt	Reduktionsmittelgehalt	Lösemittelgehalt
1.1	0,20 g Kaliumhexacyanidoferrat(III)	0,25 g Ammoniumeisen(III)-citrat	5 ml dem. Wasser
1.2	0,40 g Kaliumhexacyanidoferrat(III)	0,50 g Ammoniumeisen(III)-citrat	5 ml dem. Wasser
1.3	0,60 g Kaliumhexacyanidoferrat(III)	0,75 g Ammoniumeisen(III)-citrat	5 ml dem. Wasser

Tabelle 20 Photochrome Stoffkonzentrationen

#### 4.6.2.8 Lichtechtheit

Die Lichtechtheit wird beim reinen Pigment Preußisch Blau als ausgezeichnet bewertet (Kirby, 1993, S. 63). Bei der photochromen sensorischen Farbe bilden die einzelnen Oxidationsstufen des Eisens eine unidirektional tiefer werdende farbliche Abstufung. Dementsprechend ist die maximal belichtete Oxidationsstufe das reine Preußisch Blau, welches lichtecht und somit in seiner Erscheinung konstant ist.

### 4.6.3 Halochrom

Halochrome Farbsysteme – Halo, vom alt. griech. ἅλς (*halo*) für Salz (Gottwein, 2020) – verändern ihre Farbigkeit in Abhängigkeit vom Ladungszustand ihrer Molekularstruktur. Halochrome Verbindungen besitzen die Funktionalität von pH-Indikatoren (*Säure-Base*) zur Bestimmung des pH-Werts in Kombination mit einer Farbverschiebung (Freeman & Peters, 2000, S. 359). Die Funktionalität kann dazu verwendet werden, um den Verlauf einer chemischen Reaktion zu verfolgen oder den Zustand eines chemischen Systems zu charakterisieren. Mit halochromen Verbindungen lässt sich der Verlauf einer chemischen Reaktion verfolgen und der Zustand eines chemischen Systems beschreiben oder charakterisieren. Dadurch werden bspw. die Konzentration von Wasserstoffionen oder Hydroniumionen in einer Lösung gemessen (Masterton et al., 2012, S. 101).

Konventionell wird der Farbwert eines pH-Indikators in Form eines Papierstreifens, der mit einer pH-empfindlichen Lösung benetzt ist, mit einer Farbreferenz (*Farbskala*) verglichen, um den pH-Wert einer Substanz zu bestimmen. Verschiedene Indikatoren zeigen einen Wechsel zwischen zwei oder mehreren Farben an. Sie zeigen ihre Farben in Abhängigkeit vom pH-Wert in sauren, neutralen oder alkalischen Lösungen oder Materialien. Halochrome Verbindungen können mit anderen Materialien als Verbundwerkstoffe eingesetzt werden, um das konventionelle Anwendungsspektrum mit einer Bandbreite an Funktionalitäten zu erweitern. Dementsprechend können Sensoreigenschaften in ein vorhandenes System integriert werden, in dem unter anderem schädliche Faktoren auf einer Oberfläche in Form einer intelligenten Beschichtung nachgewiesen, farblich gekennzeichnet oder ein Oberflächenheilungsprozess an den betroffenen Teilen eingeleitet werden kann (Bilgin, 2021). Halochrome Verbindungen können jedoch auch eingesetzt werden, um das Eindringen von Wasser zu erkennen und zu verhindern. Außerdem kann in Verbundverpackungen die Verzehreignung verifiziert werden, wodurch bspw. Produkte wie Milch oder Fleischwaren farblich hinsichtlich ihrer Verzehrbeschränkungen gekennzeichnet werden.

#### 4.6.3.1 Farbveränderung

Säure-Base-Indikatoren sind schwache Säuren oder Basen, die Anionen oder Kationen bilden können. Farbveränderungen beruhen auf der Deprotonierung, der Abspaltung von Protonen, einer Farbsäure oder der Protonierung, Anlagerung von Protonen, einer Farbbase. Dementsprechend wird eine Farbveränderung durch unterschiedliche Portionierungszustände – unterschiedliche Ladungszustände bei unterschiedlichen pH-Werten (Schwartz, 2002, S. 497) – bewirkt. Dabei beruht die Farbveränderung auf dem durch den Protonenaustausch hervorgerufenen Strukturänderungen der Verbindungen (Bamfield & Hutchings, 2001, S. 41). Im sauren Bereich sind diese stärker protoniert als im alkalischen Bereich (Carey & Sundberg, 2008, S. 647 ff.). Abhängig von der Anzahl der übertragenen Protonen werden dem Zielmolekül eine oder mehrere positive Ladungen hinzugefügt (*protoniert*). Infolgedessen kann die Buntheit auf die unterschiedlichen Ladungszustände bei unterschiedlichen pH-Werten zurückgeführt werden, da Verbindungen bei der Aufnahme oder Abgabe von Protonen ihre Farbe ändern (Campbell & Farrell, 2008, S. 47)

#### 4.6.3.2 Halochrome Mechanismen

Wenn eine funktionelle Hydroxygruppe (-OH) eines Phenols, als Verbindung an eine aromatische Kohlenwasserstoffgruppe gebunden ist, also deprotoniert wird, nehmen die dabei entstehenden freien Elektronenpaare der Phenolatgruppe stärker an Resonanzstrukturen teil, sodass sich das Absorptionsmaximum aufgrund des negativen Ladungsüberschusses vom unsichtbaren (*UV*) in den sichtbaren Teil des Spektrums (*VIS*) verschiebt (Nassau, 1983, S. 126; 347 ff.). Je stärker die Elektronen delokalisiert sind, desto leichter können sie die Energie des Lichts absorbieren (Scotter, 2015, S. 4). Aufgrund des halochromen Effekts können einige Farbstoffe als Säure-Base-Indikatoren verwendet werden, z. B. Phenolphthalein, Methylrot, Methylorange oder Bromthymolblau-Farbstoffe. Einige chemische Verbindungen sind farblos und nur bei einem bestimmten pH-Wert sichtbar (z. B. *Kristallviolettlacton*). Phenolphthalein erscheint auch in leicht saurer oder neutraler Umgebung farblos (Abbildung 42), während es in alkalischer Umgebung eine violette Farbe zeigt (Nassau, 2001, S. 125). In einem schwach alkalischen Milieu ( $\text{pH} > 7$ ) setzt eine phenolische Hydroxylgruppe von Phenolphthalein ein Proton frei, das durch Elektronenaufnahme oder -freisetzung von Wasserstoffionen  $\text{H}^+$  (*Protonen*) gebildet wird und zu einem Anion führt.

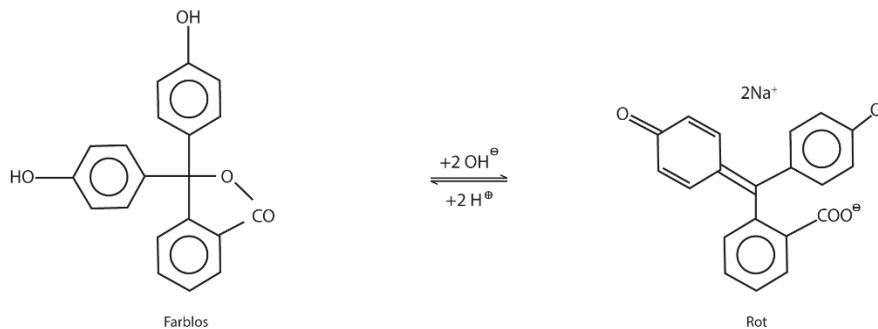


Abbildung 42 Halochrome Farbveränderung von Phenolphthalein.

Cyanidin<sup>4</sup>, ist eine in der Natur vorkommende organische Verbindung sowie ein Farbstoff und eignet sich als Indikator (Blank, 1947). Der Farbstoff gehört zur Gruppe der Anthocyane; Pelargonidin ( $\text{C}_{15}\text{H}_{11}\text{O}_5^+$ ), Cyanidin ( $\text{C}_{15}\text{H}_{11}\text{O}_6^+$ ), Delphinidin ( $\text{C}_{15}\text{H}_{11}\text{O}_7$ ) und Malvidin ( $\text{C}_{17}\text{H}_{15}\text{ClO}_7$ ) sind Bestandteil dieser Gruppe (Abbildung 43). Anthocyanidine gehen eine pH-bezogene Reaktion ein, wobei jede Struktur eine andere Farbe zeigt. Mit zunehmender Anzahl von Hydroxylgruppen (grau) wechselt die Farbe von Rot nach Blau. Bei vollständiger Entfernung der Hydroxylgruppen kommt es zu einer Farbverschiebung nach Gelb (Onslow, 1925).

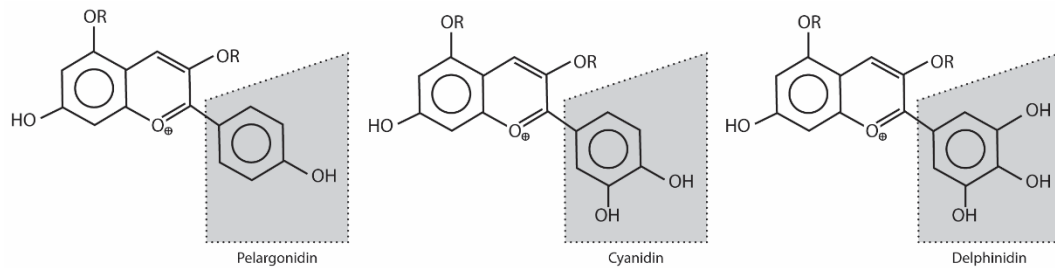


Abbildung 43 Gruppe von Anthocyanen mit typischen Strukturvariationen

<sup>4</sup> Wird häufig mit Cyanide (giftig) verwechselt.

Die Farbgebung von Cyanidin (Abbildung 44) reicht von rötlich-violett (pH < 3) über Violett (pH 6–7). Hierbei bildet das Cyanidin-Molekül ein Proton und liefert damit ein Cyanidin-Kation und reicht weiter zu Blau (pH 7–8). Das Cyanidin-Molekül setzt ein weiteres Proton frei (*Deprotonierung*) und ändert seine Farbe weiter zu Grün (pH 9–11). Bei der Freisetzung eines zweiten Protons kommt es zur Bildung eines (Di-)Anion. Dies führt zur gelben Farbausbildung (pH > 12). Gelb indiziert eine irreversible Zerstörung von Cyanidin. Das Molekül wird irreversibel in ein gelbes Chalkon-Anion umgewandelt. Die Farbveränderung des roten Cyanidin beruht also auf zwei OH-Gruppen, die Protonen freisetzen. Cyanidin kommt in vielen verschiedenen Pflanzen in glykosidischer Form (als Anthocyanin), z. B. in Rotkohl, Kirschen, Himbeeren, Trauben und Brombeeren vor (Bechtold et al., 2007). Halochrome Eigenschaften sind ebenfalls in der Blüte der Kornblume festzustellen, innerhalb dieser der Anthocyanidin-Farbstoff chemisch an andere Ionen wie  $\text{Fe}^{3+}$  oder  $\text{Al}^{3+}$  gebunden ist.

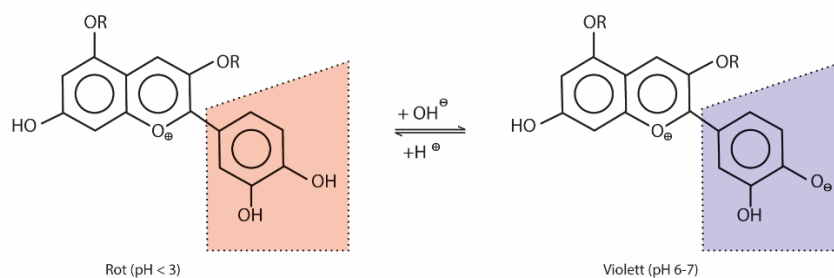


Abbildung 44 Halochrome Formen von Cyanidin / R = glycosid

#### 4.6.3.3 Auswahl Halochromer Verbindung

Um den definierten Zielerfordernissen entsprechend eine irreversible halochrome sensorische Farbe zu entwickeln, wurden in einem vorherigen Prozess verschiedene halochrome Materialien auf ihre Eigenschaften hin untersucht. Viele Materialien eignen sich nicht, da sie lediglich in einer gelösten Form als pH-Indikator wirken. Diese Eigenschaft geht jedoch durch Trocknen nach dem Verdrucken oder Auftragen auf ein Substrat verloren. Hervorragende Ergebnisse konnten mit Cyanidin, aus der Gruppe von Anthocyanen und Diferuloylmethan erzielt werden (Abbildung 45), bei der die Funktionalität nach dem Trocknen erhalten blieb. Diferuloylmethan gehört zur Gruppe der Hydroxyketon-Verbindungen und besitzt sowohl in der Keto- als auch in der Enol-Form zwei Strukturen. Hierbei kann die (Di-)Ketogruppe (grau) ebenfalls zur Bildung eines Beizenfarbstoffs verwendet werden, indem das einflussnehmende Metallsalz die Farbe bestimmt („Aluminium: orangegelb, Chrom: braun, Zinn: orangeroth, Eisen: braun“) (Klöckl, 2015, S. 281). Bei neutralem (pH 7) und saurem (pH < 8) Milieu weist die chemische Verbindung des Diferuloylmethan, fluoreszierende Eigenschaften auf und emittiert dabei gelbes Licht. Der pH-Wert bedingte Farbumschlag liegt zwischen pH 8 – 9. Dabei weist die chemische Struktur eine CH-Acidität auf, wodurch ein Proton verloren und eine Strukturänderung innerhalb der (Di-)Ketogruppe bewirkt wird, bei der die Absorption des sichtbaren Lichts verstärkt wird. Dementsprechend ist bei einem pH-Wert um 8 (Natriumbicarbonat) ein roter und bei einem pH-Wert von ca. 14 (Natriumhydroxid) ein rotbrauner Farbumschlag zu erkennen.

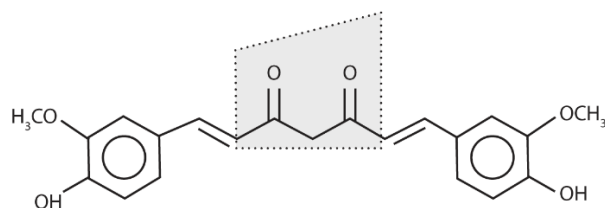


Abbildung 45 Strukturform von Diferuloylmethan / Gepunktet: (Di-)Ketogruppe

#### 4.6.3.4 Interpretation

Der halochrome Farbveränderungsprozess kann unter anderem einerseits visuell mit einer einfachen Farbvergleichsmethode, bspw. einer Nachschlagetabelle, festgestellt werden. Andererseits kann dieser mit einem optischen Lesegerät bspw. unter Verwendung eines Bilderkennungsalgorithmus oder mit einer digital hinterlegten Referenz in einer Datenbank verglichen werden. Entsprechend kann ein Hinweis auf den Grad einer Veränderung oder eines Einflussfaktors gefunden werden. Die Interpretation kann dabei wie in den *Kapiteln 4.6.1.5* und *4.6.2.5* beschrieben mit der Farbabstandsformel  $\Delta L^*_{76}$  erfolgen.

#### 4.6.3.5 Lösungsmittel Identifizierung

Halochrome Verbindungen können unter anderem in Form von Pigmenten, Farbstoffen oder Mikroverkapselung in einen Träger (*Matrix*) eingebettet werden. Das dehydrierte Cyanidin wurde als wasserlösliches Pulver in demineralisiertem Wasser gelöst. Das Diferuloylmethan wurde aufgrund seiner Unlöslichkeit in Wasser, in Ethanol (Löslichkeit von 3 g/l bei 20 °C) gelöst (RÖMPP, 2020). Um eine annähernd intensive Farbdeckung zu erzielen, wurden drei Konzentrationen des Diferuloylmethan mit jeweils 60 mg, 80 mg und 100 mg in 25 ml Ethanol gelöst. Laut RÖMPP (2020) besteht eine geeignete Löslichkeit bei 75 mg Diferuloylmethan in 25 ml Ethanol, dies erwies sich während einer Voruntersuchung als vernachlässigend, da sich Diferuloylmethan auch in höheren Konzentrationen lösen ließ und damit eine intensivere Farbdeckung erzielt wurde, auch wenn die Grenzen der maximalen Löslichkeit der identifizierten Quellen überschritten wurde.

#### 4.6.3.6 Stoffkonzentrationen

Im Folgenden werden verschiedene Konzentrationen der beiden halochromen sensorischen Farben aufgelistet (Tabelle 21), mit denen experimentell gearbeitet wurde. Die Zusammensetzungen wurden in Voruntersuchungen identifiziert, welche eine kontrastreiche Farbintensität des halochromen Stoffes fokussierten. Im späteren *Kapitel 4.7.3* wird nur noch Diferuloylmethan als exemplarisches Beispiel verwendet, da die Aufnahme weiterer halochromer Farbstoffe den Rahmen dieser Arbeit übersteigen. So können je nach pH-Bereich, unterschiedliche halochrome Materialien eingesetzt werden.

Nr.	Farbstoffgehalt	Lösungsmittelgehalt
1.1	0,06 g Diferuloylmethan	25 ml Ethanol
1.2	0,08 g Diferuloylmethan	25 ml Ethanol
1.3	0,1 g Diferuloylmethan	25 ml Ethanol
1.1	0,125 g Cyanidin	25 ml dem. Wasser
1.2	0,250 g Cyanidin	25 ml dem. Wasser

Tabelle 21 Halochrome Stoffkonzentrationen

#### 4.6.3.7 Lichtecktheit

Diferuloylmethan und Cyanidin weisen keine Lichtecktheit und somit keine farbliche Beständigkeit auf (RÖMPP, 2020). Dieser Umstand muss bei einer Farbauswertung berücksichtigt werden. Damit das Defizit der Lichtecktheit beim Diferuloylmethan kompensiert wird, wurde bereits von Zhou, Y. und Tang (2016) ein reaktiver UV-Absorber (*Lichtschutzmittel*) auf Oxalanilide-Basis erfolgreich eingesetzt. Die Lichtecktheit der Gruppe von Anthocyanen wurde von Bechtold et al. (2007, S. 2593) untersucht, bei der die schwache Lichtecktheit nachweislich verifiziert wurde. Wang, H. et al. (2017, S. 15) fanden heraus, dass das Resultat der Lichtecktheit mit steigender Farbauftragsstärke für gefärbte Gewebe einen akzeptablen Bereich annimmt. So beeinflussen Faktoren wie Bindemittel, Substrat, Pigmentvolumenkonzentration, Schichtdicke und Additive die Lichtecktheit (Herbst und Hunger, 1995, S. 96).

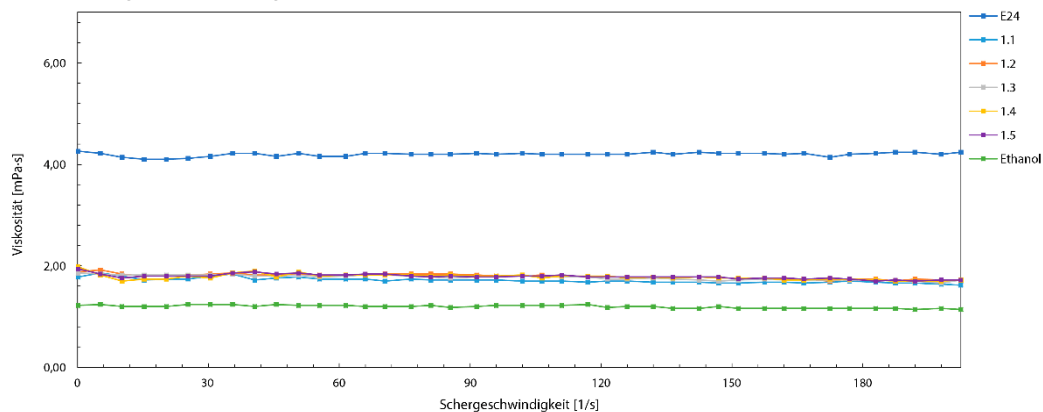
## 4.7 Entwicklung sensorischer Tinten

Dieser Abschnitt setzt sich mit der Bestimmung von Anforderungen an geeignete sensorische Tinten auseinander. Für die in *Kapitel 4.6* behandelten *sensorischen Farben* werden in diesem Kapitel Tinten formuliert, welche die sensorischen Farben zu sensorischen Tinten machen. Nachfolgend werden Tintenparameter wie Viskosität, Oberflächenspannung und Partikelgröße näher untersucht und die dafür notwendigen Zielparameterfenster definiert. Die jeweiligen Parameter der sensorischen Tinten werden dabei für den verwendeten Druckprozess angepasst und die sensorische Tinte abschließend einem mehrstufigen Filtrationsprozess (*Kapitel 3.2.4*) unterzogen.

### 4.7.1 Hydrochrom

#### 4.7.1.1 Untersuchung der rheologischen Eigenschaften

Wie bereits in *Kapitel 3.2.1* beschrieben, werden die rheologischen Eigenschaften, insbesondere das Fließverhalten des hydrochromen Sensors, analysiert. *Abbildung 46* zeigt die Scherviskosität, gemessen mit einem Kegel-Platte-Rheometer, für den hydrochromen Sensor bei unterschiedlichen Konzentrationen. Es ist hervorzuheben, dass alle Tinten und Lösungsmittel ein relativ lineares Fließverhalten mit annähernd gleicher niedriger Viskosität aufweisen.



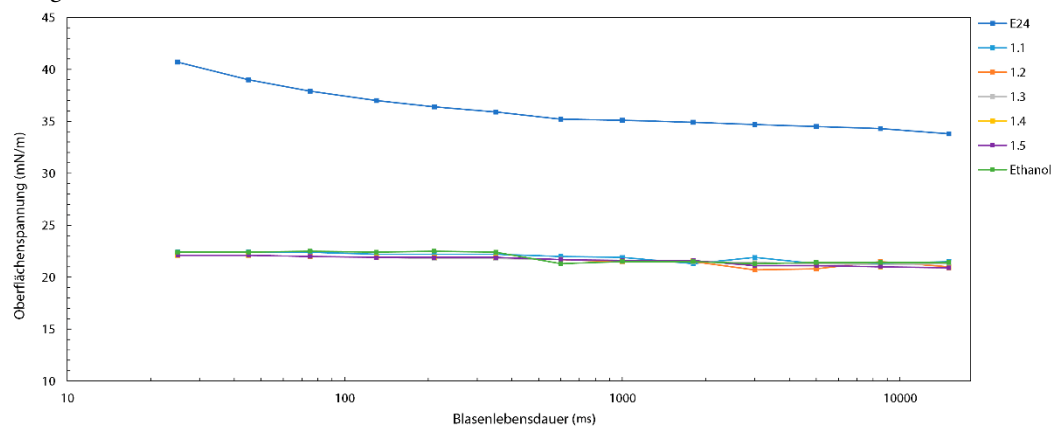
*Abbildung 46 Viskositätsanalyse hydrochromer Sensoren; Konzentrationen nach Tabelle 19*

So liegt die Viskosität der wasserbasierten Basistinte E24 bei 4,24 mPa·s. Ethanol, das als Lösungsmittel für den hydrochromen Sensor dient, bei 1,32 mPa·s und weist von allen Materialien den niedrigsten Viskositätswert auf. Die Viskositäten der unterschiedlichen Konzentrationen der hydrochromen Sensoren bewegen sich um den Wert von 1,89 mPa·s und liegen über dem Wert des verwendeten Lösungsmittels Ethanol. Die höhere Viskosität der Tinten ist auf die Bestandteile Natriumbicarbonat und Patentblau V zurückzuführen. Wenn beide Komponenten (Natriumbicarbonat und Patentblau V) in Ethanol eingebettet werden, werden beide getrennt voneinander im Lösungsmittel verteilt.



#### 4.7.1.2 Untersuchung der Oberflächenspannung

Nachfolgend wird die Oberflächenspannung entsprechend der im *Kapitel 3.2.2* definierten Methode untersucht. Das Messinstrument, der Blasendruck-Tensiometer wurde für die dynamische Bestimmung der Benetzungseigenschaften eingesetzt. Resultierend zeigt *Abbildung 47* den zeitlichen Verlauf der Oberflächenspannung für Tinten unterschiedlicher Zusammensetzung. Ziel war es, die Oberflächenspannung des hydrochromen Sensors hinsichtlich der Eigenschaften und ihrer Eignung für das Inkjetdruckverfahren zu verifizieren. Der Mittelwert der Temperatur während den Messungen der Oberflächenspannung lag bei etwa  $\vartheta = 21,6 \text{ }^\circ\text{C}$ . Wie in der *Abbildung* dargestellt, liegen die Oberflächenspannungen aller Konzentrationen bei etwa  $21,5 \text{ mN/m}$  recht nahe beieinander. Demzufolge hat in diesem Fall die Zusammensetzung der sensorischen Tinte keinen signifikanten Einfluss auf die Änderung der Oberflächenspannung. Die Oberflächenspannung der Basistinte E24 weicht erwartungsgemäß stark von den übrigen Tinten ab. Sie basiert auf Wasser, das aufgrund seiner Polarität eine hohe Oberflächenspannung besitzt, während die formulierten Tinten auf Ethanol, bei etwa  $22,55 \text{ mN/m}$ , basieren.



*Abbildung 47* Oberflächenspannung bei verschiedenen Konzentrationen hydrochromer Sensoren

Die Ergebnisse der Untersuchung der Viskosität und Oberflächenspannung werden zusammengefasst (*Abbildung 48*). Ziel war es, die rheologischen Eigenschaften und die Oberflächenspannung der hydrochromen Sensoren zu untersuchen und ihre Eigenschaften für das Inkjetdruckverfahren einzustellen. Dazu wurden Zielparameterfenster (*Abb.*, grau) zum einen für die Viskosität mit einem Bereich von  $1 \text{ mPa}\cdot\text{s}$  bis  $25 \text{ mPa}\cdot\text{s}$  (*Magdassi, 2010, S. 35*) und zum anderen für die Oberflächenspannung ein Bereich von  $20 \text{ mN/m}$  bis  $50 \text{ mN/m}$  (*ebd., S. 35; Hutchings & Martin, 2013, S. 24*) definiert. Die wasserbasierte Basistinte E24 diente dabei als zusätzlicher Referenzwert und liegt mit einer Viskosität von  $4,24 \text{ mPa}\cdot\text{s}$  und einer Oberflächenspannung von  $37,08 \text{ mN/m}$  ebenfalls im empfohlenen druckbaren Bereich.

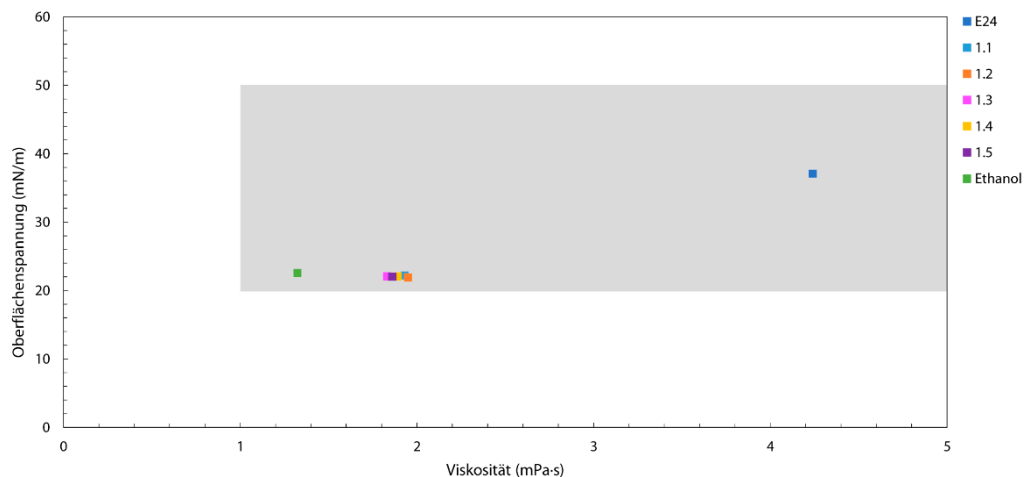


Abbildung 48 Hydrochromer Sensor und dessen Zielparameterfenster

In der Abbildung ist zu erkennen, dass die hydrochromen Sensoren verschiedener Konzentrationen (1.1 – 1.5) ebenfalls im Zielparameterfenster, mit einer durchschnittlichen Viskosität von 1,89 mPa·s und Oberflächenspannung bei etwa 21,5 mN/m, innerhalb des druckbaren Bereichs liegen.

#### 4.7.1.3 Untersuchung der Partikelgrößenverteilung

Im Kapitel 3.2.3 wurde auf die Untersuchung der Verteilung ungelöster Partikel hingewiesen. Zudem wurde bereits das Kristallisationspotential und die mäßige Löslichkeit von Patentblau V und dem zugehörigen Reduktionsmittel in Ethanol erwähnt (Kapitel 4.7.1). Entsprechend wird bei den unterschiedlichen Konzentrationen (Tabelle 19) die Verteilung ungelöster Partikel, die in der Tinte als feste Partikel suspensieren, untersucht (Abbildung 49). Die Untersuchung erfolgte mit einem Konfokalmikroskop durch eine Bildanalyse nach der Methode der mikroskopischen Partikelzählung (Kapitel 3.2.3). Hierbei unterscheidet sich die methodische Vorgehensweise bei der Art des Auftragens zum im Methodenteil beschriebenen Vorgehen, bei der jeweils 0,1 ml der hydrochromen Stoffkonzentrationen (Tabelle 19) in 2 ml Ethanol (1:20) verdünnt und eine Referenzskala von 10 µm gewählt wurde.

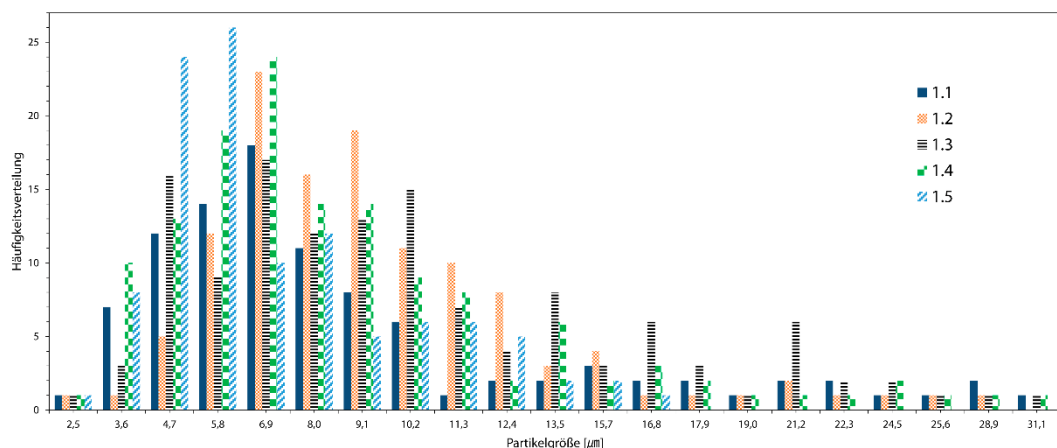


Abbildung 49 Partikelgrößenverteilung der Konzentrationen hydrochromer Sensoren

Die messtechnische Untersuchung gibt erste Hinweise auf die Partikelgrößenverteilung der ungefilterten Proben. Abbildung 49 zeigt eine Verteilung der ungelösten Partikel in den einzelnen Konzentrationen 1.1 – 1.5. Diesbezüglich fanden sich aus der repräsentativen Stichprobe Partikelgrößen in einer Bandbreite von 1,4 µm bis 31,1 µm. Die vorliegende Partikelverteilung dokumentiert zum einen den Anteil

an nahezu unlöslichem Natriumbicarbonat (Reduktionsmittel) in Ethanol und weist zum anderen auf eine Übersättigung der Lösung Patentblau V hin. Dahingehend weist die Konzentration 1.1 eine erhöhte Partikelgrößenverteilung zwischen 19  $\mu\text{m}$  und 31  $\mu\text{m}$  auf, wohingegen die Konzentration 1.5 ab 15  $\mu\text{m}$  stetig sinkt und bei ca. 23  $\mu\text{m}$  komplett abflacht. Darüber hinaus weicht die Größe der Partikel erwartungsgemäß von der empfohlenen Partikelgröße für eine geeignete Inkjetfähigkeit von weniger als 200 nm bis 300 nm (Magdassi, 2010, S. 34) erheblich ab, da das Reduktionsmittel nahezu unlöslich ist.

Bei einer vergrößerten Betrachtung mit einem Rasterelektronenmikroskop (REM) sind die Partikel zum einen als ungelöste Patentblau V Farbstoffpartikel zu erkennen und zum anderen als ungelöste Natriumbicarbonat Feststoffpartikel. Diese überdecken flächenweise die Filtermembrane (PES, 0,22  $\mu\text{m}$ ) und führen aus diesem Grund ohne Vorkehrungen (Filtrationsprozess) im späteren Verlauf zum Verstopfen der Düsen des eingesetzten Inkjetsystems (Abbildung 50). So lässt sich aufgrund der Faktoren Kristallisationspotential, Löslichkeit und Sättigung der Lösung begründen, dass vor der Verwendbarkeit des hydrochromen Sensors eine Ultrafiltration durchgeführt werden muss, bevor sie als inkjetdruckfähige Tinte eingesetzt werden kann. Die Funktionalität des hydrochromen Sensors ist dabei nach einem Filtrationsprozess immer noch gewährleistet, da genügend Farbstoffanteile in der Lösung gelöst wurden. Auf die Filtration wurde bereits in *Kapitel 3.2.4* eingegangen.

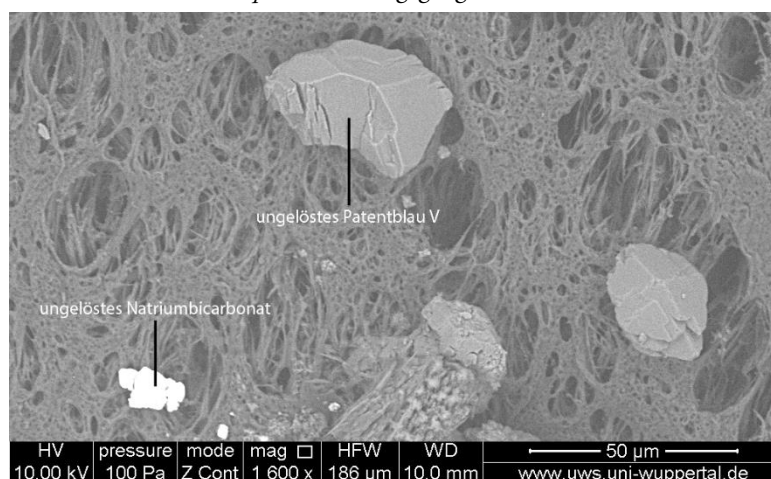


Abbildung 50 Hydrochrome Partikel auf einem 220 nm Filtermembrane

#### 4.7.1.4 Spektralphotometrische Untersuchung

Im Folgenden wird die spektralphotometrische Verteilung des hydrochromen Sensors anhand von Remissionskurven vor (türkis) und nach einer Kontamination (blau) dargestellt (Abbildung 51). Die Charakteristik des nicht kontaminierten hydrochromen Sensors ist nahezu identisch mit dem Wellenlängenbereich von Patentblau, der sich bei 425 nm bis ca. 560 nm verortet und eine charakteristische Spitze bei etwa 465 nm aufweist. Im reagierten Zustand (nach dem Einfluss von Wasser) verbreitert sich die Spitze im Bereich von 400 nm bis 640 nm und erscheint dadurch heller. Dieser Farbveränderungsprozess lässt sich ebenfalls durch die  $L^*a^*b^*$ -Werte darstellen. Die Farbveränderung kann zum einem mittels  $L^*$  und  $b^*$ -Wert vorher und nachher messtechnisch quantifiziert werden.

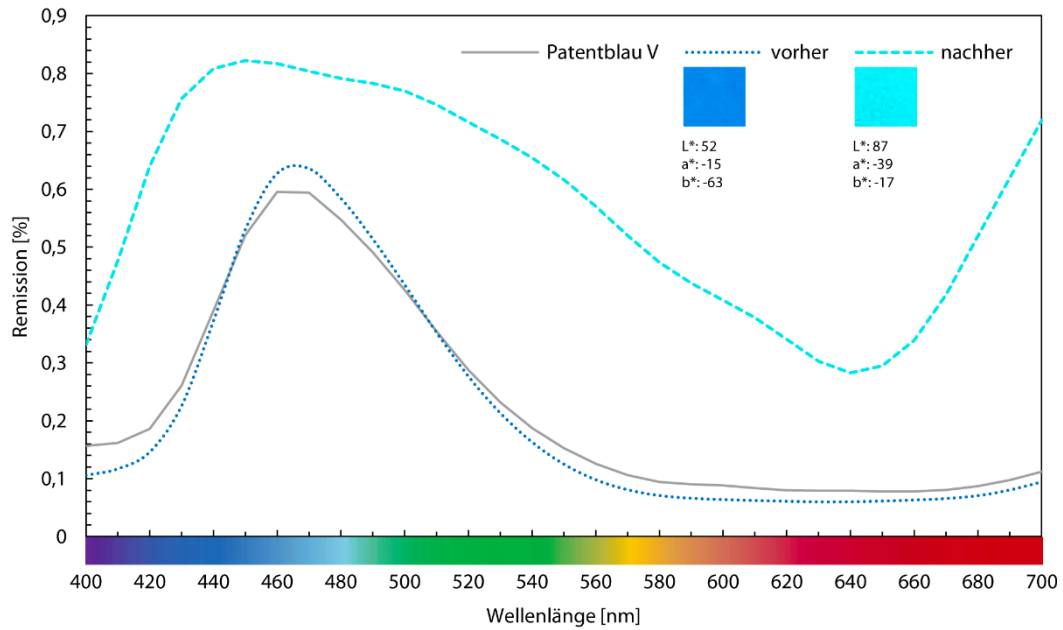


Abbildung 51 Spektralphotometrische Untersuchung des hydrochromen Sensors

## 4.7.2 Photochrom

### 4.7.2.1 Untersuchung der rheologischen Eigenschaften

Die rheologischen Eigenschaften der unterschiedlichen Konzentrationen der photochromen Sensoren wurden wie in *Kapitel 3.2.1* beschrieben, ermittelt (Abbildung 52). So ist hervorzuheben, dass auch hier neben der Basistinte E24 alle anderen wasserbasierten Tinten ein relativ lineares Fließverhalten mit einer niedrigen Viskosität aufweisen. Im Vergleich zum hydrochromen Sensor auf Ethanol Basis, liegt die Scherviskosität des wasserbasierten photochromen Sensors erwartungsgemäß höher.

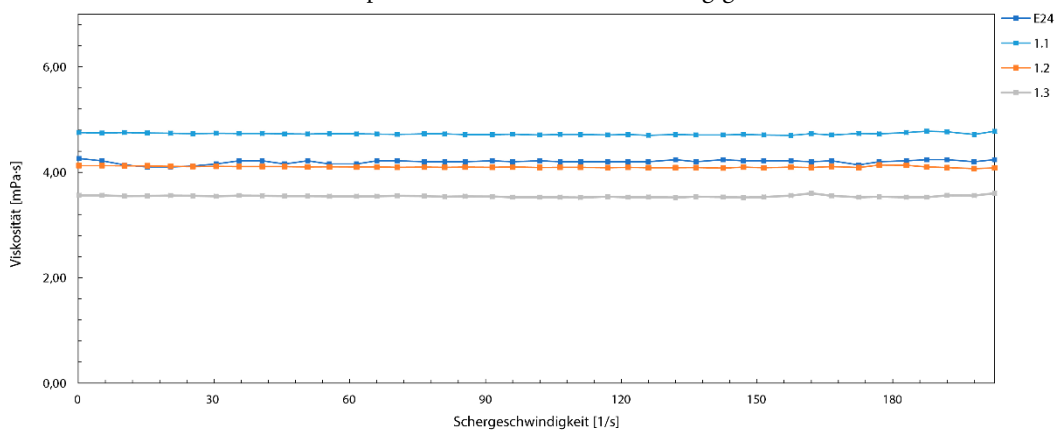


Abbildung 52 Viskositätsanalyse aller Konzentrationen photochromer Sensoren

Die Viskositäten der Konzentrationen 1.1 und 1.2 liegen gleichbleibend bei ca. 4,15 mPa·s. Die Konzentration 1.3 weist ebenfalls ein strikt newtonsches Verhalten, bei einer Viskosität von 3,5 mPa·s auf. Der im Vergleich relativ niedrige Viskositätswert dieser Konzentration kann mit der Zunahme der Stoffanteile (Tabelle 20) in der Lösung in Zusammenhang gebracht werden (Willenbacher & Georgieva, 2013, S. 38). Die Partikelanzahl in der Lösung hängt im Falle des photochromen Sensors mit der Sät-

tigung der Lösung zusammen. Hierbei führt die Abnahme der Partikelgröße in der Regel zu einer höheren Viskosität (1.1 und 1.2) und die Zunahme der Partikelgröße (1.3) zu einer niedrigeren Viskosität (ebd.; Anton Paar, 2021). Letzteres ist auf die Erhöhung des Stoffgehalts, die zu einer Annäherung an eine Sättigung der Lösung führt, zurückzuführen.

#### 4.7.2.2 Untersuchung der Oberflächenspannung

Im nachfolgenden wird die Oberflächenspannung untersucht. Abbildung 53 stellt die Oberflächenspannung der verschiedenen Konzentrationen dar. Ziel war es auch hier, die Oberflächenspannung des photochromen Sensors hinsichtlich der Eigenschaften und ihrer Eignung für das Inkjetdruckverfahren einzustellen. Um dies zu erreichen, wurden Additive als Oberflächenspannungsregler, eingesetzt (Kapitel 3.1.1). Die mittlere Temperatur bei der Messung der Oberflächenspannung lag bei etwa  $\vartheta = 20,5\text{ }^{\circ}\text{C}$ . So konnte die Oberflächenspannung von den Konzentrationen, die zuvor bei nahezu  $72\text{ mN/m}$  (Wasser) lagen, herabgesetzt werden. Wie in der Abbildung dargestellt, liegt die Oberflächenspannung aller Konzentrationen mit durchschnittlich  $46\text{ mN/m}$  nahe beieinander. Hervorzuheben ist jedoch, dass der Abstand der Konzentrationen zum Referenzwert der E24 wasserbasierten Basistinte ( $37,08\text{ mN/m}$ ) leicht vergrößert ist.

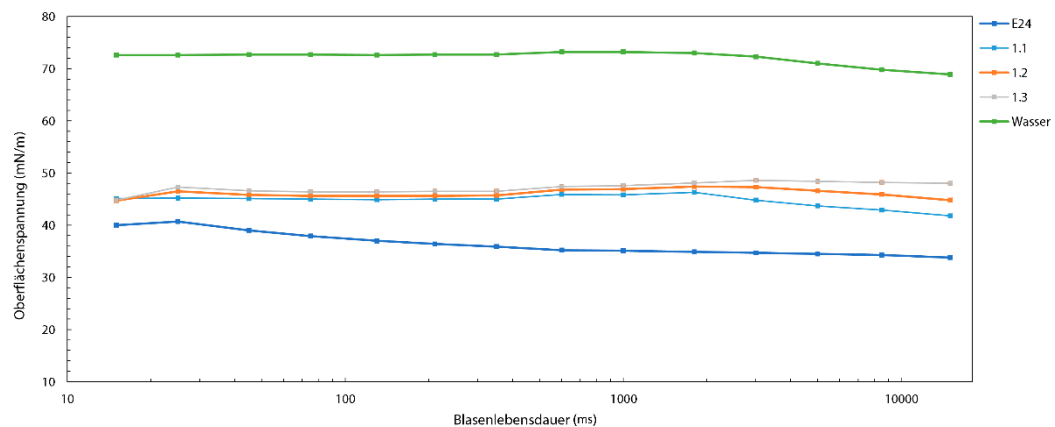


Abbildung 53 Oberflächenspannung der verschiedenen Konzentrationen photochromer Sensoren

Die Ergebnisse der Untersuchung der Viskosität und Oberflächenspannung werden zusammengefasst und entsprechend der Zielsetzung mit Hilfe eines Zielparameterfensters für das eingesetzte Inkjet-druckverfahren angepasst (Abbildung 54). Das Zielparameterfenster (Abb., grau) wurde auch hier entsprechend des empfohlenen Bereichs der Viskosität, der zwischen  $1\text{ mPa}\cdot\text{s}$  und  $25\text{ mPa}\cdot\text{s}$  (Magdassi, 2010, S. 35) liegt und den der Oberflächenspannung, der zwischen  $20\text{ mN/m}$  und  $50\text{ mN/m}$  (ebd., S. 35; Hutchings & Martin, 2013, S. 24) liegt, festgelegt. Auch hier wird die E24 Basistinte, welche als Referenzwert dient, eingezeichnet. Die Viskositäten aller Konzentrationen liegen mit einem Wert von etwa  $4,15\text{ mPa}\cdot\text{s}$  (ausgenommen von Konzentration 1.3) und mit einer Oberflächenspannung von etwa  $46\text{ mN/m}$  in dem Zielparameterfenster. So liegt die Viskosität ebenfalls im von Zapka (2018, S. 12) empfohlenen druckbaren Bereich von  $1$  bis  $50\text{ mPa}\cdot\text{s}$ .

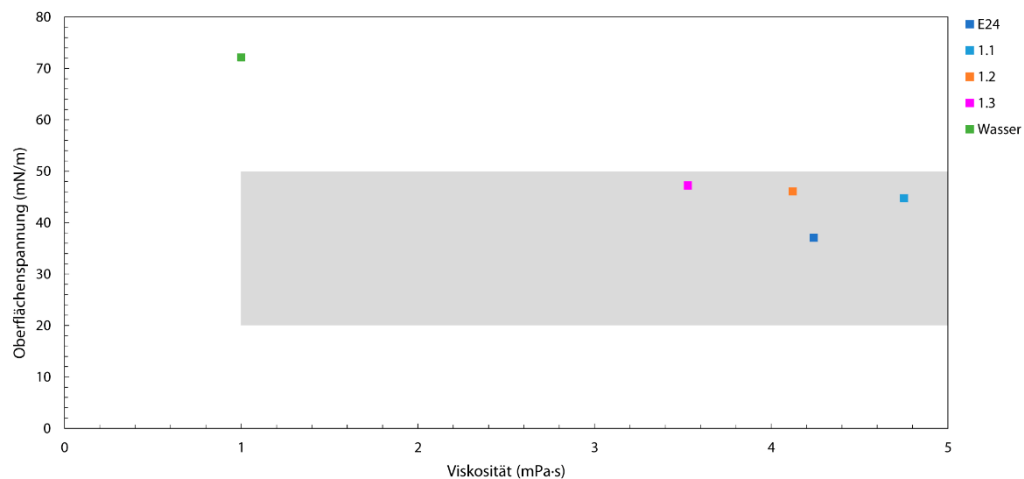


Abbildung 54 Photochromer Sensor und dessen Zielparameterfenster

#### 4.7.2.3 Untersuchung der Partikelgrößenverteilung

Vor einer Partikelanalyse wurde in einer Veröffentlichung (Bilgin & Backhaus, 2018a, S. 59) der Hypothese einer Viskositätsänderung, bedingt durch eine mögliche UV-Licht-Kontamination, nachgegangen, da zuvor eine leichte Sedimentation festgestellt wurde. Hierzu wurde die Viskosität der photochromen Sensoren der gleichen drei Konzentrationen (1.1. – 1.3.) wie in Tabelle 20 in belichtetem und unbelichtetem Zustand untersucht. Die drei Konzentrationen wurden hierzu 60-minütig in direktem Kontakt mit der UV-Lichtquelle (*Kapitel 3.5*) belichtet. Nach dem Belichtungsvorgang war eine signifikante Viskositätssteigerung bei allen Konzentrationen messbar, jedoch lagen die gemessenen Viskositäten noch im druckbaren Bereich von 1 bis 50 mPa·s (Zapka, 2018, S. 12). Die Belichtung der drei Konzentrationen führte zu einer leichten Niederschlagsbildung, die mit einer Erhöhung der Partikelgrößenverteilung einherging. Entsprechend wurde diesem Hinweis nachgegangen und es konnte eine Niederschlagsbildung bedingt durch einen vorhergehenden Belichtungsvorgang nachgewiesen werden, bei dem das Pigment Preußisch Blau gebildet wurde. Das Pigment Preußisch Blau weist nach Kraft (2012, S. 3) „in Wasser eine sehr geringe Löslichkeit“ auf und bildet bei einer Fällungsreaktion einen Niederschlag aus (Klöckl, 2015, S. 234). Es wird darauf hingewiesen, dass die gefällten Primärteilchen mit nahezu 50 nm relativ klein sind (ebd.).

Nachfolgend wurde eine mikroskopische Partikelanalyse der drei Konzentrationen (1.1. – 1.3.) aus der Tabelle 20 durchgeführt (*Kapitel 3.2.3*). Die Ergebnisse der mikroskopischen Untersuchung mit einem Konfokalmikroskop verdeutlichen (Abbildung 55), dass die drei Konzentrationen des photochromen Sensors eine gute Löslichkeit aufweisen, so dass mit dem Einsatz dieses Messinstrumentes kleine ungelöste Feststoffanteile festzustellen waren. In der Abbildung ist bei allen drei Konzentrationen im unbelichteten Zustand eine nahezu homogene und kristalline Oberfläche zu erkennen. Nach einer UVA Lichtexposition (Kontamination) mit einer Leistung von  $0.3 \text{ mW/cm}^2$  (Abbildung 16) ist eine stetige Blaufärbung der kristallinen Oberfläche, vor allem an den Randbereichen der kristallinen Struktur, zu erkennen (*unten rechts*).

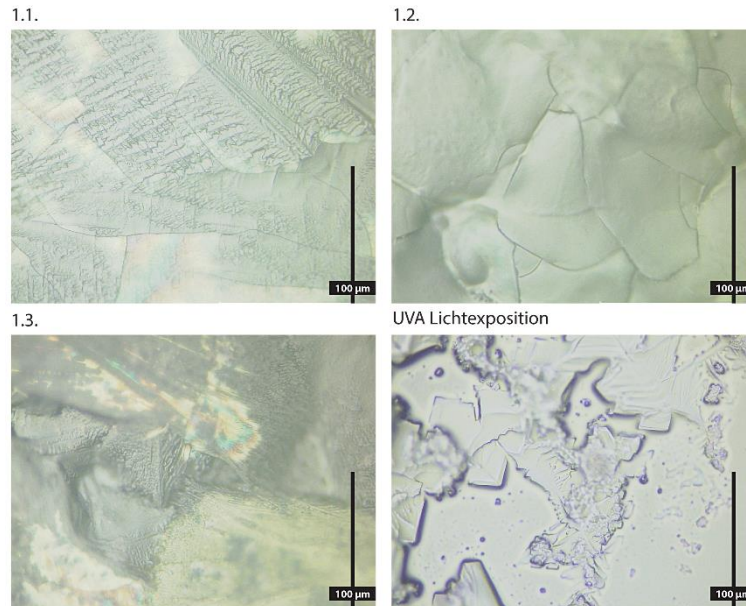


Abbildung 55 Mikroskopische Aufsichtsaufnahme des photochromen Sensors

Bei einer weiteren vergrößerten Betrachtung mit einem Rasterelektronenmikroskop finden sich nach dem Filtern der höchsten Konzentration (1.1.) kaum Restpartikel (rot eingekreist) auf der Filtermembrane (PES, 0,22 µm) (Abbildung 56). Dieses Ergebnis ist auf eine gute Löslichkeit von 1200 g/l bei 20 °C zurückzuführen (Kapitel 4.6.2).

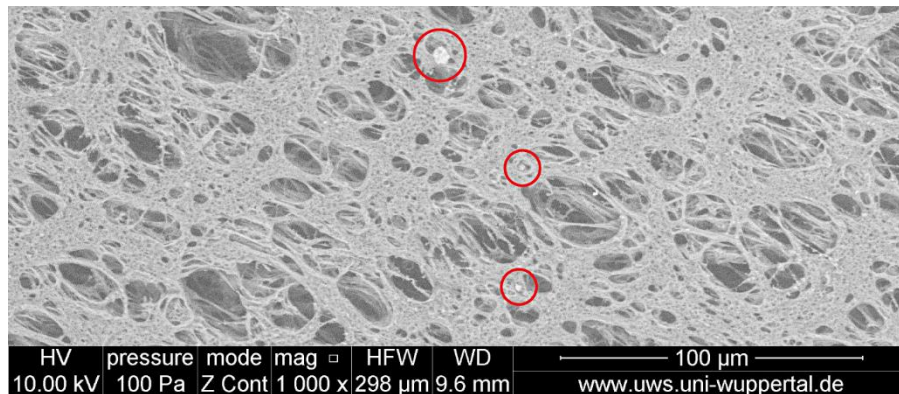


Abbildung 56 REM Aufnahme von der Filtermembran aufgefangenen photochrome Partikel

#### 4.7.2.4 Farbveränderungsverhalten

Um die Funktionalität und entsprechend das Farbveränderungsverhalten des photochromen Sensors zu untersuchen, wurden die beschriebenen Lichtquellen (Kapitel 3.5) verwendet. Das Ziel war es, die Belichtungsdauer der jeweiligen Lichtquelle mit bekanntem Spektrum auf das unidirektionale Farbveränderungsverhalten zu analysieren. In Abbildung 57 sind die belichteten Flächen zu verschiedener Dauer der Belichtung zu sehen. Hierbei sind die noch nicht aktivierten Flächen gelbgrün (0 min), die allmählich in eine blaue Farbe und somit in einen aktivierten Zustand (60 min) wechseln. Die in der Abbildung dargestellten Abstufungen gehen mit den UV-Anteilen der unterschiedlichen Lichtquellen – TL84 (geringer UV-Anteil), D65 (mittlerer UV-Anteil) und UV (hoher UV-Anteil) einher (Abbildung 15). Wie bereits im Kapitel 4.6.1 beschrieben, korrelieren, die  $L^*a^*b^*$  Werte mit der Intensität des jeweiligen Einflussfaktors. So kann mittels der Farbabstandsformel  $\Delta E_{00}$  (Kapitel 4.1) der Abstand zwischen den Farbfeldern berechnet werden (Tabelle 22).

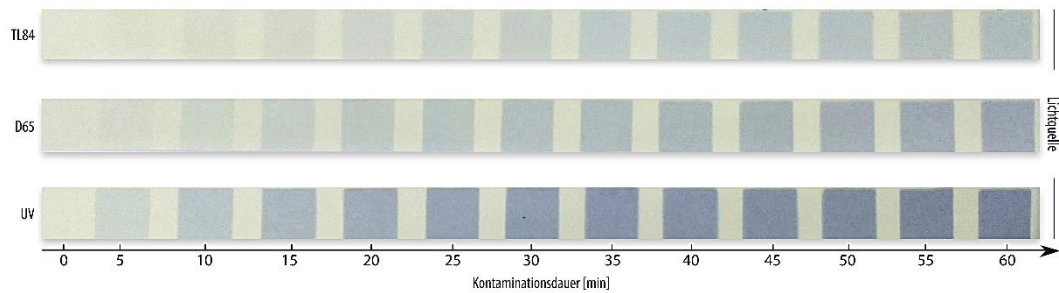


Abbildung 57 Farbveränderung des photochromen Sensors unter verschiedenen Lichtquellen

Min		0	5	10	15	20	25	30	35	40	45	50	55	60
TL84	L*	91	89	88	86	85	86	83	82	79	79	78	77	75
	a*	-4	-4	-5	-5	-5	-6	-6	-6	-7	-7	-6	-4	-7
	b*	9	9	8	6	2	4	2	-1	-2	-2	-2	-1	-3
	$\Delta E_{00}$		1,25	1,56	1,29	4,79	2,00	2,60	2,63	2,43	0	1,27	2,45	3,88
D65	L*	88	87	84	82	80	78	77	75	73	73	71	69	68
	a*	-4	-6	-7	-8	-8	-8	-7	-8	-7	-7	-5	-6	-5
	b*	10	6	5	2	0	-1	-3	-3	-4	-4	-5	-7	-8
	$\Delta E_{00}$		3,90	2,40	3,11	2,22	1,62	2,24	1,74	2,04	0	2,91	2,35	1,66
UV	L*	87	85	79	74	70	67	64	64	60	59	58	55	54
	a*	-4	-8	-7	-6	-7	-5	-5	-5	-5	-4	-4	-5	-5
	b*	8	3	-3	-6	-10	-12	-14	-12	-14	-14	-13	-14	-15
	$\Delta E_{00}$		6,37	4,18	4,62	4,33	3,65	2,73	1,2	3,63	1,33	1,06	3,06	1,20

Tabelle 22 Farbveränderung des photochromen Sensors unter verschiedenen Lichtquellen

Der Einflussfaktor von Lichtart auf die Farbintensität ist in Abbildung 58 zu sehen. Hierzu werden die L\* Helligkeitswerte der mit den drei Lichtquellen zu verschiedenen Zeitpunkten (alle 5 Minuten, in einem Maximalbereich von 60 Minuten) belichteten Flächen gegenübergestellt. Dabei lässt sich erkennen, dass alle Lichtarten einen Helligkeitsverlust hervorrufen, UV-Licht jedoch deutlich stärker als die restlichen Lichtquellen. Mittels der Farbabstandsformel  $\Delta E_{00}$  und den Farbwerten in Tabelle 22 ist ein Zusammenhang zwischen den L\*a\*b\* Farbwerten des TL84 Lichts bei 60 min, mit dem des D65 Lichts bei 35 min ( $\Delta E_{00}$  0,97) und dem des UV-Lichts bei 10 min ( $\Delta E_{00}$  2,85) zu erkennen. Zudem ist zwischen dem D65 Licht bei 60 min und dem des UV-Lichts bei 20 min mit  $\Delta E_{00}$  2,84 ein Farbwertzusammenhang festzustellen. Der kurzwellige UV-Anteil in der jeweiligen Lichtart hat somit einen Einfluss auf die Farbintensität, die auf eine energiereichere UV-Strahlung zurückzuführen ist (Herbst und Hunger, 1995, S. 577).

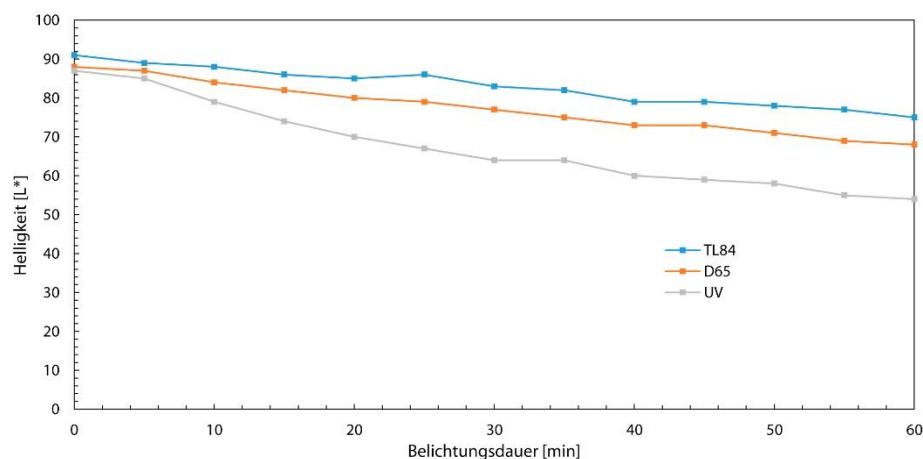


Abbildung 58 Lichtquellenbedingte L\*-Wertveränderung (Tabelle 22) des photochromen Sensors



#### 4.7.2.5 Spektralphotometrische Untersuchung

Mit einem Spektral-Densitometer (*Kapitel 3.4*) wurden die Remissionskurven der mit den drei Lichtquellen (TL84, D65 und UV) belichteten photochromen Flächen (Abbildung 57) messtechnisch ermittelt (Abbildung 59). Alle Proben zeigen im Spektralbereich von 400 nm bis 450 nm (blauer Spektralbereich) nahezu das gleiche, zeitlich unabhängige Remissionsverhalten. Im Bereich des längerwelligen sichtbaren Bereichs (500 nm – 700 nm) nimmt die Remission bei allen Lichtarten mit der Dauer der Exposition ab. Selbst die Höhe der Abnahme (Abstände) differiert je nach Lichtart. Die Abstände zwischen den Belichtungsstufen von TL84 sind nahezu kontinuierlich und weisen auf den niedrigen UV-Anteil der Lichtquelle hin, welches sich mit der gemessenen spektralen Verteilung und dem geringen UV-Anteil von TL84 (*Kapitel 3.5*) deckt. Im Vergleich dazu sind die Abstände von D65 und UV nicht gleichmäßig und werden mit der Dauer der Exposition kürzer. Je intensiver die Farbe, desto größer ist der Anteil des absorbierten Lichts. Bei der UV-Lichtart wird im Betrachtungszeitraum annähernd eine Sättigung der Remissionskurven erreicht. Der 60-Minuten Vergleich (Abb., unten rechts) weist auf den Einflussfaktor der Strahlungsintensität ( $I$ ) des energiereicheren UV-Anteils der jeweiligen Lichtquellen auf die Intensität der Blaufärbung hin. Es wird ein Minimum an Energie (Photonenenergie) benötigt, um die Elektronen auf höhere Energieniveaus anzuheben (Elektronenanregung) – dies erklärt die geringe Farbintensität durch TL84. Infolgedessen reagiert der photochrome Sensor auf UV-Licht, in dem die Farbe allmählich mit der Dauer und Intensität des UV-Lichts wechselt und intensiviert.

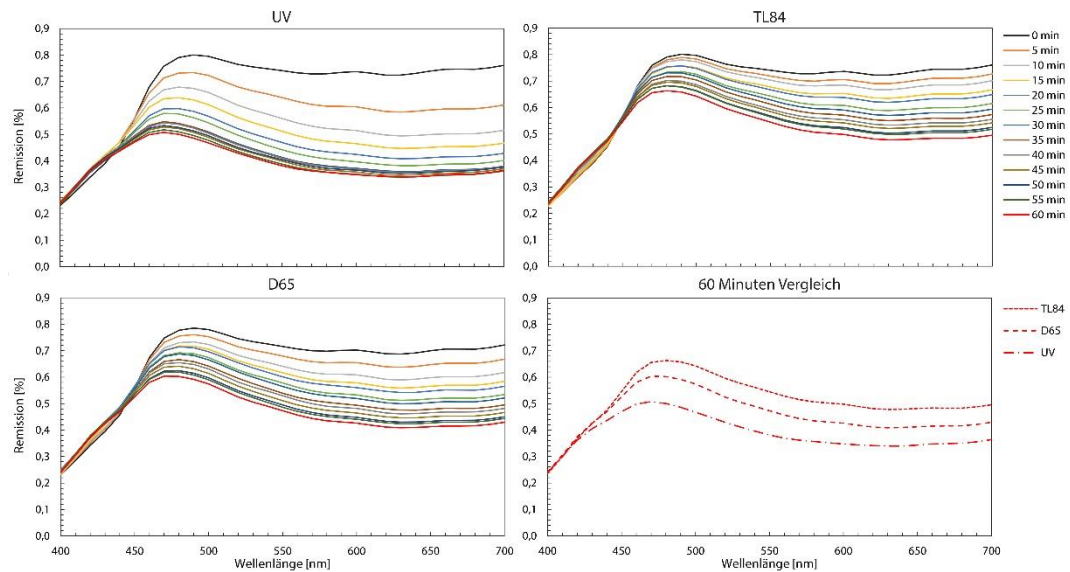


Abbildung 59 TL84, D65 und UV im Vergleich und ihre Exposition auf den photochromen Sensor

## 4.7.3 Halochrom

### 4.7.3.1 Untersuchung der rheologischen Eigenschaften

Die rheologischen Eigenschaften des halochromen Sensors werden auf Basis von Diferuloylmethan in drei unterschiedlichen Konzentrationen (Abbildung 60) untersucht. Entsprechend weisen alle Konzentrationen, die auf Ethanol basieren, ein relativ newtonsches Fließverhalten auf, mit einer im Vergleich zu E24 (4,24 mPa·s) niedrigen Viskosität.

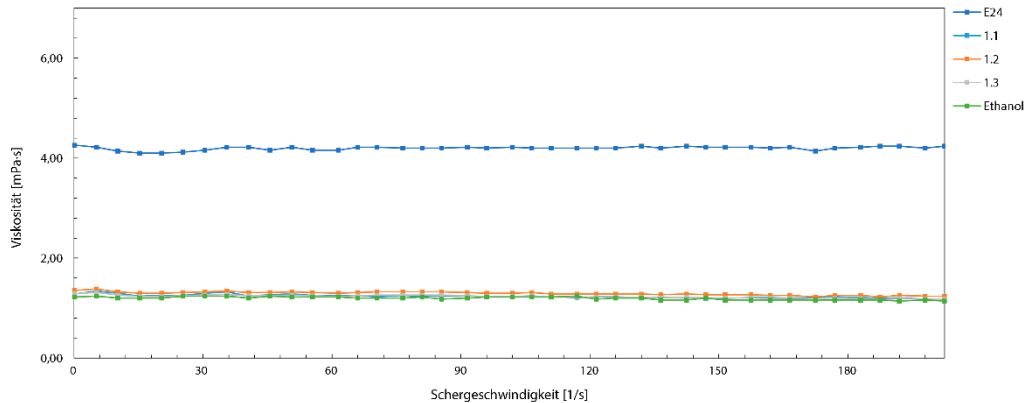


Abbildung 60 Viskositätsanalyse aller halochromen Konzentrationen im Vergleich

Die Grundmatrix, in der die halochromen sensorischen Farben eingebettet sind, bildet das Lösungsmittel Ethanol (1,32 mPa·s). Ethanol weist von allen Materialien den niedrigsten Viskositätswert auf und bildet dabei eine Untergrenze. Die Viskosität der unterschiedlichen Konzentrationen der halochromen Sensoren liegen bei 1,38 mPa·s (Mittelwert) und damit kaum über der Viskosität von Ethanol. Ziel war es, die rheologischen Eigenschaften der halochromen Sensoren zu bestimmen und ihre Parameter für das verwendete Inkjetdruckverfahren einzustellen (Abbildung 13).

### 4.7.3.2 Untersuchung der Oberflächenspannung

Die Oberflächenspannung der verschiedenen Konzentrationen des halochromen Sensors wurden untersucht (Abbildung 61). Ziel war es auch hier, die Oberflächenspannung hinsichtlich der Eigenschaften und ihrer Eignung für das Inkjetdruckverfahren zu verifizieren. Während der Messung betrug die Temperatur konstant  $\vartheta = 20,5 \text{ °C}$ . Wie in der Abbildung dargestellt, liegen die Oberflächenspannung aller Konzentrationen bei 21,68 mN/m und sind damit fast identisch mit der des Lösungsmittels Ethanol (21,87 mN/m).

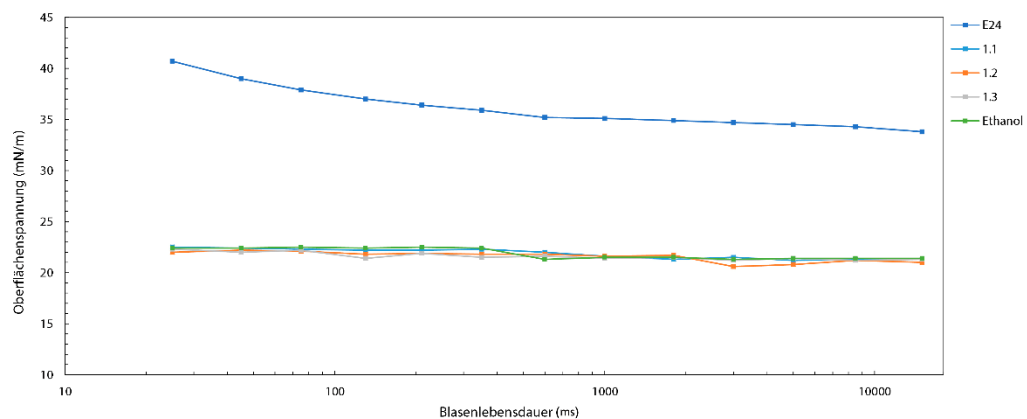


Abbildung 61 Oberflächenspannung verschiedener halochromer Konzentrationen

Die Ergebnisse der Untersuchung der Viskosität und Oberflächenspannung werden nachfolgend zusammengefasst (Abbildung 62). Entsprechend wird auch hier ein Zielparameterfenster definiert, das im empfohlenen Bereich der Oberflächenspannung (ebd., S. 35; Hutchings & Martin, 2013, S. 24) von 20 mN/m bis 50 mN/m und der Viskosität von 1 mPa·s und 25 mPa·s (Magdassi, 2010, S. 35) liegt (Abb., grauer Hintergrund). Die E24 Basistinte dient zusätzlich als Referenz (37,08 mN/m). Demzufolge liegt die Zusammensetzung des halochromen Sensors mit einer Viskosität bei etwa 1,38 mPa·s und einer Oberflächenspannung mit 21,68 mN/m (Mittelwert) innerhalb des druckbaren Bereichs des Zielparameterfensters.

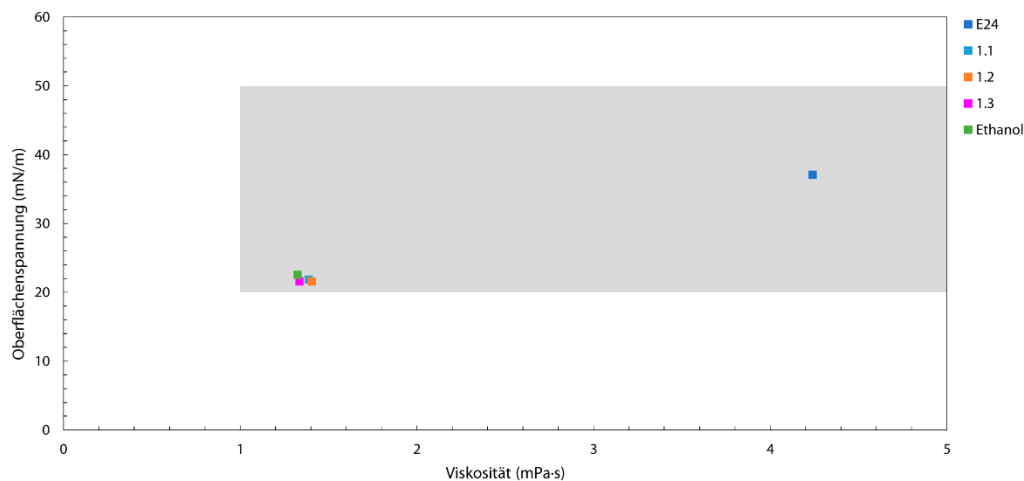


Abbildung 62 Halochromer Sensor und dessen Zielparameterfenster

#### 4.7.3.3 Untersuchung der Partikelgrößenverteilung

Diferuloylmethan besitzt laut RÖMPP (2020) bei 3 mg/ml in Ethanol eine gute Löslichkeit (*Kapitel 4.6.3.5*). Nachfolgend wurde eine mikroskopische Untersuchung der ungelösten Partikel in der Lösung herangezogen (*Kapitel 4.6.3*). Hierzu wurde die niedrigste Konzentration 1.1. (Tabelle 21) gewählt, um diese unter dem von der Quelle angegebenen maximalen Löslichkeitsbereich zu verorten. Die Aufnahmen der mikroskopischen Untersuchung stellen bereits dar (Abbildung 63), dass die niedrigste Konzentration der halochromen sensorischen Farbe für den Einsatz als Tinte, wie erwartet, eine schlechte Löslichkeit aufweist. Wie im *Kapitel 4.6.3.5* erwähnt, wurde durch die Konzentrationserhöhungen vielmehr eine höhere Farbintensität erzielt, die sich signifikant von den anderen Konzentrationen abhebt und ein höheres Kontrastverhältnis aufweist. Die homogene Verteilung der gelösten Anteile verwirkt ihre Inkjetfähigkeit aufgrund von vermehrten (>10 µm) Kern-Satellit-Partikeln<sup>5</sup> (Abb., rechts). Während einer stetigen Verdunstungstrocknung der Lösung aufgrund des Ethanols, finden sich vermehrt agglomerierte Einschluss-Partikel (Abb., links), die sich nicht gelöst haben. Diese einzelnen Partikel können jedoch, um eine Inkjetdruckfähigkeit zu gewährleisten, durch einen nachfolgenden Filtrationsprozess beseitigt werden (*Kapitel 3.2.4*). Hierbei erfolgt kein Verlust der Reaktionsfähigkeit des halochromen Sensors (*Kapitel 4.7.3.4*). Außerdem wird die jeweils erzielte Farbintensität des halochromen Sensors, die durch die unterschiedlichen Konzentrationen hervorgerufen wird, nicht gemindert.

<sup>5</sup> Kleine Partikel, die sich an einem Feststoffpartikelkern sammeln.

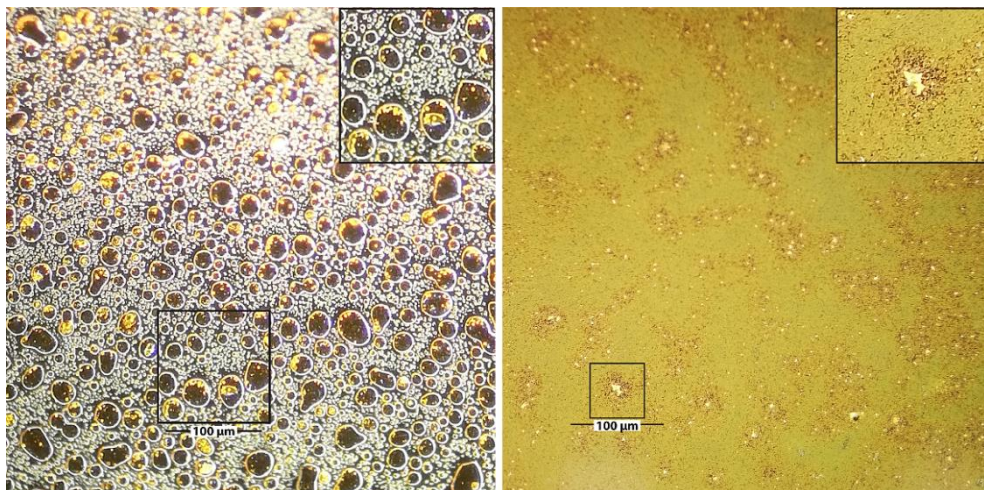


Abbildung 63 Ungelöste Diferuloylmethan Partikel im flüssigen Zustand / nach stetiger Verdunstungstrocknung innerhalb der Flüssigkeit

#### 4.7.3.4 Spektralphotometrische Untersuchung

Im Folgenden werden die Remissionskurven des halochromen Sensors in seinen unterschiedlichen pH-bedingten Farbveränderungen dargestellt (Abbildung 64). So sind die Remissionskurven klar voneinander abgegrenzt. Dies ist ebenfalls in den Farbfeldern (Abb., rechts) zu erkennen. Der Farbveränderungsprozess lässt sich ebenfalls mit Hilfe der  $L^*a^*b^*$ -Werte beschreiben. Hierzu können die einzelnen Farbinformationen über den Bereich der pH-Werte in einer Referenztafel (Lookup-Tabelle) abgelegt, abgerufen und verglichen werden. Das Umgebungslicht kann die Messwerte beeinflussen. Entsprechend können jedoch mindestens zwei pH-Werte-Bereiche (pH 2,5; Neutral; pH 10) des halochromen Sensors auf Diferuloylmethan Basis trotz umgebungslichtbedingter farblicher Schwankungen in folglich zwei Bereichen interpretiert werden, da die farbliche Differenz zwischen den pH-Wertbereichen groß genug ist. So kann mittels Farbabstandsformel  $\Delta E_{00}$  (Kapitel 3.4) der Abstand ( $pH_{2,5} \rightarrow pH_{Neutral} = \Delta E_{27,99}$  und  $pH_{10} \rightarrow pH_{Neutral} = \Delta E_{62,67}$ ) zwischen den beiden pH-Wertbereichen zur neutralen Mitte berechnet werden.

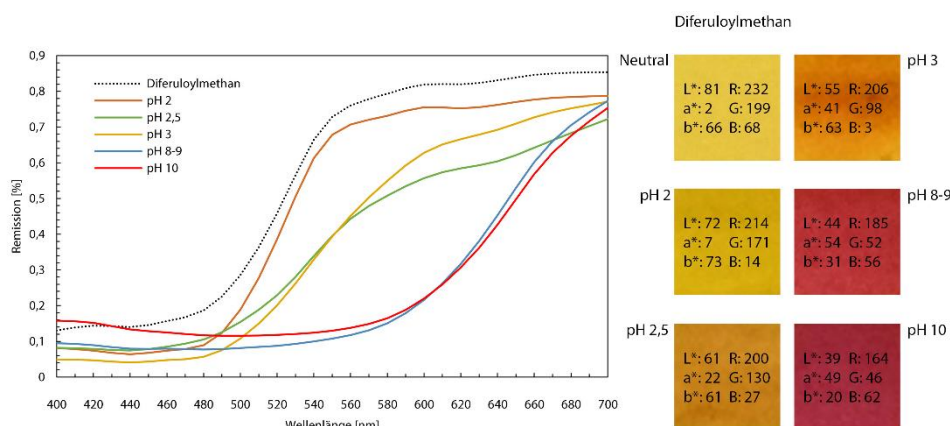


Abbildung 64 pH-Wert Veränderungen halochromer Sensoren

## 4.8 Untersuchung eines geeigneten Haftpapiers

Die zuvor entwickelten sensorischen Tinten, werden auf ihre Wechselwirkung, mit den in Tabelle 10 aufgeführten Substraten, untersucht. Hierbei kommen die in *Kapitel 3.1.2* beschriebenen Methoden zum Tragen. Das Ziel dieses Kapitels ist es, ein geeignetes Substrat, genauer ein Haftpapier, für die sensorischen Tinten zu identifizieren, da die sensorischen Tinten als integraler Bestandteil eines intelligenten Codes auf einem Haftpapier verdruckt werden sollen.

### 4.8.1 Hydrochrome

Der hydrochrome Sensor wurde auf unterschiedlichen Substraten (A – G) gedruckt. Die Abbildung 65 stellt die Farbfelder vor und nach einer Reaktion dar. Die jeweiligen Farbwerte und ihre Farbabstände können dabei aus Tabelle 23 entnommen werden. So sind bereits erhebliche Farbunterschiede zwischen den einzelnen Papieren, bei den Farbfeldern ohne Sensorreaktion (*vorher*) zu erkennen, wodurch zwei grobe Einteilungen möglich sind. Die Substrate A – B weisen den typisch blauen Farbton von Patentblau V auf (entspricht etwa dem Patentblau V Farbwert in Pulverform) und haben einen geringen Farb- abstand von höchstens  $\Delta E_{00} = 1,77$ . So ist die entsprechende hydrochrome Funktionalität in der unteren Reihe mit einem max. Farbabstand von höchstens  $\Delta E_{00} = 3,64$  zu erkennen (nachher). Der Übergang zwischen den Substraten B und C weist einen großen Farbabstand von  $\Delta E_{00} = 20,59$  auf. Dabei erscheint der Sensor auf den nachfolgenden Substraten C – G grünlich, die ebenfalls typisch für Patentblau V sind. Entsprechend weist Patentblau V halochrome Eigenschaften auf und ändert durch den pH-Wert der Substratoberfläche bedingt seine Farbe in saurem Milieu (korallblau). Nach der Farbreaktion (nachher) ist zu sehen, dass die hydrochrome Funktionalität von Patentblau V größtenteils eliminiert wurde und somit die Substrate C – G (gestrichelt) für eine weitere Verwendung ungeeignet sind.

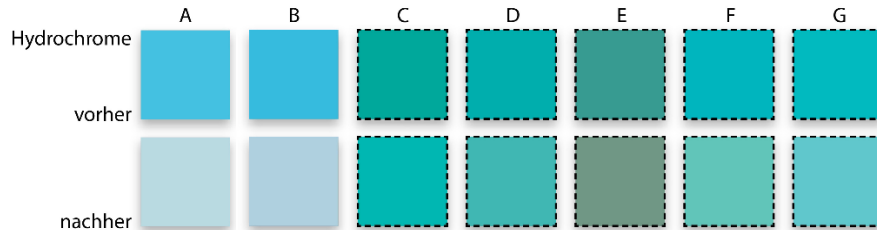


Abbildung 65 Einflusspotential der Substrate auf das Farbwechselverhalten des hydrochromen Sensors

Substrat		A	B	C	D	E	F	G
vorher	L*	72	70	59	63	58	68	71
	a*	-28	-28	-62	-56	-32	-51	-47
	b*	-27	-29	-9	-14	-4	-18	-14
	$\Delta E_{00}^{a)}$		1,77	20,59	4,95	10,01	12,53	3,14
nachher	L*	85	82	68	68	59	78	77
	a*	-11	-10	-45	-35	-17	-43	-36
	b*	-7	-11	-9	-9	4	-3	-13
	$\Delta E_{00}^{a)}$		3,64	20,04	3,71	14,33	19,44	7,32
$\Delta E_{00}^{b)}$		15,46	14,49	9,13	8,06	9,55	11,25	5,92

Tabelle 23  $L^*a^*b^*$ -Werte der Einflusspotentiale auf den hydrochromen Sensor in Abbildung 65

Legende:

a) Abstand zu den Nachbarn; b) Farbabstand vorher / nachher

## 4.8.2 Photochrome

Wie bereits im (Kapitel 4.3) erwähnt, hatte sich nach dem unmittelbaren Kontakt zwischen dem photochromen Sensor und dem Substrat F eine sofortige und vollständige Blaufärbung und bei G eine sofortige Braunfärbung (Reaktion) ergeben (Abbildung 22). Diese ist auch in Abbildung 66 zu erkennen, in der sich, wie schon beim hydrochromen Sensor, Gruppierungen auszeichnen lassen, deren Farbwerte aus Tabelle 24 entnommen werden können. Die auf den Substraten A und B gedruckten Sensoren weisen einen Farbabstand von höchstens  $\Delta E_{00} = 3,41$  auf. Wobei der Farbabstand zwischen B und C mit  $\Delta E_{00} = 10,88$  relativ hoch ist. Das Substrat C weist zu D einen Farbabstand von  $\Delta E_{00} = 10,94$  auf, was einem nahezu ähnlichen Übergang entspricht. Der Abstand zwischen D und E beträgt dabei  $\Delta E_{00} = 10,55$  und unterscheidet sich mit  $\Delta E_{00} = 32,85$  sehr zwischen F und G. Somit eignen sich die gestrichelten Substrate C und E – G aufgrund ihres geringen Kontrastumfangs und ihrer Farbverfälschung nicht. Wohingegen sich die Substrate A, B und D besser eignen, jedoch A und D aufgrund des höheren Kontrastumfangs und der Farbzuverlässigkeit am besten eignen.

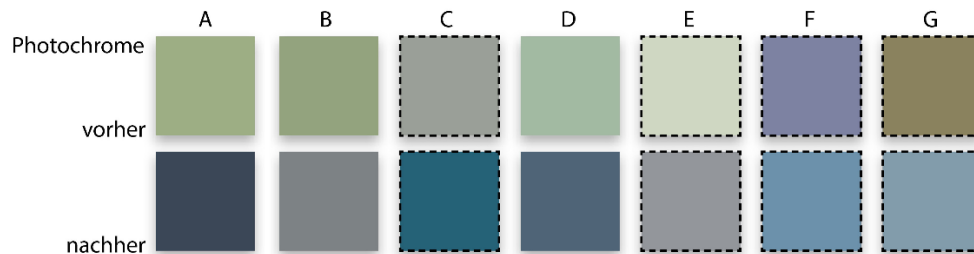


Abbildung 66 Einflusspotential der Substrate auf das Farbwechselverhalten des photochromen Sensors

Substrat		A	B	C	D	E	F	G
vorher	L*	69	65	65	73	85	55	54
	a*	-12	-12	-4	-13	-6	3	-1
	b*	19	17	3	9	9	-18	-20
	$\Delta E_{00}^{a)}$		3,41	10,88	10,94	10,55	32,85	5,05
nachher	L*	29	54	38	41	62	58	63
	a*	-2	-2	-15	-5	-1	-9	-7
	b*	-12	-2	-19	-14	-3	-18	-12
	$\Delta E_{00}^{a)}$		23,71	21,25	8,37	22,61	12,61	5,67
$\Delta E_{00}^{b)}$		47,41	18,79	32,03	35,52	20,73	12,57	11,27

Tabelle 24 L\*a\*b\*-Werte der Einflusspotentiale auf den photochromen Sensor in Abbildung 66

Legende:

a) Abstand zu den Nachbarn; b) Farbabstand vorher / nachher

## 4.8.3 Halochrome

Die geringsten Einflüsse des Substrats auf den halochromen Sensor konnten mit den Substraten A, B, D und G erreicht werden. Diese liegen mit einem geringen Farbabstand nahezu beieinander. So kann auch im Vorher-Nachher-Vergleich der typische pH bedingte rote Farbwert (Abbildung 67) erkannt werden. Tabelle 25 stellt weitere Farbabstände zur Verfügung, so können auch die Abstände vor und nach einer pH-bedingten Reaktion entnommen werden. Ungeeignet sind dagegen die Farbfelder C, E und F.

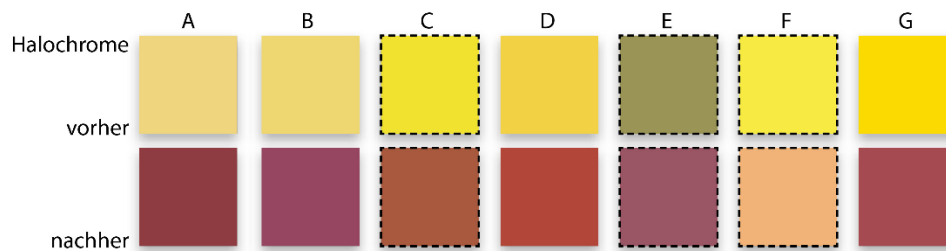


Abbildung 67 Einflusspotential der Substrate auf das Farbwechselverhalten des halochromen Sensors

Das Substrat C wechselt nach einer pH-bedingten Reaktion in einen untypischen Braunton. Das Substrat E wiederum veränderte sich erheblich und weist zwischen den Substraten D und E (vorher) einen Farbabstand von  $\Delta E_{00} = 21,58$  auf. Das Substrat F ist ebenfalls ungeeignet, da es in einen untypischen Orangeton wechselt und sich ebenfalls aufgrund der Farbverfälschung nicht als Substrat für einen pH-Nachweis eignet.

Substrat		A	B	C	D	E	F	G
vorher	L*	86	86	89	85	61	92	88
	a*	1	-1	-8	1	-5	-9	-1
	b*	46	52	79	69	33	77	86
	$\Delta E_{00}^{a)}$		2,33	7,87	6,11	21,58	25,63	5,68
nachher	L*	37	41	47	45	45	78	44
	a*	35	36	31	43	30	18	38
	b*	14	1	30	31	4	39	15
	$\Delta E_{00}^{a)}$		8,71	19,50	5,85	14,65	37,99	36,68
$\Delta E_{00}^{b)}$		53,67	48,66	47,97	48,19	38,77	24,34	56,46

Tabelle 25  $L^*a^*b^*$ -Werte der Einflusspotentiale auf den halochromen Sensor in Abbildung 67

Legende:

a) Abstand zu den Nachbarn; b) Farbabstand vorher / nachher

#### 4.8.4 Zusammenfassung

Entsprechend lässt sich zusammenfassen, dass bereits bei dem hydrochromen Sensor die Substrate C – G aufgrund ihrer eliminierten Funktionalität einschließlich der Farbverfälschung ungeeignet sind. Die Substrate A, B und D wurden für den photochromen Sensor identifiziert, wobei das Substrat D aufgrund der im vorherigen Kapitel identifizierten Eliminierung der hydrochromen Funktionalität, nicht verwendet werden kann. Beim halochromen Sensor wurden ebenfalls die Substrate A, B und zusätzlich D und G identifiziert. Da jedoch D und G ebenfalls in Kombination mit den hydrochromen und photochromen Sensoren nicht verwendet werden können, werden die Haftpapiere A und B als geeignete Substrate qualifiziert.

## Kapitel 5

# Aufbau und Funktion eines intelligenten Codes

Nachfolgend werden die notwendigen Grundlagen und Kriterien für die Entwicklung und Ausgabe eines intelligenten Codes untersucht und vorgestellt. Im *Kapitel 2 Stand der Forschung*, insbesondere in *Kapitel 2.2*, wurden *intelligente Code-Technologien* untersucht und der Bedarf für einen zu entwickelnden intelligenten Code identifiziert.

Im Folgenden wird darauf abgezielt, einen intelligenten Code mit vordefinierten Bestandteilen für die zuvor herausgearbeiteten sensorischen Tinten (*Kapitel 4.7*) zu entwickeln, der auf ein ausgewähltes Haftpapier verdruckt werden kann. Aufgrund der sensorischen Farben, kann der intelligente Code ein dynamisches Verhalten zeigen, in dem sich einzelne Informationen innerhalb des Codes z. B. durch Umwelteinflüsse ändern. Diese Eigenschaft steht im Gegensatz zu konventionellen statischen Codes, deren Inhalt unveränderlich sind. Die intelligenten Codes weisen neben einem statischen Bereich, z. B. zur Identifizierung eines Produktes (anhand einer ID) mehrere sensorische Farben im dynamischen Bereich auf, die Einflussfaktoren aus der unmittelbaren Umwelt detektieren. Dadurch können neben statischen Produktinformationen gleichermaßen dynamische Informationen über mehrere Einflussfaktoren detektiert werden. Die Erfassung des intelligenten Codes kann dabei On-Demand, über eine serverseitige Leseapplikation mittels eines Endgeräts (Smart Device) erfasst, transferiert und über die Leseapplikation analysiert und als Ergebnisbericht dem Benutzer visualisiert und zur Verfügung gestellt werden. Wird bspw. ein Paket mit einem intelligenten Code versehen, dann kann die Verpackungshistorie zu jederzeit während der Transport- und Lagerstrecke nachvollzogen werden. Es können zusätzlich mögliche Aussagen über den Zeitraum eines kritischen Umwelteinflusses getroffen werden. So kann der intelligente Code darüber hinaus als Instrument zur Untersuchung und Präsentation weiterer Funktionen sensorischer Farben, unter anderem von Forschern und Entwicklern, eingesetzt werden.

## 5.1 Grundlagen von Code-Technologien

In der Informationstechnik wird ein Code als ein „System von Regeln und Übereinkünften, das die Zuordnung von Zeichen, auch Zeichenfolgen, [...] erlaubt“, definiert (Duden, 2020). Die kodierten Daten werden dabei größtenteils binär in schwarzen und weißen Modulen dargestellt. Ein Modul ist dabei die kleinste quadratische Einheit eines Codes. Um den Aufbau und den Umfang eines intelligenten Codes nachzuvollziehen, vermittelt dieses Kapitel die notwendigen Grundlagen.

Zu den maschinenlesbaren Codes gehören bspw. DataMatrix Codes, QR-Codes (Quick Response) und weitere im folgenden besprochene Formen. Mehrdimensionale Codes zeichnen sich durch eine hohe optische Speicherdichte sowie eine schnelle und effiziente Lesbarkeit aus. Zudem ist bei einem Großteil der Codetypen eine abgestufte Fehlerkorrektur integriert, die Fehler in Folge von Verunreinigungen, Kratzern, Rissen und auch fehlenden Segmenten durch mathematische Korrekturalgorithmen kompensiert.



Um einen Überblick zu vermitteln, werden die unterschiedlichen Codes zunächst in Dimensionen eingeordnet:

- a) Eindimensionale Codes (1D)  
Charakteristisches Merkmal dieser Codes ist die Kodierung der Daten entlang einer geraden Achse (X-Achse). Zu den Vertretern der 1D-Codes gehören Strich- oder Barcodes.
  
- b) Zweidimensionale Codes (2D)  
Codes in Form von Composite-, Punkt- oder Matrix-Codes verorten ihre Datenkodierung in einem Feld von zwei senkrecht zueinanderstehenden Achsen (X- und Y-Achse), wodurch eine zum 1D-Code vergleichbar höhere Punktdichte (Datendichte) erzielt wird. Ein bekannter Vertreter ist hier der QR-Code.
  
- c) Dreidimensionale Codes (3D)  
Die Datendichte wird bei dieser Code-Form durch das Hinzuführen einer weiteren Achse, der Z-Achse, erhöht. Erzielt wird die Erhöhung bspw. durch das Hinzufügen einer Tiefeninformation (Höhenvariation), einer Farbtiefe oder mehrerer Farbinformationen. In einem solchen Fall können mit steganographischen Verfahren zusätzliche verdeckte Informationen durch den gezielten Einsatz eines Pseudoruschmusters (verschlüsselte Informationen im Rauschen des Bildes) in einem Bild eingebettet werden (Dittmann, 2000, S. 14 f.). Im unkomprimierten digitalen und indizierten 8-Bit RGB-Farbmodus (*Kapitel 5.1.3*) können 256 ( $2^8$ ) Farben dargestellt werden<sup>6</sup>, womit im Code durch Abstufungen eine höhere Informationsdichte zu erreichen ist. Beispielhaft sind in dieser Code-Kategorie der HCCB-Code (Microsoft Tag), Ultracode, JAB-Code und das Photo-TAN-Verfahren (Haupt, 2020) zu nennen.
  
- d) Vierdimensionale Codes (4D)  
Durch das Hinzuführen einer weiteren Dimension, der Zeit, kann durch den zeitlich variierten Bildwechsel die Informationsdichte in einem Code vergrößert werden (Labs, 2007). Der in *Kapitel 5.3* entwickelte intelligente Code mit seinen integrierten sensorischen Farben kann verschiedene Einflussfaktoren detektieren und gehört in eben diese Kategorie.

---

<sup>6</sup> So können im 8 Bit RGB-Farbraum pro Kanal (R, G, B),  $(2^8)^3 \approx 16,8$  Mio. Farben dargestellt werden.

### 5.1.1 DataMatrix Code

Der DataMatrix Code (ISO/IEC 16022, S. 3) zeichnet sich durch seine hohe Informationsdichte im kleinen Maßstab und einer Reed-Solomon-Fehlerkorrektur (*ECC200*) aus, bei der, ähnlich dem QR-Code, bis zu 25 % der Daten wiederhergestellt werden können. Der DataMatrix Code ist Bestandteil des statischen Bereichs (für bspw. Produktinformationen) in *Kapitel 5.3*. Die kleinste Speichergröße des quadratischen DataMatrix Codes beträgt, bei einer Modulgröße von 10 x 10, 1 Byte, wohingegen die größte Modulgröße von 144 x 144 eine Speichergröße von 1555 Byte aufweist (ebd., S. 16). Hierbei unterscheidet sich die Speicherkapazität je nach numerischem oder alphabetischem Inhalt. Der DataMatrix Code ist charakteristisch durch seine vertikale und horizontale Begrenzungslinie an der linken Ecke zu erkennen (Abbildung 68, links). Dabei ist die L-Form ein Modul breit und dient der Identifizierung, Entzerrung und Ausrichtung des Codes. Ein weiteres Merkmal ist das Synchronisationsmuster, das auch als Taktzelle bezeichnet wird (Mitte). Es dient der Bestimmung der Matrixgröße, bei der die Anzahl von Zahlen und Spalten identifiziert werden. Ein weiterer Bereich ist der Datenbereich (rechts), in dem sich binär codierte Daten befinden. Schließlich wird der Code von einer Ruhezone mit Weißraum umgeben, welche den Code von anderen Inhalten abgrenzt und einfacher identifizieren lässt.



Abbildung 68 DataMatrix Code

#### 5.5.1.1 Kodierungsverfahren

Die Daten des DataMatrix Codes werden in drei Schritten kodiert. Im ersten Schritt werden die eingegebenen Daten analysiert und einem Zeichensatz zugeordnet. Hierzu stehen verschiedene Zeichensätze (Zeichensatzkodierungen) zur Verfügung (ASCII, ISO/IEC 646, Base 256, etc.). Häufig kommt dabei das ASCII<sup>7</sup>-Kodierungsschema (ASCII set encodation scheme) zum Einsatz, bei dem ein Datensatz bspw. ein Text oder eine Ziffernfolgen in Codewörter (hexadezimal) umgewandelt werden. Dabei kann der zu kodierende Inhalt bspw. ein Benutzername, zusammengesetzt aus einer Ziffernfolge von „1, 2, 3, 4, 5, 6“, bestehen (ebd., S.114). Beim Kodieren der Ziffernfolge mithilfe des ASCII-Kodierungsschemas werden die Ziffern in einen Dezimalwert transformiert „49, 50, 51, 52, 53, 54“ (ebd., S. 90). Bei der ASCII-Kodierung wird dem Standard entsprechend der aus Ziffern bestehende Benutzername in 3 Bytes umgewandelt. Dies geschieht unter Verwendung der folgenden Formel für Ziffernpaare, bei der zwei-stellige Daten von 00 bis 99 mit den Codewörtern 130 (bis 229) kodiert werden (ebd., S. 7) (23):

$$\text{Codewort} = (\text{numerischer Wert des Zahlenpaares}) + 130 \quad (23)$$

<sup>7</sup> American Standard Code for Information Interchange

Aus der Formel heraus resultierend, lässt sich dies wie folgt berechnen ( 24 ):

$$\begin{aligned}
 1 \text{ Byte: } „12“ &= 12 + 130 = 142 \\
 2 \text{ Byte: } „34“ &= 34 + 130 = 164 \\
 3 \text{ Byte: } „56“ &= 56 + 130 = 186
 \end{aligned}
 \tag{24}$$

So besteht der Datensatz nach der Datenkodierung aus den Dezimalwerten „142, 164, 186“. Der kodierte Benutzername (Daten-Codewörter), inkl. 5 Fehlerkorrektur-Codewörter passt genau in eine Codegröße von 10 x 10 hinein. Um die für die Codeerstellung notwendige maximale Anzahl an Codewörtern zu erreichen, werden bei Bedarf Auffüllzeichen (Platzhalter) mit dem Codewortwert von 129 in ASCII-Kodierung eingesetzt (GS1, 2018, S. 12). Dabei wird stets versucht, die kleinste Matrixgröße mit dem Datensatz aufzufüllen. Bei Überschreitungen wird die nächstgrößere Matrixgröße verwendet. Der zweite Schritt ist die Fehlerkorrektur. Hierzu werden die Codewörter der Fehlerkorrektur mit einem Reed-Solomon-Algorithmus generiert (rot) und den zuvor kodierten Daten auf Basis von Dezimalwerten angehängen (ISO/IEC 16022, S. 114):

Benutzername:	1, 2	3, 4	5, 6						
Dezimal:	142	164	186	114	25	5	88	102	
Hexadezimal:	8E	A4 BA		72	19	05	58	66	( 25 )
	Daten -----			Prüfsumme-----					

Der dritte Schritt ist die binäre Modulplatzierung innerhalb der Matrix (Abbildung 69). Die endgültigen Codewörter (*Hexadezimalwerte*) des vorherigen Schrittes werden in eine binäre Matrix überführt ( 26 ):

Hexadezimal	Binär	
8E	10001110	
A4	10100100	
BA	10111010	
72	1110010	
19	11001	
05	101	
58	1011000	
66	1100110	( 26 )

Die Binärziffern werden in Module umgewandelt, wobei jedes Modul eine nominell quadratische Form aufweist und jeweils ein binäres Bit (0 für weiß, 1 für schwarz) darstellt. Dabei wird jede Binärziffer in der Vertikalen und Horizontalen durch 8 Module dargestellt. Abschließend werden unter anderem die Taktzelle und das Suchmuster in der typischen L-Form in die Matrix eingefügt (ebd., S. 60).

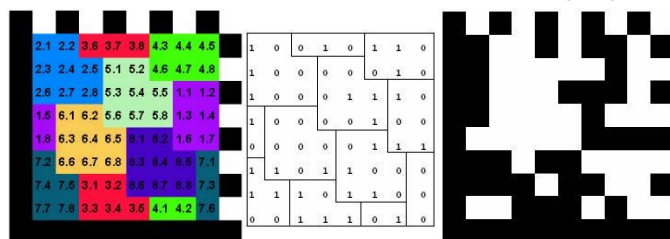


Abbildung 69 DataMatrix Code Anordnungsschema und Resultat (ISO/IEC 16022, 2006, S. 23)

## 5.1.2 Lesbarkeit

Bar- oder Strichcodes werden mittels optoelektronischen Barcodelesegeräten ausgelesen. Diese arbeiten mit Licht einer Leuchtdiode im Rot- oder Nahinfrarotbereich (Wellenlänge von 700 nm – 1000 nm) (Lenk, 2002, S. 101 ff.). Die Lichtreflexionen der einzelnen Elemente des Codes werden durch die Dekodiereinheit des Lesegeräts ausgewertet und als Information ausgegeben.

Eine weitere Möglichkeit, ein- oder mehrdimensionale Codes auszulesen, sind Leseapplikationen (Apps). Diese Apps können sowohl lokal als auch serverseitig über unterschiedliche Endgeräte (Smart Devices) wie bspw. Smartphones, Tablets, Smart Glasses und weiteren kommunikationsfähigen Endgeräten, die über eine integrierte Kamera verfügen, ausgelesen werden. Letztere, insbesondere Smartphones, werden für die vorliegenden Experimente eingesetzt. Barcode-Scanner Apps erkennen zumeist nur bestimmte 2D-Code-Typen, die sie auslesen können. Strukturelle Veränderungen des Codes (Änderung des Datenbildes), wie sie aus sensorischen Farben resultieren, vermögen bestehende Apps nicht zu identifizieren. Alle bekannten Code- und Korrekturalgorithmen werden damit hinfällig. Infolgedessen ist ein entsprechender Code für dynamische Informationen zu entwickeln, der mit unterschiedlichen Smart Device Kameras ausgelesen werden kann. Die notwendigen Grundlagen zum Verständnis von CCD und CMOS Sensoren (*Kapitel 5.1.4*), welche integrativer Bestandteil von Smart Device Kameras sind, werden im folgenden Kapitel näher behandelt.

## 5.1.3 Color Lookup Table

Indizierte Farbbilder sind mindestens 1 bis 8 Bit Bilder, die aus mindestens 2 und bis zu maximal 256 darstellbaren Farben (max. 8 bpp = bit pro Pixel) bestehen können. Die Farbinformationen werden dabei nicht in den Pixeldaten, sondern in einem separaten zweidimensionalen Array gespeichert, bei dem jedes Element im Array eine Farbe repräsentiert, die durch ihre Position innerhalb des Arrays indiziert wird (Lookup-Table). Ein Lookup-Table zeichnet sich dadurch aus, dass es die Umrechnung der Werte des Originalbildes in die eines Zielbildes ermöglicht. Bei der Konvertierung eines Originalbildes in einen indizierten Farbmodus wird seitens des Algorithmus ein entsprechendes Color Lookup-Table (CLUT) in Form einer Farbtabelle angelegt, in der die Bildfarben ohne Pixelzuordnung gespeichert und über Indexstrukturen (Indices) zugeordnet werden.

Dabei gibt die Farbtiefe die maximale Anzahl an verwendbaren Einträgen der Farbtabelle an. Durch die Beschränkung der Farbpalette (8 Bit) des Bildes über eine Indexfarbentabelle (Farbauswahlsystem) wird ein indiziertes Farbbild erzielt. Ist bei der Konvertierung des Originalbildes eine Farbe nicht im CLUT enthalten, wird dieser der Farbe zugeordnet, die ihr am nächsten steht. Die Umwandlung in den indizierten Farbraum über eine CLUT erfolgt dabei über den Framebuffer (Bildspeicher) der Grafikkarte und kann im Falle eines GIF und PNG-8 Dateiformats als Bestandteil der Bilddatei gespeichert werden (Bühler et al., 2018, S.16; Bühler et al., 2018; W3C, 2003, 3.1.23). Der Einsatzzweck der indizierten Farbbilder dient jedoch primär der begrenzten Datenspeichereinsparung mit einhergehender Beschleunigung der Bildschirmdarstellung. Im späteren Verlauf dieser Dissertation wird auf ein Lookup Tabelle zurückgegriffen, um Pixeltransformationen durchzuführen.

## 5.1.4 CCD und CMOS

Als optische Sensoren (optoelektronischer Sensor) sind die beiden Sensortypen CCD (Charge Coupled Device) oder CMOS (Complementary Metal Oxide Semiconductor) in Kamerasystemen (wie bspw. Spiegelreflexkameras und Smart Device Kameras) eingebaut (Abbildung 70). Diese wandeln Licht (Photonen) in elektrische Signale um, basieren auf dem Halbleitermaterial Silizium und sind empfindlich für den Spektralbereich von ca. 300 bis 1000 nm. CCD-Sensoren weisen einen vertikalen (Y-Achse) und horizontalen (X-Achse) Ladungstransport (wie bspw. in einem Schieberegister) auf, wobei die Ladungs- und Spannungsumwandlung der Pixel extern in der Kamera erfolgt. Von dort aus wird das analoge Signal in ein digitales Signal umgewandelt (A/D Wandlung). CMOS Sensoren hingegen können jedes Pixel des Sensors direkt adressieren (spaltenweises Decodieren und Auslesen), wobei direkt in den CMOS Sensoren die Ladungs- und Spannungsumwandlung erfolgt (Hering et al., 2017, S. 365 f.). Des Weiteren sind CMOS Sensoren schneller und verbrauchen weniger Energie als CCD-Sensoren, was gerade für kompakte Endgeräte wie Smartphones einen relevanten Faktor darstellt.

Durini (2020, S. 185) beschreibt, dass die „Kamera des Mobiltelefons [...] in der Regel mit einer sehr leistungsfähigen und hochauflösenden Kamera für allgemeine Foto- oder Videoanforderungen ausgestattet“ ist und die meisten Smartphones, Tablets, usw., CMOS-Bildsensoren verwenden. Diese sind im Vergleich zu CCD-Sensoren kostengünstiger und auch mit einem System-on-a-Chip, also einer Ein-Chip-Kamera, realisierbar (ebd., S. 189).

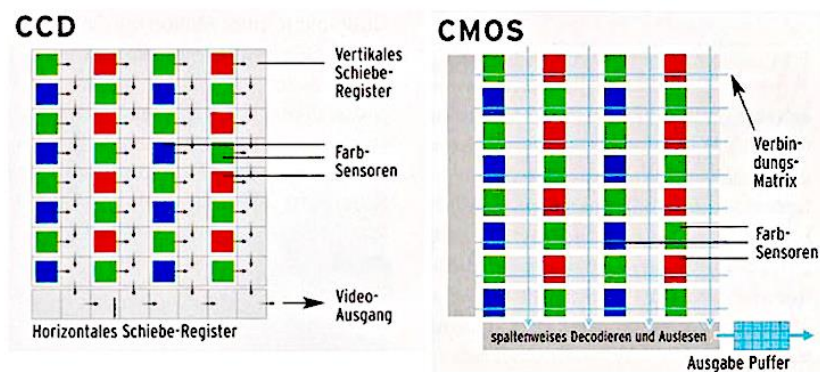


Abbildung 70 Funktionsweise von CCD- und CMOS-Sensoren (Hessler et al., 2003)

Monochrome CCD-Sensoren sind bspw. mit einer einzelnen Empfindlichkeit für den gesamten sichtbaren Spektralbereich ausgelegt (Abbildung 71). Bei der Anordnung der Sensorelemente und ihrer Signalverarbeitung nach Bayer (Bayer, 1976), werden den einzelnen Pixeln jeweils eine Primärfarbfilterschicht vorgeschaltet, die jeweils für den R-, G- oder B-Farbanteil der Bildinformationen ausgelegt ist und sich an den Empfindlichkeitseigenschaften des menschlichen Auges ausrichtet (Böhringer et al., 2014, S. 120). So finden sich verschiedene Kombinationen von Farbmustern in Primär- und Sekundärfarben sowie mit Weißanteilen als unterschiedliche Filter dem CCD-Sensor vorgeschaltet. Durch die Kombination der in der Abbildung vorgestellten Farbmustervariationen in Kombination mit den Sensor-Verarbeitungsalgorithmen versprechen sich Gerätehersteller qualitativ bessere Aufnahmen erreichen zu können. Die Sensorfläche wird dabei nacheinander zeilenweise ausgelesen (ebd., S. 121). Durch eigens entwickelte Algorithmen werden die Teilfarbinformationen durch eine nachfolgende Interpolation, eine Scharfzeichnung, (zur Korrektur der interpolationsbedingten Weichzeichnung) und abschließend durch die Durchführung eines Weißabgleichs zu einem dreifarbigem Pixelwert zusammengerechnet und

je nach Dateiformat bei Bedarf komprimiert (ebd., S. 120). Dementsprechend finden sich auch unterschiedliche Farbabweichungen zwischen den Smartphones verschiedener Hersteller. Auch wenn verschiedene Hersteller den gleichen Kamertyp verbauen, unterscheiden sich die Bilder aufgrund der herstellereigenen Varianten von Filtern und Algorithmen zur Farbauswertung und -darstellung. Das Sensorsignal der dynamischen Codes soll hersteller- bzw. bauartunabhängig mit Smartphones erfasst werden. Dazu sind eine Farbkorrektur und ein Colormanagementprozess erforderlich, mittels derer gerätebedingte Farbschwankungen und Farbinterpretationen weitestgehend kompensiert werden.

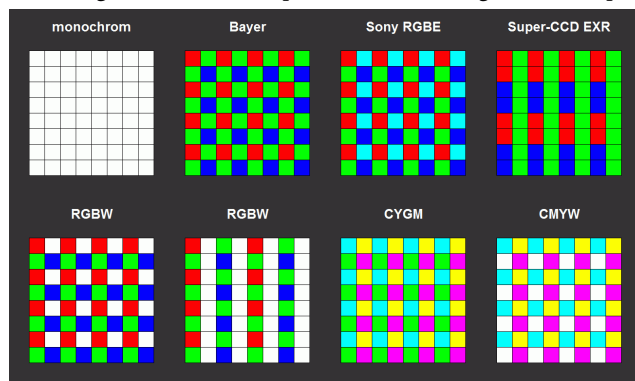


Abbildung 71 Für CCDs verwendete Farbmuster (Klemm, 2010)

### 5.1.5 Arten von Codes

Im folgenden Unterkapitel stehen die Suchmuster unterschiedlicher Codes im Fokus. Sie werden zur Identifizierung des Codes, des Datenbereichs und zur Ausrichtung des Codes verwendet. So besteht der Dateninhalt häufig aus kodierten Daten, Formatinformationen, Fehlerkorrekturbereichen und weiteren Codeelementen (Abbildung 72). Durch die Identifizierung und messtechnische Erfassung der geometrischen Eigenschaften des jeweiligen Codes, wie bspw. Größe, Form und Orientierung, kann jeder Sektor innerhalb des Codes bestimmt und ausgelesen werden. Bei der Entwicklung eines Lesealgorithmus sind die Zuverlässigkeit in der Auswerterroutine, die Robustheit in der Erkennbarkeit und Differenzierbarkeit eines Codes wesentliche Faktoren, die die Qualität der Lesbarkeit bedingen. Insbesondere für eine Echtzeiterkennung bspw. eines bewegten Ziels, zu dem ebenfalls 2D-Codes zählen, sind diese Qualitätsmerkmale wichtig.

Einer der bekanntesten Codes ist der QR-Code. Dieser wurde 1994 entwickelt (JP2938338) und 1997 als AIM-Standard (Automatic Identification Industry) genehmigt (DensoWave, 2020). Der QR-Code wurde eingesetzt, um die Komponenten eines Kraftfahrzeugs für eine schnelle Montage zu erfassen. Die Speicherkapazität eines Codes geht einher mit seiner Größe (Onoda & Miwa, 2013). Bei dem QR-Code kann eine beschädigte oder verunreinigte Fläche je nach Größe (als Level bezeichnet) bis zu ca. 30 % (Level H) durch einen Fehlerkorrekturalgorithmus rekonstruiert werden (ISO/IEC 18004, S. 5).

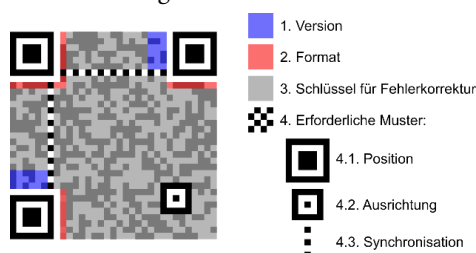


Abbildung 72 Aufbau eines beispielhaften QR-Codes nach (Wheeler & Kurpjuweit, 2009)

Charakteristisch für den QR-Code sind die Quadrate in der oberen linken, oberen rechten und unteren linken Ecke. Sie bilden die Suchmuster des Codes. Das Suchmuster dient der Positionsfindung, wobei alle drei Quadrate aus 7 x 7 äußeren schwarzen Modulen, 5 x 5 inneren weißen Modulen und 3 x 3 schwarzen Modulen besteht und außerdem im Inneren ein einfarbiges schwarzes Quadrat besitzt. Durch das Zählen dieser Module und ihrer festgelegten Anzahl können die Suchmuster und ihre Position identifiziert werden. Die Weißraumbereiche trennen die Suchmuster vom Datenbereich und bilden eine Ruhezone (Quiet zone). Dieser Bereich verbessert die Erkennbarkeit und Unterscheidbarkeit des Lesealgorithmus. Die Synchronisationsmuster (horizontal und vertikal) verbinden die Suchmuster miteinander und geben Auskunft über die Codeversion (Lenk, 2002, S. 503 ff.).

Anhand des QR-Codes wurde die Relevanz von Suchmustern, ihr Einfluss auf die Erfassbarkeit und somit auf die Lesbarkeit eines 2D-Codes aufgezeigt. Aus diesem Grund werden in Abbildung 73 unterschiedliche Codetypen dargestellt und auf ihre Suchmuster untersucht. Für die Entwicklung eines speziellen dynamischen Codes ist es notwendig, die Unterschiede und Beziehungen der bestehenden 2D-Codes zu verstehen. Insbesondere sind die Suchmuster für den ersten Schritt der Codeerkennung und somit für die Lokalisierung des Datenbereichs wichtig (Bilgin & Backhaus, 2019a).

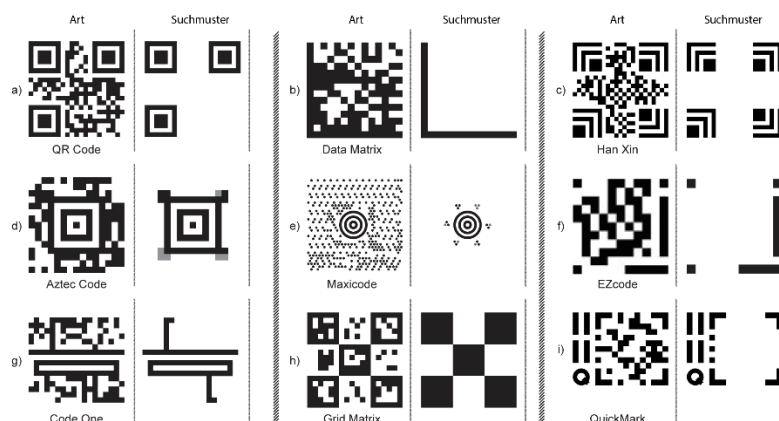


Abbildung 73 2D-Codes und ihre Suchmuster

Die Suchmuster des QR-Codes (a) wurden bereits beschrieben. Die L-Form des DataMatrix Codes (b) dient der Identifizierung, Entzerrung und Ausrichtung des Codes. Dabei werden die einzelnen Module abgefragt und der zugehörige Datenbereich identifiziert und dekodiert. Der Han-Xin-Code (c), basiert auf sparrenförmigen quadratischen Mustern. Das Suchmuster des Aztec-Codes (d) basiert auf vier sparrenförmigen quadratischen Mustern. Das Suchmuster des Aztec-Codes (d) basiert auf konzentrisch-quadratischen Rahmen mit einem einzelnen schwarzen Modul in der Mitte und vier Quadranten, die zur Orientierung und Ausrichtung des Codes dienen. Diese sind in der Abbildung schwarz und grau markiert. Das Suchmuster des Maxicodes (e) basiert auf konzentrischen Kreisen mit unterschiedlichen Durchmessern (auch *Bull's eye* genannt). So kann das zentrale Bull's eye-Muster über eine Circle Hough Transformation (CHT) detektiert werden, auf die in den Grundlagen des *Kapitel 6.1* eingegangen wird.

Des Weiteren weist der Maxicode sechs Muster auf, die symmetrisch um den äußeren Kreis angeordnet sind und zur Orientierung dienen. Die Suchmuster des EZcode (f) basierten auf drei schwarzen Modulquadraten, die sich jeweils in den äußersten Ecken befinden. Zudem dienen zwei unterschiedlich lange, senkrecht zueinanderstehende Linien an zwei Außenseiten des Quadrats ebenfalls zur Identifizierung

des Codes. Das Suchmuster des Code One (g) basiert auf einer Kombination von senkrecht zueinanderstehenden Linien zum Zentrum des Codes. Eine integrierte Variante eines Suchmusters findet sich im Grid Matrix-Code (h), der aufgrund des abwechselnden Schwarz-Weiß-Kontrastes (Schachbrettmuster), eine integrierte Suchmuster Funktionalität aufweist. Der QuickMark (i) ist durch die beiden doppelten vertikalen Linien, dem Buchstaben Q und dem sparrenförmigen Muster in den vier Ecken in Form eines Suchers eindeutig zu erkennen und begrenzt den Datenbereich des Codes. Zusammenfassend lässt sich hervorheben, dass viele Suchmuster auf Quadraten basieren. Der Maxicode wiederum verwendet als einziger Code Punkte für den Datenbereich und konzentrische Kreise als Suchmuster.

Abbildung 74 stellt für eine Vergleichbarkeit zwischen kreis- und quadratförmigen Codes den binären Erkennungsprozess der Mustererkennung (*Pattern Matching*) dar. Hierin sind Sequenzmuster der zwei Codetypen des QR-Codes und des Maxicodes dargestellt (rot). Durch einen entsprechenden Algorithmus zum Finden von Sequenzmustern, kann das jeweilige Suchmuster des Codes identifiziert werden (Bilgin & Backhaus, 2019b). Eine Kreisform (rot) stellt bei der Sequenzmustererkennung eine Schwierigkeit bei der Erfassbarkeit dar. Aufgrund der digitalisierten Identifikation, insbesondere bei kleinen Strukturen, erscheint das aus einzelnen Pixeln zusammengesetzte Bild eines Kreises nicht mehr ideal rund, sondern verzerrt. Diese Pixelstruktur erschwert eine in jedem Fall robuste Identifikation des Musters, die bei quadratischen Formen weniger stark fehleranfällig ist. Durch andere Verfahren, wie dem CHT, können kreisförmige Suchmuster besser erkannt werden.

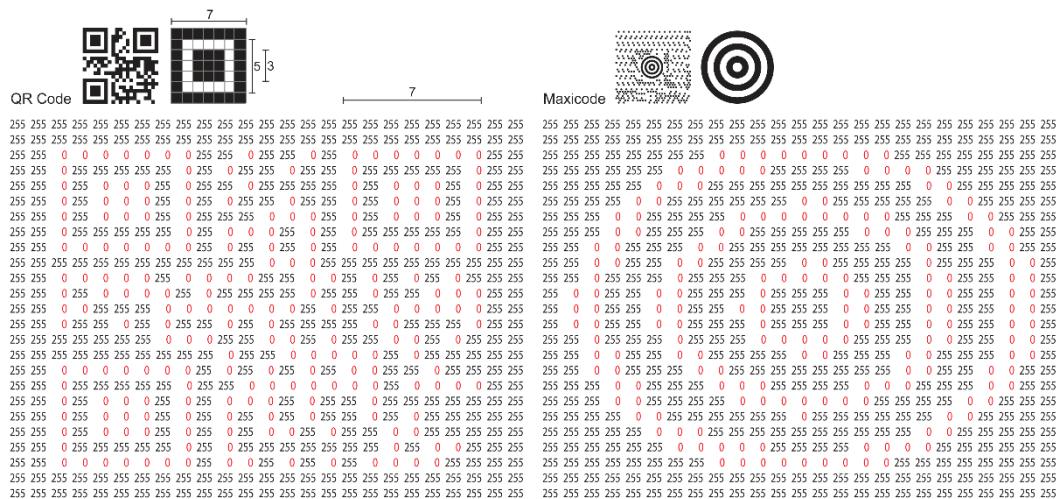


Abbildung 74 Codeerkennungsprozess

In Abbildung 75 werden drei geometrische Formen (Quadrat, Dreieck und Kreis) von Suchmestern in verschiedenen Größen (1:1, 1:2, 1:3) dargestellt. Wie zuvor beschrieben, erschwert die Pixelstruktur eine in jedem Fall robuste Identifikation des Suchmusters. Entsprechend hat die Größe und Auflösung der Suchmuster einen zusätzlichen Einfluss auf die Identifizierbarkeit der Suchmuster. Im direkten Vergleich der Suchmuster sind quadratische Formen robuster. So trägt die Größe der Module bspw. in einem QR-Code zur Zuverlässigkeit der Codefassung bei. Diese hängt auch von der Anzahl der Punkte (dpi) ab, welche der Druckkopf erzeugt, wobei die Anzahl der Punkte die Druckqualität erhöht. DensoWave (2020) hebt hervor, dass für eine zuverlässige Codeerkennung, jedes Modul aus vier oder mehr Punkten bestehen sollte. Weitere Einflussfaktoren auf das sichere Lesen eines Codes sind der Scanabstand und die Pixelauflösung.



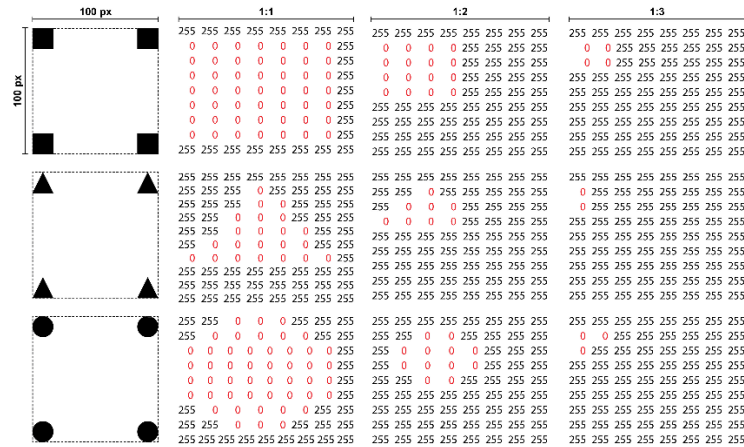


Abbildung 75 Erkennbarkeit von Suchmustern

Weitere Code-Typen können unter Lenk (2002, 2007) eingesehen werden.<sup>8</sup> Anhand der zuvor vorgestellten exemplarischen Codetypen konnte gezeigt werden, dass die Identifizierung eines Codes mit dem passenden Algorithmus in Kombination mit dem Suchmuster zusammenhängt.

Für die Entwicklung von Suchmustern ist folgendes wichtig:

- I. Die Marker sollten aus dem gesamten Datenmuster als markantes Muster hervorstechen, andernfalls könnte die Identifizierung des Codes erschwert werden.
- II. Die Position (Position an den Rändern), Rhythmus (wiederholende Frequenz), Reihenfolge (nacheinander) und Form eines Suchmusters (Kreise oder Quadrate) sind für die Entwicklung eines Codes relevant.
- III. Speziell die Erkennbarkeit der Suchmuster (geometrische Formen), wirken sich auf die Lesbarkeit aus (Abbildung 75).
- IV. Ungenauigkeiten (Unschärfe, Ausfransung, etc.) können die Lesbarkeit des Codes erschweren.
- V. Eine Mindestgröße des Codes selbst, insbesondere des Suchmusters, ist notwendig.

<sup>8</sup> So finden sich unter dem Literaturverweis (Lenk, 2007) weitere Codes wie der Color Ultra Code, Array Tag, Beetag, Blotcode, CoolData-Matrix, Dandelion Code, Data Glyphs, Dot Code, High Capacity Color Barcode (HCCB), JagTag und weitere.

## 5.2 Kriterien für einen intelligenten Code

Bisher ließen sich keine Hinweise, Regeln oder Kriterien für die Entwicklung der hier und im *Kapitel 2.2* behandelten intelligenten Code-Technologien in der Fachliteratur auffinden. Aus diesem Grund wurden für den Entwicklungsprozess eines intelligenten Codes Kriterien aufgestellt. Diese gewährleisten einerseits die Nachvollziehbarkeit des Entwicklungsprozesses und geben darüber hinaus einen Anhaltspunkt für weitere Forschungsvorhaben.

### *Kriterium 1*

#### *Bestandteile für intelligente Codes (Aufbau)*

Ein intelligenter Code muss je nach Einsatzzweck und Anwendungsbereich mindestens folgende Bestandteile aufweisen, die die Funktionalität und Lesbarkeit des Codes gewährleisten: Geometrische Formen der Suchmuster zur Mustererkennung mit einhergehender Positionsfindung und Ausrichtung stellen einen wichtigen Aspekt bei der Codeentwicklung dar. Anhand der Suchmuster kann der dazu ausgewählte und angepasste Algorithmus die Erkennung ausführen und der intelligente Code kann dadurch über die Elemente eines Programmiercodes bei Bedarf entzerrt werden. Alternativ können die Suchmuster jedoch auch über eine vorgegebene statische Platzierung der dynamischen Bereiche des intelligenten Codes als ein markantes Cluster, in dem sich die sensorischen Farben befinden, realisiert werden. Somit kann bspw. auf eigenständige Suchmuster verzichtet werden, da die Anordnung der dynamischen Bereiche bereits als Suchmuster dient. Je nach Bedarf kann ein Code zusätzlich statische Bereiche für Informationen wie bspw. Produktinformationen oder eine Domain etc. enthalten. Selbstverständlich sind dynamische Bereiche notwendig, die sensorisch auf einen oder mehrere Einflussfaktoren aus der Umwelt reagieren und sich irreversibel farblich verändern. Diese Bestandteile eines intelligenten Codes können durch weitere Features erweitert werden.

### *Kriterium 2*

#### *Anzahl integrierbarer sensorischer Farben und Art der gespeicherten Daten*

##### *(Statische und dynamische Daten)*

Sensorische Farben besitzen die Eigenschaft, Daten von Einflussfaktoren zu speichern. Entsprechend sollte bereits vor der Entwicklung eines intelligenten Codes geklärt werden, ob Wechselwirkungen zwischen den Sensoren bestehen und ob ein Sensor ggf. auf mehrere Einflüsse reagiert. Hierzu ist das eingesetzte Druckverfahren für die Anzahl der sensorischen Tinten innerhalb des intelligenten Codes ausschlaggebend. Im Falle des kontaktlosen Inkjetdruckverfahrens ist die Anzahl der gleichzeitig im Code applizierbaren Sensoren durch die Anzahl der Druckköpfe des Druckers begrenzt. Beim Entwurf eines intelligenten Codes mit sensorischen Farben ist in Erwägung zu ziehen, inwieweit auch statische Daten im Code enthalten sein sollen und ob diese einer implementierten Fehlerkorrektur bedürfen. Zudem ist die Darstellung der Daten und des Datenformats festzustellen (bspw. kann ein vorhandenes Schriftsystem, z. B. wie die Brailleschrift, oder eine Binärcodierung verwendet werden). So erfordert die Gestaltung des intelligenten Codes eine sorgfältige Abwägung der Methode der Datenverarbeitung und der Größe des Datenbereichs.

### *Kriterium 3*

#### *Codeerkennung entsprechend der Zielsetzung (Algorithmen)*

Entsprechend der Auswahl des Suchmusters (Zielsetzung) und der Anordnung der Bereiche von dynamischen oder statischen Daten im intelligenten Code ist ein Algorithmus zu erstellen bzw. anzupassen, der den Code erkennt und auswertet. Die Wahl sowohl eines bereits bekannten als auch eines eigens entwickelten Algorithmus bedarf einer Evaluation der Funktionalität des Algorithmus, insbesondere im Hinblick auf die dynamischen Bereiche des Codes. Zur Evaluation und Anpassung eignen sich bereits kleine Prototypen des intelligenten Codes.

### *Kriterium 4*

#### *Verortung der Datenverarbeitung (Plattform, Datenbank)*

Vor der Entwicklung eines intelligenten Codes ist geeigneter Weise der Speicherort für die ausgelesenen Daten festzulegen, da diese im Algorithmus als Array selbst kurzfristig hinterlegt, als externe Datei oder in einer Datenbank gespeichert und abgerufen werden können. Dazu müssen die Daten jedoch zuvor strukturiert, organisiert (bspw. in Kategorien in einer Datenbank hinterlegt) und bei Bedarf visuell für ein vereinfachtes Verständnis dargestellt werden.

### *Kriterium 5*

#### *Art der Datenauswertung (Analyse, Algorithmen)*

Vor der Auswertung der Daten sind zunächst die Reaktionsbereiche und die Farbveränderungsverläufe der jeweiligen einzelnen sensorischen Farben grundlegend zu analysieren und für eine nachfolgende Interpretation zu identifizieren und zu validieren. In diesem Kontext können Schwellenwerte definiert werden, die zur Skalierung der Farbverläufe der jeweiligen sensorischen Farben dienen. Ebenfalls können Farbdifferenzen errechnet und die identifizierten Schwellenwerte der sensorischen Farben zur Interpretation gegenübergestellt werden.

## 5.3 Entwicklung und Aufbau eines intelligenten Codes

Der QR-Code als einer der bekanntesten 2D-Codes sticht mit seinen drei großen quadratischen Suchmustern markant hervor. Es finden sich jedoch auch Codetypen, wie der Grid Matrix Code oder der EZcode, die ebenfalls klare quadratische Grundformen als Suchmuster aufweisen. So ist angesichts der in Abbildung 75 dargestellten drei geometrischen Formen nicht untypisch, dass vermehrt quadratische Formen als Suchmuster verwendet werden, die sich anhand einer höheren Robustheit und geringeren Fehleranfälligkeit auszeichnen. In Abbildung 76 werden dazu quadratische Suchmustervariationen vorgestellt, auf deren Weiterentwicklung das Suchmuster des intelligenten Codes basiert.

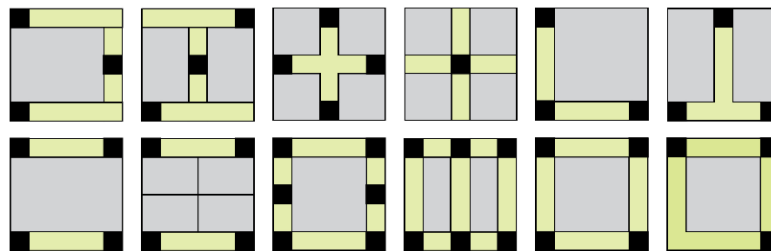


Abbildung 76 Varianten quadratförmiger Suchmuster

Hierzu wurden die entworfenen Variationen nach drei Teilbereichen in die Suchmuster (schwarz), die statischen Bereiche für Daten (grau) und die dynamischen Bereiche für sensorische Farben (grün) unterteilt und randomisiert angeordnet. So war es das Ziel, eine Form zu identifizieren, die einen zentriert-gebündelten statischen Bereich aufweist, in dem sich der Datenbereich klar erkennbar und von anderen Bereichen abgegrenzt. Darüber hinaus sollte die Anordnung und Anzahl der Suchmuster wenig Fläche verbrauchen, der Ausleseprozess der dynamischen Bereiche erleichtert und dafür eine relativ große Fläche bereitgestellt werden. Entsprechend dieser Kriterien wurde die vorletzte Variante (Abb., vorletzte Form unten rechts) ausgewählt, die wiederum durch das Streichen eines Suchmusters, ähnlich dem Vorbild des QR-Codes, auf drei Suchmuster reduziert wurde (Abb., unten rechts), um möglichst viele dynamische Bereiche für sensorische Farben bereitzustellen. Das fehlende vierte Suchmuster, kann jedoch durch die Koordinaten der drei anderen Suchmuster berechnet werden.

Nachdem diese Suchmusteranordnung als geeignetste identifiziert wurde, wurde ein Gestaltungsraster (Abbildung 77) auf Basis von  $29 \times 29$  Modulen entwickelt, wobei jeweils ein Modul (kleinstes Element) eine Abmessung von  $3 \times 3$  mm aufweist (Abb., links). Die quadratische Form des intelligenten Codes ermöglicht eine vereinfachtere Separierung (Kapitel 6.4) der jeweiligen Segmente für eine spätere Analyse. Die Abmessung eines Moduls ist dabei die kleinste sowie sicher zu erfassende Größe des in Kapitel 6.3 vorgestellten Algorithmus. Das Design des Gestaltungsrasters und die Modulgröße beeinflussen dabei die Anordnung der Komponenten (wie bspw. Suchmuster) des intelligenten Codes. Die Suchmuster basieren auf  $5 \times 5$  Modulen und werden von 11 Modulen (nicht mit Daten besetzt) in Form einer Ruhezone umrandet. Sie dienen der Abgrenzung und besseren Erkennbarkeit durch den eingesetzten Algorithmus (Abb., rechts). Zwischen den Ruhezonen befinden sich dynamische Bereiche (je 17 Module lang), die aus 12 Feldern, mit jeweils vier Modulen, für sensorische Farben bestehen. Einen Ausnahmefall bildet aufgrund des Wegfalls eines Suchmusters dabei der dynamische Bereich unten links, der aus 28 Feldern je vier Modulen besteht. Die Ruhezonen zwischen den Feldern mit je vier Modulen bedingen die ungerade Zahl des Gestaltungsrasters ( $29 \times 29$  Module).

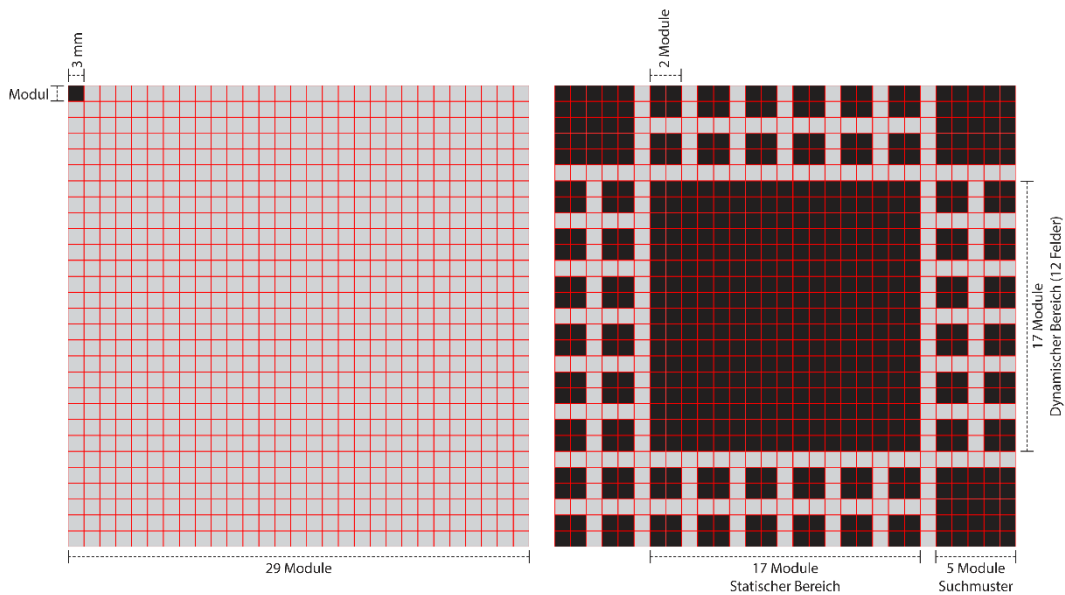


Abbildung 77 Gestaltungsraster auf Basis von Modulen

Der intelligente Code (Abbildung 78) entstand in einem iterativen Prozess und wurde unter Anwendung des eingesetzten Algorithmus (Kapitel 6.3) schrittweise optimiert und angepasst. Dieser Prozess ist interdependent zum einen zu der Auswahl der Algorithmen (Kapitel 6.1) für das Computer Vision und zum anderen zu der Kompatibilität des oben behandelten Gestaltungsraster (Kapitel 5.3) mit dem eingerichteten Algorithmus. So ist der abgebildete intelligente Code das Ergebnis dieses iterativen Prozesses.

Der intelligente Code weist im dynamischen Bereich verteilt drei verschiedene eigens entwickelten sensorische Farben auf, die bereits im Kapitel 4.7 behandelt wurden. Besonders hervorzuheben sind die Suchmuster, die neben ihrer Funktionalität als Identifikationsmarker, die Funktion als Sensoren aufweisen. Des Weiteren finden sich in der unteren linken Ecke des intelligenten Codes vier verschiedene Graustufenfelder, die für einen Algorithmus zur Helligkeits- und Kontrastanpassung verwendet werden (Kapitel 6.6.2). Das Zentrum des intelligenten Codes (17 x 17 Module) nimmt in Form eines Data-Matrix Codes (Kapitel 5.1.1) statische Daten auf.

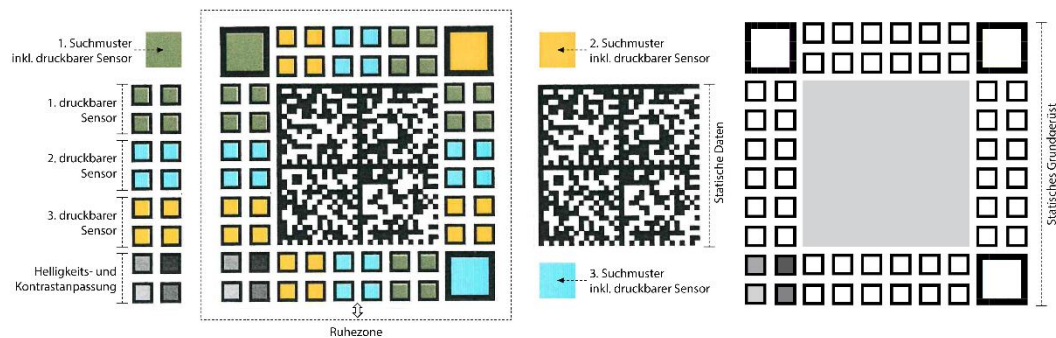


Abbildung 78 Aufbau eines intelligenten Codes

Außerhalb des intelligenten Codes dient eine Ruhezone mit einem Weißraum sowohl der einwandfreien Erfassung des intelligenten Codes als auch als Referenzfeld für die Eigenfarbe des Substrats zum Zwecke der Farbkorrektur seitens des dazu entwickelten Algorithmus (Kapitel 6.6.1). Dieses spezielle Design macht den intelligenten Code sehr robust. Denn auch bei einem potenziellen Ausbleichen bzw. Entfärben der dynamischen Bereiche ist das verbleibende statische Gerüst (schwarze Module) hinreichend

markant, um den Code zuverlässig zu erfassen und auszulesen. Beispielweise neigt gerade die ursprünglich intensiv blaue hydrochrome Farbe bei massivem Kontakt mit Wasser zum Entfärben (*Kapitel 4.7.1*), welches ohne das statische Gerüst, zum Verlust des Suchmusters führen würde.

## 5.4 Ausgabe des intelligenten Codes

Die Ausgabe des intelligenten Codes erfolgt durch einen Codegenerator der die Anzahl der sensorischen Tinten auswählt, erweitert und den statischen Bereich erzeugt. Alle notwendigen Bestandteile eines intelligenten Codes, die von einem Codegenerator erzeugt werden, sind bereits innerhalb dieses Kapitels *Aufbau und Funktion eines intelligenten Codes (Kapitel 5)* aufgezeigt worden. Die Funktionalität und der Aufbau eines Codegenerators werden innerhalb dieser Dissertation nicht behandelt, da dies zum einen den Rahmen dieser Dissertation übersteigen und zum anderen keinen weiteren Mehrwert zur Erfüllung der im *Kapitel 1.1* behandelten *Zielsetzung* bringen würde. Die drucktechnische Ausgabe des intelligenten Codes wurde bereits im Methodenteil des *Kapitel 3.3* beschrieben.

## Kapitel 6

# Softwarearchitektur

Das vorliegende Kapitel setzt sich mit der Gesamtheit der programmiertechnisch umgesetzten und serverseitig implementierten Softwarearchitektur auseinander. Zunächst stehen die Grundlagen zum Verständnis dieses Kapitels im Fokus. Daraufhin werden die eingesetzten und angepassten Algorithmen zur Erkennung und Erfassung des intelligenten Codes vorgestellt. Die entsprechend der *Zielsetzung (Kapitel 1.1)* entwickelten Instrumente dienen dabei dem Erfassen, Kontrollieren und Validieren der Auswirkungen von bestimmten Umwelteinflüssen auf ein Objekt.

## 6.1 Grundlagen der Bildverarbeitung und Computer Vision

Computer Vision (CV), „auch maschinelles Sehen genannt, umfasst verschiedene Methoden [der Interoperabilität zur] Erfassung, Verarbeitung, Analyse und Interpretation von Bildern“ (Priese, 2015, S. V). Neben einzelnen Bildern können jedoch auch Videosequenzen und Echtzeitvideoübertragungen, in Form von Bildwechselfrequenzen (frames per second), sequenziell verarbeitet werden. Somit können mit CV Bildinformationen aus der realen Umgebung in die digitale Umgebung transformiert werden. Dementsprechend ist CV ein interdisziplinäres und wissenschaftliches Feld, dass sich mit dem Einsatz und der Weiterentwicklung von Techniken befasst, die dem Computer helfen sollen den Inhalt eines einzelnen Bildes oder eines Videos zu analysieren oder zu verstehen.

Abbildung 79 visualisiert die Wechselwirkung zwischen einerseits technischen Komponenten, wie Objektive (Linsen), CMOS-Sensoren, in bspw. Smartphones, und andererseits digitalen Prozessen. Zu letzteren gehören, wie bspw. die Echtzeitbildverarbeitung, der Einsatz höherer Programmiersprachen (wie bspw. Python, inkl. diverser Programmibibliotheken, unter anderem CV). Das bedeutet, dass die Einspeisung der Informationen eines Objekts in die digitale Umgebung durch die Wechselwirkung beider Komponenten bedingt wird.

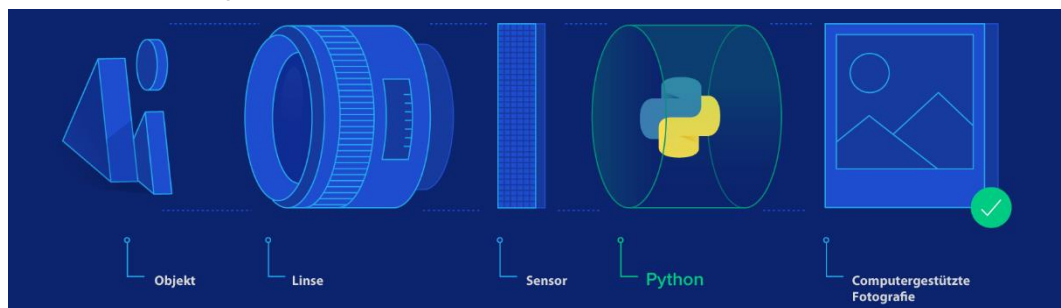


Abbildung 79 Computer Vision (Balaban, 2018)

Nach Priese (2015, S.2) findet bereits in der Kamera eine softwarebasierte Bildvorverarbeitung statt, die bspw. eine automatisierte Helligkeits- und Kontrastanpassung sowie weitere farbliche Änderungen vornimmt. „Diese für den menschlichen Betrachter schöneren Resultate verfälschen [...] den Bildinhalt und können eine anschließende Analyse des Bildes [...] empfindlich stören“ (ebd.). So wurde bereits im *Kapitel 3.7.1* auf die Notwendigkeit einer möglichst *gerätenutralen Auswertung* hingewiesen, indem methodisch direkt über einer Schnittstelle auf die Rohdaten der Kamera zurückgegriffen wird.

Eine Bildverarbeitung weist nach Süße und Rodner (2014, S. 211) folgende Schritte auf:

- I. Daten- oder Bildeingabe,
- II. Vorverarbeitung der Bilder zur Optimierung,
- III. Segmentierung der Bilder,
- IV. Klassifikation und / oder Analyse,
- V. Ausgabe.

Die vorgestellten Schritte finden sich ebenfalls im Entwicklungsprozess der Leseapplikation.

### 6.1.1 Pixel

Bereits in der Antike finden sich Mosaikbilder, die aus einzelnen Quadraten verschiedenfarbiger Gesteinsarten und ihrer entsprechenden Anordnung bestehen. Die Kunstform des Mosaiksteinlegens verhält sich ähnlich den Eigenschaften von digitalen Pixelgrafiken<sup>9</sup>, die bei einer niedrigen Bildauflösung „pixelig“ erscheinen. Entgegengesetzt verfließen die Übergänge bei höheren Auflösungen. Digitale Bilder bestehen ebenfalls aus mosaikartigen Formen die, als *Pixel* bezeichnet werden. Ein digitales Bild besteht in seiner Datenstruktur aus einem 3-Dimensionalen-Array (dreischichtige Matrix), bei der die dritte Dimension die verschiedenen Farbkanäle (bspw. R, G, B) des Bildes darstellt (Abbildung 80). Innerhalb des jeweiligen Kanals in der Array-Struktur nimmt jeder Pixelwert eine Graustufung von 0 bis 255 (8 Bit) an. Bei einer Ausgabe der Pixelinformationen werden die Werte kombiniert aus den drei Kanälen als sogenanntes Array ausgegeben. So kann auf die einzelnen Pixelwerte über die Zeilen- und Spaltenkoordinaten ( $i, j$ ) zugegriffen werden. Bei einem RGB-Bild werden die RGB-Werte in Form eines Arrays [155, 15, 20] und bei einem Graustufenbild lediglich die entsprechende Intensitätsstufe [115] ausgegeben. Dabei stellen sich die beispielhaften Pixelfarbwerte wie folgt dar: rot [255, 0, 0], orange [255, 100, 0], türkis [0, 255, 255].

	15	5	255	0	100		
	25	255	5	78	150	255	
Höhe	100	50	35	155	253	154	10
	5	15	50	84	8	150	5
	255	0	222	26	210	22	15
		122	125	253	2	63	85
			128	25	174	58	47

Abbildung 80 Bild als Array

#### 6.1.1.1 RGB-Farbmodell

Das RGB-Farbmodell wird in vielen digitalen Endgeräten auch als Farbprofile in verschiedenen Smartphones eingesetzt. Wie zuvor beschrieben, lässt sich in diesem Modell aus den unterschiedlichen Intensitätsstufen (Graustufungen) der Farbkanäle ( $256^3$ ) eine große Bandbreite an Farben (theoretisch ca. 16,8 Millionen Farben) durch additive Farbmischung hervorbringen (Süße & Rodner, 2014, S. 246 f.). Der detaillierte Aufbau und die Berechnung des RGB-Farbmodells sind in einschlägigen Lehrbüchern beschrieben.

<sup>9</sup> Pixel sind die kleinste Informationseinheit, aus der ein Bild besteht.



### 6.1.1.2 Graustufen

CV benötigt für seine Algorithmen häufig ein Bild mit 8 Bit Graustufen (monochromatisch,  $2^8 = 256$ ), die in einem 2-Dimensionalen-Array gespeichert werden. Die Graustufenkonvertierung eines Farbbildes für CV ist notwendig, um die Verarbeitungszeit und die Bildinformationen, auf die wesentlich zu verarbeitende Größe zu verringern. Für die Konvertierung eines Farbbildes in ein Graustufenbild kommen statistische Verfahren des gewichteten arithmetischen Mittelwerts zum Einsatz – dessen Summe eins beträgt, bei dem die Werte jedes Farbkanals ( $x_i$ ) unterschiedlich stark gewichtet ( $w_i$ ) in die Berechnung des gewichteten arithmetischen Mittelwerts einfließen. Die Berechnung erfolgt aus einem RGB-Farbbild ( $Y$ ) heraus, bei der die folgende gewichtete Summe zum Tragen kommt, wie die folgende Formel zeigt (OpenCV, 2020b) ( 27 ):

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B$$

$$\sum_{i=1}^n w_i = 0,299 (w_1) + 0,587 (w_2) + 0,114 (w_3) = 1$$

$$\bar{x}_w \sum_{i=1}^n w_i \cdot x_i = w_1 \cdot x_1 + \dots + w_n \cdot x_n \quad (27)$$

$$\frac{\bar{x}_w \sum_{i=1}^n w_i \cdot x_i}{\sum_{i=1}^n w_i}$$

## 6.1.2 Schwellenwertverfahren

Bei einer Bildsegmentierung werden die Pixelpunkte eines Bildobjekts nach ihren Gemeinsamkeiten und Zugehörigkeiten untersucht. Das Ziel ist es dabei, die identifizierten Teilbildobjekte für eine nachfolgende Analyse zu zerlegen (segmentieren). Das resultierende Ergebnis ist ein Binärbild. Der Einsatz von Schwellenwertverfahren ist ein wichtiger Schritt hin zur Kantenerkennung, Segmentierung oder zu weiteren generellen Methoden der Bildanalyse. Am einfachsten ist es, ein Binärbild ausgehend von einem definierten globalen Schwellenwert ( $T$ ) zu erstellen. Hier kann über statistische Verfahren ein Schwellenwert berechnet oder vorgegeben werden, bei dem die Pixelwerte ( $i, j$ ) eines Graustufeneingabebilds  $f(i, j)$  in ein Binärausgabebild  $f'(i, j)$  überführt und jeweils einem Wert 0 oder 1 zugewiesen werden (Stüße & Rodner, 2014, S. 212) ( 28 ):

$$f'(i, j) = \begin{cases} 1, & \text{falls } f(i, j) \geq T \\ 0, & \text{falls } f(i, j) < T \end{cases} \quad (28)$$

### 6.1.2.1 Anpassungsfähige Schwellenwertverfahren

Eine weitere Möglichkeit der Erzeugung eines Binärbildes ist die dynamische Binarisierung. Diese bietet den Vorteil, gerade bei variierenden Lichtverhältnissen anpassungsfähig zu sein. Durch die mehrfache Schwellenwertberechnung an unterschiedlich eingestellten Regionen des Bildes (Aufnahme), können verschiedene Nachbarschaftswerte  $N(i, j)$  bei der Schwellenwertberechnung berücksichtigt werden, um ein bestmögliches Binärbild zu erzeugen. Diese Berechnung wird unter anderem mittels eines statistischen Verfahrens erzielt, wie bspw. dem arithmetischen Mittelwert oder durch einen gewichteten arithmetischen Mittelwert, bei dem eine zuvor definierte Konstante von der gewichteten Summe der Nachbarschaftswerte subtrahiert wird.

### 6.1.3 Morphologische Transformationen

Morphologische Transformationen sind einfache Operationen auf der Grundlage einer binären Abbildungsform, mit dem Ziel der Manipulation der sich im Vordergrund befindenden Objekte (Vordergrundobjekt). So wird angenommen, dass sich die im Bild verorteten Objekte innerhalb eines Bildrahmens befinden. Um die Objekte an den Randbereichen verarbeiten zu können, muss ein zusätzlicher Schwarzraum (Rand mit schwarzen Pixeln), im Sinne einer Randbehandlung, erweitert werden (Priese, 2015, S. 156).

Die zu Grunde liegenden morphologischen Operatoren sind unter anderem:

#### I. Erosion $\ominus$

Ähnlich dem Erodieren des Bodens, fungiert ein Erosionsoperator (Abbildung 81). Es werden dabei jedoch die weißen<sup>10</sup> Bereiche des Vordergrundobjekts in einem Binärbild erodiert (verringert). D. h., dass weiße Bereiche verringert und dunkle Bereiche erweitert werden, denn die Erosion entfernt Rauschen, jedoch schrumpft die Erosion auch das zu erodierende Objekt. Hierbei werden die Pixel des binären Bildes durch eine binäre Filtermaskenmatrix, auch als *strukturierendes Element* bezeichnet, verändert. Dabei besitzt die Filtermaske ein eigenes Koordinatensystem im Mittelpunkt der Matrix (*anchor Pixel*). Hierzu wird das strukturierende Element (S) als abfragender Operator pixelweise über das Binärbild (B) verschoben und überprüft die jeweilige Übereinstimmung mit dem Operator – findet keine Übereinstimmung statt, wird der nicht übereinstimmende Pixel erodiert ( $\ominus$ ). Dabei gilt, dass der Operator ein Pixel vom Wert 1 zum Wert 0 ändert, wenn mindestens ein Nachbapixel 0 vorhanden ist und das Strukturelement vollständig in die Bildregion hineinpasst (ebd., S. 157).

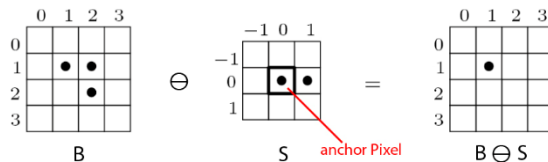


Abbildung 81 Prinzip der Erosion (Burger & Burge, 2005)

#### II. Dilatation $\oplus$

Das Gegenteil von Erosion ist Dilatation (Erweiterung) (Abbildung 82). Dabei gilt, dass das strukturierende Element beim Auftreffen auf einen Pixel (Wert 1) diesen um seinen festgelegten Operator erweitert. Normalerweise folgt auf Erosion in vielen Fällen, wie bspw. der Rauschunterdrückung, eine Dilatation, um bspw. die verjüngten Konturen nachträglich zu verstärken. Dieser Operator der Dilatation ist ebenfalls nützlich, um weggebrochene Teile eines Objekts zusammenzufügen (ebd.).

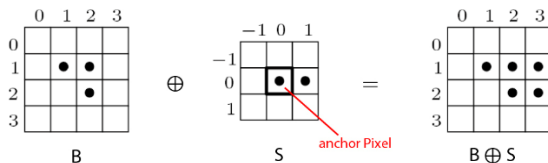


Abbildung 82 Prinzip der Dilatation (Burger & Burge, 2005)

<sup>10</sup> In Binärbildern wird der Hintergrund häufig mit dem Wert 0 und Objekte im Vordergrund mit dem Wert 1 versehen.

## 6.1.4 Lineare Filteroperationen (Gradient)

Die Umwandlung eines Graustufenbildes in ein Gradientenbetragsbild zur nachfolgenden Kantenerkennung ist Inhalt dieses Abschnitts. Damit die Konturen durch die nachfolgenden Algorithmen optimal verarbeitet werden, sollte ein Gradientenbetragsbild erzeugt werden, bei dem das Objekt innerhalb des Bildes abgegrenzt wird. Ein Gradientenbetragsbild weist dazu an den Objektgrenzen starke Gradienten auf (steiler Anstieg an Grauwertübergängen), die anzeigen, in welcher Richtung eine Änderung am größten oder auch keine Änderung aufzufinden ist. Der berechnete Gradient ( $\nabla$ ) zeigt dabei die Richtung  $(x, y)$  der ausgeprägtesten Grauwertveränderung eines Pixels  $f(i, j)$  an und wird mit  $\nabla f(i, j)$  bezeichnet (Süße & Rodner, 2014, S. 443). Der Sobel-Operator kann als Filteroperator ( $h$ ) zur sogenannten Faltung<sup>11</sup> eingesetzt werden. Dabei werden bei einer Filterung die Bildpixel in Abhängigkeit der Nachbarpixel manipuliert. Der Operator berechnet (Ableitung in jeweils  $x$  und  $y$ ) dabei die Steigung für jeweils den Gradienten in Richtung  $h_x$  und  $h_y$ , zur Bestimmung horizontaler und vertikaler Kanten (ebd., S. 45) (31):

$$\begin{aligned} \nabla f(i, j) &= \begin{pmatrix} f(i, j) \cdot h_x \\ f(i, j) \cdot h_y \end{pmatrix} \\ \frac{\Delta f(i, j)}{\Delta i} &\approx f(i, j) \cdot h_x \text{ mit } h_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \\ \frac{\Delta f(x, y)}{\Delta j} &\approx f(i, j) \cdot h_y \text{ mit } h_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \end{aligned} \quad (29)$$

Zusätzlich wird ein korrigierender Filterabschnitt  $[1, 2, 1]$  (Filterkoeffizienten, Gewichte) angewandt, mit dem eine gewichtete Mittelwert-Bereinigung durchgeführt wird, die störende Pixel an den Seitenbereichen beseitigt. Abschließend kann unter anderem durch die Berechnung der Hypotenuse (30) oder der Summe der Absolutwerte aus den beiden erzeugten Bildern ( $h_x$  und  $h_y$ ) ein Gradientenbetragsbild mit hervorgehobenen Kanten generiert werden, das anschließend zu einem Binärbild transformiert (Kapitel 6.1.2) wird.

$$|\nabla f(i, j)| = \sqrt{\left(\frac{\Delta f(i, j)}{\Delta i}\right)^2 + \left(\frac{\Delta f(i, j)}{\Delta j}\right)^2} \approx \left|\frac{\Delta f(i, j)}{\Delta i}\right| + \left|\frac{\Delta f(i, j)}{\Delta j}\right| \quad (30)$$

## 6.1.5 Hough-Transformation

Die Kantendetektion von Hough (1962) ist ein Verfahren zur Erkennung von Linien, Kreisen und parametrisierbaren geometrischen Formen, wie bspw. Quadraten. Dieses Verfahren dient dabei dem Verstehen von Bildern, spezifischer dem Erkennen von Formen, die sich innerhalb eines Bildes befinden. Wie auch zuvor wird auch in diesem Verfahren zunächst aus einem Graustufenbild ein Gradientenbetragsbild und schließlich über ein Schwellenwertverfahren ein Gradienten-Binärbild erzeugt, das nachfolgend analysiert wird. Der Einsatzzweck kann dabei neben vielen Bereichen bspw. in der Erkennung von Straßen- und Bahnbegrenzungen für das autonome Fahren liegen oder es können Industrieroboter oder die Suchmuster von intelligenten Codes sein. Eine Linie im Bildraum kann mit der Hough-

<sup>11</sup> „Bei einer Faltung berechnet sich jeder Pixel des Ausgangsbildes als gewichtete Summe der Pixel einer Bildnachbarschaft.“ (Konen und Zielke, 2006, S.3)

Transformation dabei mit zwei Parametern ausgedrückt werden. Zum einen im kartesischen Koordinatensystem ( 31 ) mit den Parametern ( $m^{12}$ ,  $b^{13}$ ) und zum anderen im Polar-Koordinatensystem, mit den Parametern ( $r$ ,  $\alpha$ ) ( 32 ). Innerhalb der Hough-Transformation wird für eine nachfolgende Erkennung von geometrischen Formen ein Dualraum aufgebaut (Abbildung 83). Dieser besteht zum einen aus einem Ortsraum (Gradienten-Binärbild; Abb., links) im Euklidischen Koordinatensystem ( $x$ ,  $y$ ) und zum anderen aus einem Parameterraum in Form eines Akkumulators zum Speichern von Werten (2D-Array, dessen Elemente inkrementiert werden können; Abb., rechts). Der Parameterraum weist eine zentrierte Achse auf, die für die Steigung ( $y$ ) einer Geraden und die Höhe ( $r$ ) als  $y$ -Achsenabschnitt entspricht (OpenCV, 2020d; Haenselmann, 2016, 0:44:40 ff.):

$$y = m \cdot x + b \quad (31)$$

$$y = \frac{\sin(\alpha)}{\cos(\alpha)} \cdot x + \left( \frac{r}{\sin(\alpha)} \right) \quad (32)$$

In einem Gradienten-Binärbild können mehrere (zusammenhängende) Punkte bzw. Pixel ( $P$ ) eine Linie formen. Damit ein Zusammenhang zwischen den Punkten durch die verarbeitende Einheit (Computer) nachvollziehbar wird, erfolgt die Hough-Transformation. So wird dabei zunächst ein Punkt vom Ortsraum in den Parameterraum transformiert, in dem für jede mögliche Gerade, die durch einen Punkt im Ortsraum verläuft, der Winkel ( $\alpha$ ) und der  $y$ -Achsenabschnitt berechnet wird und diese Werte ( $\alpha$ ,  $r$ ) in den Akkumulator im Parameterraum an der Stelle ( $\alpha$ ,  $r$ ) eingeordnet werden. Somit entstehen durch die Berechnung mehrerer Punkte ( $P_{1-3}$ ) aus dem Ortsraum in den Parameterraum unterschiedlich geschwungene Kurven, die sich in einem Schnittpunkt ( $H$ ) nahezu mit einer Streuung schneiden (ein Cluster ergeben) und somit durch ihre Gesamtanzahl auf eine verbindende Gerade hinweisen. Somit kann im Allgemeinen eine Linie durch die Anzahl der Schnittpunkte zwischen den Kurven erkannt werden. Dabei gilt, je mehr Kurven sich schneiden, desto mehr Punkte hat die durch diesen Schnittpunkt dargestellte Gerade. Wenn die Anzahl der Schnittpunkte über einem bestimmten Schwellenwert liegt, deklariert die Hough-Transformation diese als Linie mit den Parametern ( $\alpha$ ,  $r$ ) des Schnittpunktes (Priese, 2015, S. 208 ff.). „Duda und Hart schlugen [...] die deutlich vorteilhaftere Hesse-Form vor.“ (Priese, 2015, S. 210) Dennoch wird zum besseren Verständnis einer Kantendetektion im Sinne des Grundlagenprinzips auf die Variante von Hough (1962) zurückgegriffen.

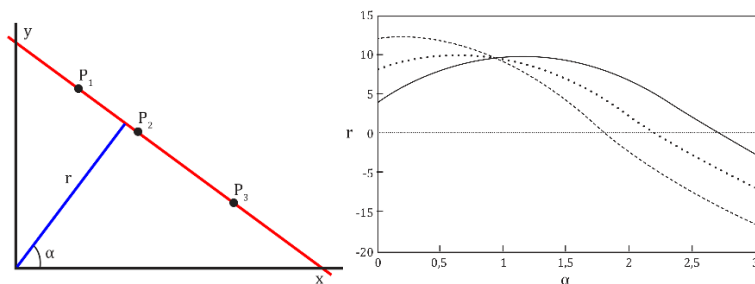


Abbildung 83 Dualraum: Ortsraum und Parameterraum, mit Schnittpunkt ( $H$ ) (OpenCV, 2020d)

<sup>12</sup>  $m$  für Steigung mit einer Steigung mit Schnittpunkt  $y$ .

<sup>13</sup>  $b$  für Höhe.

## 6.1.6 Suzuki-Abe Algorithmus

Die topologische Strukturanalyse digitalisierter Binärbilder auf der Basis eines Border-Following-Algorithmus (Randverfolgung) von Suzuki, S. und Abe (1985), ist ein Algorithmus zur Konturerkennung und Konturextraktion. Der Algorithmus von Suzuki und Abe definiert dabei die hierarchischen Beziehungen (*Kapitel 6.1.7*) zwischen den Grenzen (Rändern) von Vordergrundobjekten in Binärbildern. So unterscheidet der Algorithmus (Abbildung 84) auch zwischen dem äußeren Rand (Außenkontur eines Vordergrundobjekts) oder dem Lochrand (Zwischenbereich zwischen einem und mehreren Vordergrundobjekten). Für eine Bildverarbeitung benötigt der Algorithmus von Suzuki und Abe ein Binärbild, das vorübergehend um Pixelwerte mit 0 erweitert wird (Pixelwerte mit 0 als weiße Umrandung). Durch diesen Vorgang wird gewährleistet, dass mögliche Pixelwerte mit 1 in den Grenzen des Binärbilds identifiziert werden können. Danach wird das Binärbild ohne die erweiterten Pixelwerte im Randbereich auf Pixelebene gescannt, um diskrete Ränder zu identifizieren. Der erste Schritt eines Algorithmus zur Randverfolgung ist dabei in der Regel die Identifizierung des ersten Randpixels an der Stelle  $f(i, j)$  (Startpunkt) eines Vordergrundobjekts, in dem eine Kontur als geordnete Folge von Randpixeln betrachtet werden kann (a). Hierbei ist das Anfangspixel bedeutend, da dieses gespeichert und in der Regel mit Richtungs-codes (Freeman-Code) von 0 bis 7 versehen wird. Der Freeman-Code kodiert dabei pixelweise eine Kontur durch die Angabe der Richtung (bspw. Himmelsrichtung) der Lage des nächsten benachbarten Pixels. Somit ist es möglich, die Strecke zwischen zwei Pixeln (zusammenhängende Komponenten) in Binärbildern zu beschreiben. Der Algorithmus von Suzuki und Abe scannt zunächst zeilenweise das Binärbild von links nach rechts ab, bis ein Objektpixel als Startpunkt (Pixelwert 1) gefunden wurde. Dabei wird geschaut, ob es sich um einen äußeren Rand oder einen Lochrand (bspw. bei einer Punze) handelt. Die Kriterien für die Überprüfung eines äußeren Randes (a) oder Lochrandes (b) sind dabei:

$$i \begin{array}{|c|c|} \hline j-1 & j \\ \hline 0 & 1 \\ \hline \end{array} \quad i \begin{array}{|c|c|} \hline j & j+1 \\ \hline \geq 1 & 0 \\ \hline \end{array}$$

Abbildung 84 (a) äußerer Rand, (b) Lochrand (Suzuki & Abe, 1985, S. 35)

Dabei durchläuft der Algorithmus von einem benachbarten Pixelwert 0 beginnend gegen den Uhrzeigersinn alle benachbarten achter Nachbarschaften (3 x 3 Matrix, mit Pixel als Mittelpunkt) des Startpunkts, bis ein Pixelwert 1 und mit ihm eine Fortführung eines äußeren Rands gefunden wird. Sobald ein Pixel mit dem Wert 1 angetroffen wird, nimmt der Algorithmus das Pixel als Mittelpunkt einer 3 x 3 Matrix und überprüft die acht umliegenden Pixel. Wenn alle äußeren Pixel einen Pixelwert von 0 aufweisen, wird das zentrale Pixel auf -2 gesetzt, um das identifizierte Einzelpixel anzuzeigen. So wird das zentrale Pixel als 2 markiert, wenn der Wert seines rechten Nachbarpixels ungleich Null ist. Entsprechend wird die Randverfolgung bei diesem erkannten Pixelwert von 1 fortgesetzt und dessen benachbarte achter Nachbarschaft erneut gegen den Uhrzeigersinn durchlaufen und fortlaufend wiederholt. Durch diese Wiederholung werden im Bild alle zusammenhängenden Komponenten und ihre Punkte (P) ausfindig gemacht (Konturpunkte Liste) und daraus in einem nachfolgenden Schritt ein Polygon, bspw. in Form einer konvexen Hülle (*Kapitel 6.1.8*), erzeugt (Süße & Rodner, 2014, S. 213 f.). Eine detailliertere Beschreibung kann in der Veröffentlichung von Suzuki & Abe (1985) eingesehen werden.

## 6.1.7 Hierarchische Konturen

Um die zuvor erkannten Konturen in eine Struktur zu bringen, werden unterschiedliche Sortierverfahren eingesetzt. Dabei können die identifizierten Konturen eines Bildes unter anderem von der äußeren zur inneren Struktur hin oder auch nach hierarchischer Struktur sortiert werden. Dies ist vor allem bei der Identifizierung von Suchmustern relevant. Hierzu werden verschiedene statistische Verfahren eingesetzt, die bspw. die Konturen entsprechend dem Flächeninhalt nach Median, Mittelwert, minimales und maximales Element, etc. sortieren. Bei der sogenannten Eltern-Kind-Beziehung (engl. parent-child relationship), werden alle erkannten (verschachtelten) Konturen abgerufen und daraus resultierend eine Liste in Form einer Baumstruktur erstellt. Hierin kann die hierarchische Struktur bildhaft als eine verschachtelte Figur betrachtet werden, bei der die äußere Schale als Elternteil und die innere Schale, als Kind fungieren. Auf diese Weise haben die Konturen in einem Bild eine zueinander abhängige Beziehung. So kann definiert werden, wie die Konturen miteinander verbunden sein sollen. Hierzu werden die hierarchischen Strukturen in mehrdimensionalen Arrays ausgegeben, in der jede Kontur Informationen über diese Strukturen beinhaltet (OpenCV, 2020c).

## 6.1.8 Konvexe Hüllen

Die Koordinaten der identifizierten Suchmuster können für die Generierung einer Außenkontur bspw. zur Entnahme eines Codes verwendet werden. Damit Ansammlungen von Punkten eines Objekts innerhalb einer Liste approximiert (angenähert) werden können, kann die individuelle Form einer konvexen Hülle verwendet werden. Dabei heißt eine Menge ( $M$ ) konvex, wenn zu je zwei Punkten  $P_0, P_1$  in  $M$  die Strecke, nachfolgend als Strahl bezeichnet, nicht verlässt (Priese, 2015, S. 273). Damit die konvexe Hülle einer Menge (Objekt) identifiziert werden kann, wird ein Strahl ( $P_0 \rightarrow P_1$ ) ausgehend von  $P_0$  berechnet. Dieser muss jedoch mindestens auf der konvexen Hülle liegen. Um dies zu überprüfen, wird dabei innerhalb des kartesischen Koordinatensystems ( $x, y$ ), bspw. ein  $y$ -Minimum und  $y$ -Maximum berechnet, dass jeweils die Punkte der Ober- und Untergrenzen eines Objekts identifiziert. Hierzu wird ausgehend vom Strahl  $P_0$  gegen den Uhrzeigersinn rotiert und die dabei berührten näheren Punkte werden somit fortlaufend nummeriert ( $P_0 (n+1)$ ). Bei zwei gleichzeitig identifizierten Punkten, wird lediglich der entfernteste Punkt berücksichtigt und der andere wird entfernt. Der eigentliche Algorithmus basiert dabei auf einer Schleife, die zunächst die erste identifizierte Kante ( $P_0 \rightarrow P_1$ ) in einem Stapelspeicher ablegt. Dazu wird der nächste Punkt ( $P_n$ ) und die davorliegende Kante ( $k$ ) betrachtet, wobei der Raum ausgehend von  $k$ , in einen linken und rechten Halbraum aufgeteilt wird (zur Identifizierung von Konvexitätsfehlern). Anhand einer bedingten Anweisung wird geprüft, ob  $P_n$  im linken Halbraum liegt. Falls diese Anweisung zutrifft, wird eine neue Kante vom Endpunkt von  $k$  nach  $P_n$  erzeugt und erneut auf den Stapelspeicher abgelegt. Trifft die Anweisung nicht zu, wird die letzte Kante vom Stapel entfernt und mit dem nächsten Punkt fortgefahren (Haenselmann, 2017, 7:54 ff.). Dieser Vorgang wird dabei fortlaufend wiederholt. So kann der Algorithmus gezielt eingesetzt werden, um eine Kurve auf Konvexitätsfehler zu prüfen und diese ggf. zu korrigieren. So können bspw. auch die quadratischen Suchmuster des intelligenten Codes identifiziert und die konvexe Hülle ausgeschnitten und für einen weiteren Teilprozessschritt eines Algorithmus zur Separierung (Kapitel 6.4) entnommen werden.

Alternativ können bei rechteckigen Objekten auch ein Bounding Box Algorithmus (ebd.) verwendet werden, der die Ecken, ausgehend der Menge  $M$ , aus den kleinsten und größten vorkommenden  $(x, y) \cdot (\min. / \max.)$  Koordinaten ermittelt. Um eine perspektivische Transformation durchführen zu können, werden jedoch ausgehend von der konvexen Hülle noch die Rotationsparameter (Rotated Rectangle) ermittelt.

## 6.1.9 Perspektivische Transformation

Perspektivische Verzerrungen finden sich bspw. in Bildaufnahmen von Kameras mit einem Weitwinkelobjektiv, die nicht über einen entzerrenden Algorithmus verfügen. So erscheinen bei einer Aufnahme von bspw. einem Schachbrettmuster mit deutlichen Geraden, leichte und unerwünschte Krümmungen, wie bspw. im extremen Fall bei einem Fischaugenobjektiv. Ebenfalls bekannt bei Aufnahmen von Papierformaten wie bspw. bei Dokumentenscannern, wird die perspektivische Verzerrung entsprechend der Oberansicht einer geometrischen Perspektive korrigiert. So wird im Folgenden die geometrische Transformation im Sinne einer perspektivischen Verformung (Perspective Warping) behandelt. Zur perspektivischen Korrektur werden zunächst die äußeren Punkte eines Rahmens und ihre Koordinaten in Form einer Transformationsmatrix benötigt, die bspw. als Array vorliegt. Eine Transformationsmatrix ist eine Matrix, die beschreibt, wie Bildpixel an eine neue Stelle umgeordnet werden, um ein transformiertes Bild zu erzeugen. Dabei erfolgt die Berechnung der perspektivischen Transformation anhand von vier Punktpaaren. Hierbei wird die perspektivische Transformation berechnet und durch eine  $3 \times 3$ -Matrix mit acht Freiheitsgraden  $(a_{1-4}, b_{1-2}, c_{1-2})$  definiert. Dabei wird  $a_{1-4}$  in Form einer Transformation, wie bspw. Rotation, Skalierung etc.,  $b_{1-2}$  als Verschiebungsvektor und  $c_{1-2}$  als Projektionsvektor definiert. Bei der Anwendung einer perspektivischen Transformation auf ein Eingabebild  $f(i, j)$  wird unter Verwendung der angegebenen Matrix ( $M$ ) ein transformiertes Ausgabebild  $f'(i, j)$  ausgegeben (OpenCV, 2020e) (33):

$$\begin{bmatrix} f'_i \\ f'_j \end{bmatrix} = M \cdot \begin{bmatrix} f_i \\ f_j \end{bmatrix} \rightarrow \begin{bmatrix} f'_i \\ f'_j \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ c_1 & c_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} f_i \\ f_j \\ 1 \end{bmatrix} \quad (33)$$

### 6.1.9.1 Rotation

Das Ziel ist es, das ausgeschnittene Bild eines intelligenten Codes für einen nachfolgenden Prozess zu drehen: Der Drehwinkel ( $\alpha$ ) soll dabei gegen den Uhrzeigersinn auf eine bestimmte Position rotieren. Bei der Rotation ( $R$ ) wird jeder Bildpunkt  $f(i, j)$  in einen neuen Bildpunkt  $f'(i, j)$  um den Winkel  $\alpha$  in einer Transformationsmatrix  $R(\alpha)$  überführt (34) (Süße & Rodner, 2014, S. 269):

$$\begin{bmatrix} f'_i \\ f'_j \end{bmatrix} = R(\alpha) \cdot \begin{bmatrix} f_i \\ f_j \end{bmatrix} \rightarrow \begin{matrix} f'_i = \cos(\alpha) \cdot f_i - \sin(\alpha) \cdot f_j \\ f'_j = \sin(\alpha) \cdot f_i + \cos(\alpha) \cdot f_j \end{matrix} \quad (34)$$

## 6.2 Erkennungs-, Analyse- und Auswertungsprozess

Im Folgenden werden die entwickelte Gesamtstruktur der Softwarearchitektur (Abbildung 85) erläutert und ihre Zusammenhänge aufgezeigt (Bilgin & Backhaus, 2020d). Hierzu werden neun Prozessschritte vorgestellt. Zusätzlich sind die Softwarekomponenten im Methodenteil (*Kapitel 3.8*) aufgeführt. Die Grundlagen und die entsprechende Vertiefung der jeweiligen Kommunikationstechnologien, können dem *Kapitel 6.7* entnommen werden. Die Abbildung zeigt den zusammenhängenden Prozess, bei dem ein intelligenter Code erkannt, analysiert ausgewertet und dargestellt wird.

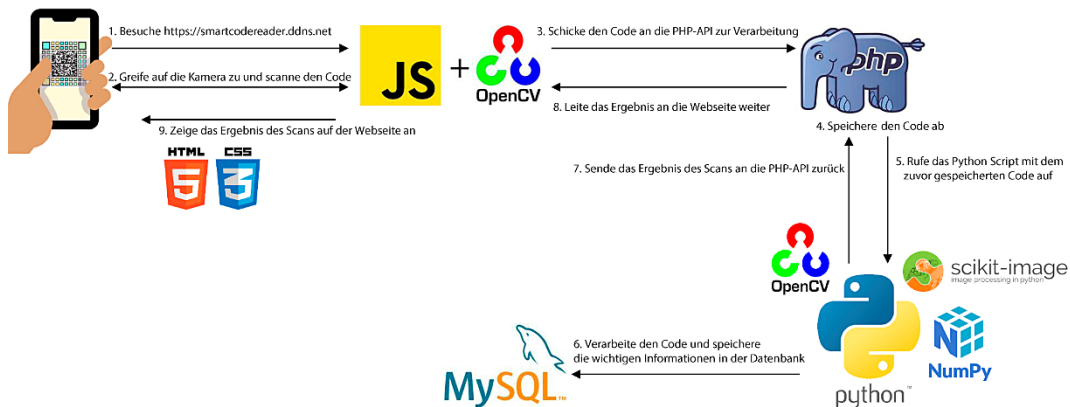


Abbildung 85 Softwarearchitektur (Legende in Tabelle 13)

1. **Domainzugriff**  
Das Erfassen des intelligenten Codes (*Kapitel 5.3*) wird über eine auf dem Smart Device vorinstallierte und als Benutzerschnittstelle fungierende Client Applikation (Browser), im Sinne eines Thin Clients, initiiert, der eine Kommunikation gewährleistet. So wird über einen Domainnamen eine HTTP / TLS verschlüsselte bidirektionale Verbindung auf die serverseitig hinterlegte Leseapplikation nach dem Prinzip eines Client-Server-Modells zugriffen. Serverseitig sind mehrere Algorithmen für die Codeerfassung unter anderem auf Basis des maschinellen Sehens (Computer Vision) und für eine entsprechende Analyse implementiert. Dadurch gewährleistet die eingesetzte Softwarearchitektur zum einen eine Geräteunabhängigkeit, da über verschiedene Smart Devices und auch über Desktop-Computer der Zugriff ermöglicht und zum anderen eine Plattformunabhängigkeit geschaffen wird, da weder Apps noch sonstige spezifische Software zu implementieren und zu warten sind.
2. **Kamerazugriff**  
Die Leseapplikation auf dem Server (Host) greift nach einer erfolgreichen Initialisierung durch eine Anfrage eines Clients (Endgerät) auf die integrierte Rückkamera des Endgeräts zu (*Kapitel 3.8*) und leitet nach einem automatisierten Erkennungsprozess über die Suchmuster des intelligenten Codes den Ausschnitt des Codes für eine nachfolgende Analyse weiter. Die Leseapplikation basiert dabei auf einem responsive Webinterface (*Webschnittstelle*), das sich automatisch an die Bildschirmauflösung der jeweiligen Endgeräte anpasst. So wird zudem, durch den Einsatz von HTML5 (Hypertext Markup Language), eine Auszeichnungssprache und CSS3 (Cascading Style Sheets), eine Stylesheet-Sprache die grafische Benutzeroberfläche (GUI) für eine abgestimmte Darstellung der Inhalte auf der Plattform gewährleistet. Der Algorithmus wurde für den Erkennungsprozess des Codes auf der Skriptsprache JavaScript (JS) inklusive einer zugeschnittenen Computer Vision (OpenCV.js) Programm-bibliothek entwickelt.



3. API-Schnittstelle  
Die API-Schnittstelle (Anwendungsprogrammierschnittstelle) auf Basis der Skriptsprache, PHP (Hypertext Preprocessor) baut eine Verbindung zwischen zwei Softwarekomponenten auf und ermöglicht den Transfer des zuvor entnommenen intelligenten Codes. Dabei handelt es sich bei den beiden Softwarekomponenten um einen nachfolgend zu behandelnden Algorithmus (*Kapitel 6.6.3*) zur Codeerkennung und um die nachfolgend zu erläuternden Schritte, die über eine API-Schnittstelle kommunizieren.
4. Datensicherung  
Der über die API-Schnittstelle transferierte Code wird in einer mit einem Zeitstempel versehenen und vorbestimmten Verzeichnisstruktur für eine nachfolgende Analyse, im Sinne einer Referenzierung, auf dem Server abgelegt und somit gesichert. Abschließend erfolgt die anschließende Initiierung des nachfolgenden Prozesses (5).
5. Analysieren und Referenzieren  
Ausgehend von der Initiierung, wird das in der höheren Programmiersprache Python geschriebene Skript zur Analyse und Referenzierung des intelligenten Codes abgerufen. Das Python-Skript setzt dabei speziell auf Python zugeschnittene Programmbibliotheken ein, wie Numpy, das unter anderem n-dimensionale Arrays und Werkzeuge für numerische Berechnungen zur Verfügung stellt. Des Weiteren wird die Pylibdmx Programmbibliothek für das Dekodieren und Generieren eines DataMatrix Codes verwendet und abschließend wird die Computer Vision Programmbibliothek von OpenCV eingesetzt. Das Python-Skript auf Basis von Computer Vision greift dabei für die auszuführenden Analyse- und Verarbeitungsschritte auf die zuvor in einem Dateiverzeichnis abgelegte Datei zu. Die Grundlagen zum Verständnis der Programmstruktur wurden bereits in *Kapitel 6.1* ausführlich behandelt. Die aus der Analyse des intelligenten Codes resultierenden Informationen werden dabei über die API-Schnittstelle an die Datenbank transferiert und für den 7. Prozessschritt vorbereitet.
6. Datenbankeinträge  
Die im vorherigen Prozessschritt analysierten Informationen des intelligenten Codes werden in die bereits Klassifikationen beinhaltende Datenbank (MySQL) eingetragen, gesichert und somit verarbeitet.
7. Ergebnisbericht  
Die in der vorherigen Datenbank verarbeiteten Informationen sowie weitere Informationen, von vorherigen Analyseschritten, andere Codezustände (Historie des intelligenten Codes), inklusive ihrer entsprechenden Datenbankklassifizierung, werden dabei zusätzlich ausgewertet und direkt an die API-Schnittstelle transferiert.
8. Weiterleitung  
Die Informationen des statischen Bereichs sowie der dynamischen Bereiche werden von der API-Schnittstelle direkt an das Webinterface weitergeleitet.
9. Ergebnisdarstellung  
Die von der API-Schnittstelle an das Webinterface transferierten Ergebnisse werden in Form eines Ergebnisberichtes visuell aufgearbeitet und an der Benutzerschnittstelle dargestellt.

### 6.3 Erkennungsprozess des intelligenten Codes

Die zuvor behandelten Grundlagen (*Kapitel 6.1*) dienen dem Verständnis des vorliegenden Kapitels. Der Erkennungsprozess des intelligenten Codes, wird anhand einzelner Snippets (*kleine Codeabschnitte*) näher erläutert. Die gesamte Programmstruktur kann im Anhang (C) eingesehen werden. In Abbildung 86 wird vorab das Resultat des nachfolgend erläuterten Erkennungsprozesses, bei dem ein intelligenter Code identifiziert wird, vorgestellt.

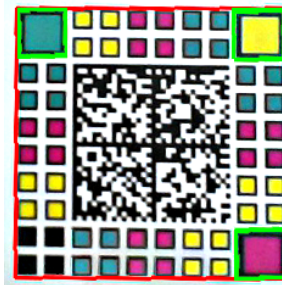


Abbildung 86 Erkennungsprozess des intelligenten Codes

In der Abbildung sind die drei grün umrandeten Suchmuster zu erkennen, dessen äußere Konturen den vollständigen intelligenten Code für eine nachfolgende Codeentnahme umranden (rot). Das für den Erkennungsprozess folgend zu behandelnde Skript wurde mittels der Skriptsprache JavaScript (JS) geschrieben. Die auf einer für das Web reduzierten CV (OpenCV.js) Programmibibliothek basiert. Die Kombination von OpenCV.js, WebRTC und JS ermöglicht es neue Möglichkeiten in der Entwicklung von Webanwendungen, für das Web- Virtual Reality (WebVR) und Augmented Reality (WebAR) zu schaffen. In Abbildung 87 ist die Browserkompatibilität einzusehen<sup>14</sup>, wobei bis auf den Browser Internet Explorer alle übrigen mit der Spezifikation seitens des World Wide Web Consortium (Jennings et al., 2020) übereinstimmen (developer.mozilla, 2020). Spezifischer findet unter anderem eine Übereinkunft mit dem Navigator Interface Extensions statt (ebd., § 9.1), in der das hier verwendete Objekt mediaDevices spezifiziert ist. Das MediaDevices Objekt ist der Einstiegspunkt zur WebRTC-API (*Kapitel 3.8*).

	🖥️						📱					
	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	Android webview	Chrome for Android	Firefox for Android	Opera for Android	Safari on iOS	Samsung Internet
mediaDevices	47	12	36	No	34	11	47	47	36	34	11	5.0
getUserMedia	53 *	12	36 *	No	40 *	11	53	53 *	36 *	41 *	11	6.0

Abbildung 87 Browserkompatibilität der eingesetzten Funktionen (developer.mozilla, 2020)

<sup>14</sup> Die grüne Hinterlegung steht dabei für die Kompatibilität mit dem Browser und rot für die nicht Kompatibilität.

Beginnend wird überprüft, ob die OpenCV Programmbibliothek geladen und verfügbar ist, wobei daraufhin die Funktion *opencvIsReady* ausgeführt wird. Über diese Funktion werden zum einen die *initCameraUI*, die Kamera Benutzeroberfläche, und zum anderen die *initCameraStream*, welche das Videoobjekt für den Videostream ist, initialisiert (Snippet 1):

```

1 function opencvIsReady() {
2     initCameraUI();
3     initCameraStream();}

```

Snippet 1 Prüfung der Verfügbarkeit von OpenCV

Es wird zunächst die Funktion und Aufgabe der Browserkonsole erläutert. Diese protokolliert bspw. Sicherheitsfehler, verschiedene Warnungen sowie Meldungen. Sie beinhaltet jedoch auch Entwicklerwerkzeuge wie Implementationen zur Analyse des jeweiligen Programmierskripts. Auf diese Weise können JavaScript Ausdrücke über die Browserkonsole vorgenommen werden. So kann in diesem Fall die Variable des Streams der Kamera in der Browserkonsole ausgegeben werden. Dazu werden alle Medienzugriffe jeglicher Art zunächst gestoppt (*track.stop*), da verschiedene Einstellungen und Berechtigungen durch den Benutzer gesetzt werden müssen (Snippet 2):

```

1 function initCameraStream() {
2     if (window.stream) {
3         window.stream.getTracks().forEach(function(track) {
4             track.stop();
5         });}

```

Snippet 2 Initialisierung des Streams der Kamera

Darauffolgend werden Einstellungen den Videostream betreffend gesetzt. Die Syntax *navigator.mediaDevices* gibt ein Objekt von Typ *MediaDevices* zurück. Diese Schnittstelle ermöglicht den Zugriff auf angeschlossene Medieneingabegeräte, wie die der Kameras, Mikrofone und auch die gemeinsame Bildschirmnutzung. Somit wird der Zugriff auf die Hardwarequellen von Mediendaten ermöglicht:

```

6 navigator.mediaDevices
7     .getUserMedia(constraints)
8     .then(handleSuccess)
9     .catch(handleError);

```

Snippet 3 Zugriff auf angeschlossene Medieneingabegeräte

Die Funktion *getUserMedia* schaltet mit der Zustimmung (*handleSuccess*) des Benutzers in Form eines *successCallback* über eine Eingabeaufforderung die Kamera und bei Bedarf auch ein Mikrophon ein (Snippet 3) und stellt ein Medienstream (*MediaStream*) Objekt mit einer Schnittstelle für einen Datenstrom der Medieneingabegeräte zur Verfügung, auf die seitens OpenCV, zugegriffen wird (Snippet 4):

```

10 function handleSuccess(stream) {
11     window.stream = stream;
12     video.srcObject = stream;
13     video.play();

```

Snippet 4 Zurverfügungstellung des Videostreams

Der Zugriff auf den Videostream erfolgt dabei, nach Freigabe des Benutzers, über die OpenCV.js Programmbibliothek und die Instanz *video.addEventListener*. Dabei erzeugt der *new* Operator eine weitere Instanz eines benutzerdefinierten Objekttyps, in diesem Falle eine *cv.VideoCapture*, die das Fundament für die Verwertung, der jeweiligen Frames, seitens der Computer Vision Programmbibliothek ermöglicht. Zusätzlich wird *startVideoProcessing* zur Videoverarbeitung initialisiert (Snippet 5):

```

14 video.addEventListener("canplay", function (ev) {
15     if (!streaming) {

```

```

...
16         streaming = true;
17         vc = new cv.VideoCapture(video);
18         startVideoProcessing();}, false);

```

Snippet 5 Zugriff der Computer Vision Programmbibliothek

Die Methode `requestAnimationFrame` teilt dem Browser mit, dass eine Animation ausgeführt werden soll und fordert den Browser auf, die Animation vor dem nächsten Neuzeichnen zu aktualisieren. Dabei wird mit dem Argument `processVideo`, ein Animationsframe angefordert (Snippet 6):

```

20     function startVideoProcessing() {
21         src = new cv.Mat(videoHeight, videoWidth, cv.CV_8UC4);
...
22         requestAnimationFrame(processVideo);}

```

Snippet 6 Anfordern eines Animationsframes

Die Funktion `processVideo` lädt einzelne Frames (Eingabebild) ein und lässt diese durch die Instanz `readCode` analysieren. Hierbei wird das Ergebnis der nachfolgend zu erläuternden Analyse in einem HTML Canvas Element dargestellt und mit einer ID `canvasOutput` versehen. Das Canvas Element zeichnet Grafiken die dem Benutzer angezeigt werden. Dabei zeigt `cv.imshow` das Ausgabebild integriert in den Videostream im dafür festgelegten Fenster, sichtbar für den Benutzer an. Abschließend wird erneut ein `requestAnimationFrame` mit dem Argument `processVideo` angefordert, wobei eine in sich wiederholende Rückruffunktion (Callback) erstellt wird. Die Callback-Funktion ruft sich selbst und somit die einzelnen Frames wiederholend auf, wobei der zu scannende intelligente Code fortlaufend abgesehen wird (Snippet 7):

```

23     function processVideo() {
24         vc.read(src);
25         cv.imshow("canvasOutput", readCode(src));
26         requestAnimationFrame(processVideo);}

```

Snippet 7 Ausgabe des Analyseergebnisses

Die Funktion `readCode` zeichnet, sofern Suchmuster vorhanden, um die Suchmuster eine Kontur (`lastEdgesMarker0 - 2`) und stellt diese auf dem jeweiligen Einzelframe dar. Damit die Suchmuster ohne Performanzprobleme des Prozessors identifiziert werden können, wird nur jedes 15. Frame (`scancounter % 15 != 0`) ohne den Verlust von Informationen entnommen und gezeichnet (Snippet 8):

```

27     function readCode(src) {
28         scancounter++;
29         if (scancounter % 15 != 0) {
30             if (lastEdgesCode != null && lastEdgesMarker0 != null) {
31                 drawEdges(lastEdgesCode, new cv.Scalar(255, 0, 0, 255));
32                 drawEdges(lastEdgesMarker0, new cv.Scalar(0, 255, 0, 255));
33                 drawEdges(lastEdgesMarker1, new cv.Scalar(0, 255, 0, 255));
34                 drawEdges(lastEdgesMarker2, new cv.Scalar(0, 255, 0, 255));]
35         return src;}

```

Snippet 8 Darstellung der Marker und der Umrandung des Codes

Damit das Einzelframe für den nachfolgenden Analyseprozess bereit steht, wird der Einzelframe zunächst, über die Computer Vision Programmbibliothek mit der Funktion `cv.cvtColor` in ein Graustufenbild konvertiert. Dieses wird daraufhin in einem Schwellenwertverfahren (Kapitel 6.1.2) mit der Funktion `cv.threshold` in ein Binärbild umgewandelt. Das Binärbild dient der Funktion `cv.findContours`

als Grundlage für die Erkennung der jeweiligen Konturen. Dabei identifiziert die Funktion die jeweiligen Konturen für eine shape detection (Formerkennung) anhand des Suzuki-Abe Algorithmus (*Kapitel 6.1.6*) (Snippet 9):

```

36 cv.cvtColor(resizedSrc, dstC1, cv.COLOR_RGBA2GRAY);
37 cv.threshold(dstC1, dstC4, 120, 200, cv.THRESH_BINARY);
38 cv.findContours(dstC4, contours, hierarchy, 2, 2, {x: 0, y: 0});

```

Snippet 9 Schwellenwertverfahren und Identifizierung von Konturen

Im Folgenden werden alle Konturen zu einem Array (*sortedContours*) hinzugefügt (Snippet 10):

```

39 let sortedContours = [];
40 for (let i = 0; i < contours.size(); i++) {
41     sortedContours.push(contours.get(i));

```

Snippet 10 Konturen zu einem Array zusammenfassen

In diesem Schritt werden alle Konturbereiche, mittels der Funktion *cv.contourArea*, nach Größe aufsteigend berechnet (Snippet 11):

```

42 sortedContours.sort(function(x, y) {
43     if (cv.contourArea(x) < cv.contourArea(y)) return -1;
44     if (cv.contourArea(x) > cv.contourArea(y)) return 1;
45     return 0; });

```

Snippet 11 Konturen nach Konturfläche sortieren

Hierin werden die acht größten Konturen entnommen. Dazu wurden alle größeren Konturen von groß nach klein sortiert und geprüft, ob es sich bei den Konturen um Quadrate (Suchmuster) handelt. Die Funktion *cv.approxPolyDP* basiert auf dem Douglas-Peucker-Algorithmus (1973) und ist ein Algorithmus zur Kurvenglättung. Dieser approximiert eine Kurve oder ein Polygon, um diese zu vereinfachen. Die letzten Zeilen (53 und 54) haben die Aufgabe, die Konturen auf ihre quadratische Gestalt zu überprüfen. Dabei wird geschaut, ob die Kontur vier Kanten und nahezu identische Breiten und Höhen aufweist, was charakteristisch für eine quadratische Form ist (Snippet 12):

```

46 let biggestContours = sortedContours.slice(sortedContours.length-8, sortedContours.length);
47 for (let i = biggestContours.length-1; i >= 0; i--) {
48     let approx = new cv.Mat();
49     let cnt = biggestContours[i];
50     cv.approxPolyDP(cnt, approx, 0.03 * cv.arcLength(cnt, true), true);
51     let rect = cv.boundingRect(cnt);
52     let ar = rect.width / rect.height;
53     if (approx.rows == 4 && ar >= 0.9 && ar <= 1.1 && rect.width <= 55) {
54         foundMarker.push_back(approx);

```

Snippet 12 Konturentnahme nach Größe

In diesem Abschnitt wird geprüft, ob die drei quadratischen Suchmuster und die Umrandungskontur des intelligenten Codes gefunden wurden (Snippet 13):

```

55 if (foundMarker.size() == 3) {
56     let cnt0 = foundMarker.get(0);
57     let cnt1 = foundMarker.get(1);
58     let cnt2 = foundMarker.get(2);
59     let allMarkerHull = new cv.Mat();

```

Snippet 13 Gefundene Konturen und Umrandungskontur auf drei eingrenzen

Im Folgenden wird, als eine Vorbereitung für den nächstfolgenden Schritt, eine Matrix mit allen gefundenen Koordinaten erstellt (Snippet 14):

```

60 let allMarkerMatrix = cv.matFromArray(12, 2, cv.CV_32S,

```

*Snippet 14 Matrix mit allen gefundenen Koordinaten*

Die zuvor erstellte Matrix wird für die mit der Funktion `cv.convexHull`, zu erzeugende konvexe Hülle (Kapitel 6.1.8) verwendet. Mit ihr kann aus allen aufgelisteten Konturen eine Außenkontur erstellt werden. Die Funktion `cv.minAreaRect` gibt dabei eine sogenannte Box2D-Struktur zurück, die Angaben wie obere linke Ecke (x, y), Breite, Höhe und Drehwinkel beinhaltet. Hierbei ist unter anderem ist der Drehwinkel von Relevanz, um ein Rechteck samt Rotation zu zeichnen. Dies erfolgt über von OpenCV noch nicht aufgelistete Unterfunktion `cv.boxPoints` (OpenCV, 2020a) (Snippet 15):

```
61 cv.convexHull(allMarkerMatrix, allMarkerHull, false, true);
62 let minAreaHull = cv.minAreaRect(allMarkerHull);
```

*Snippet 15 Eine konvexe Hülle um alle Konturen ziehen*

In diesem Abschnitt wird der intelligente Code aus dem Einzelframe entnommen und ausgerichtet. Hierzu fließen die zuvor ermittelten Koordinaten der Ecken der konvexen Hülle in die Funktion `cv.getPerspectiveTransform` ein. Diese berechnet eine perspektivische Transformation (Kapitel 6.1.9) aus den vier Punkten und die Funktion `cv.warpPerspective` wendet diese perspektivische Transformation auf alle Pixel des Eingabebildes an. Die Funktion `cv.imshow` mit der ID `foundCode` dient dazu, den identifizierten intelligenten Code zu speichern (Snippet 16):

```
63 let perspectiveTransform = cv.getPerspectiveTransform(srcMatrix, dstMatrix);
64 cv.warpPerspective(src, finalCode, perspectiveTransform, dsize, cv.INTER_LINEAR, ...);
65 cv.imshow('foundCode', finalCode);
```

*Snippet 16 Perspektivische Transformation*

Der identifizierte und zugeschnittene intelligente Code (`finalCode`) wird für eine nachfolgende Analyse an den Server, genauer an die Auswerteeinheit, transferiert (Snippet 20):

```
66 sendCodeToServer();
```

*Snippet 17 Codetransferierung, zur Analyse an den Server*

## 6.4 Separierung des intelligenten Codes

Der in *Kapitel 6.3* identifizierte und entnommene intelligente Code wird nachfolgend durch eine Separierung in seine Bestandteile geteilt. Für diesen Vorgang basiert dabei das folgend zu behandelnde Skript auf der höheren Programmiersprache Python. In diese werden verschiedene Programmbibliotheken wie OpenCV, Numpy und Pylibdmtx eingebunden. Die gesamte Programmstruktur ist im Anhang (D) zu finden. Aufgrund der Komplexität der gesamten Programmstruktur werden auch hier die wichtigsten Bausteine der Programmierung in Form von Snippets vorgestellt und näher erläutert. So wird auch in diesem Abschnitt (Abbildung 88) das Resultat der nachfolgenden Separierung, bei dem der intelligente Code in seine Bestandteile geteilt wird, vorgestellt.

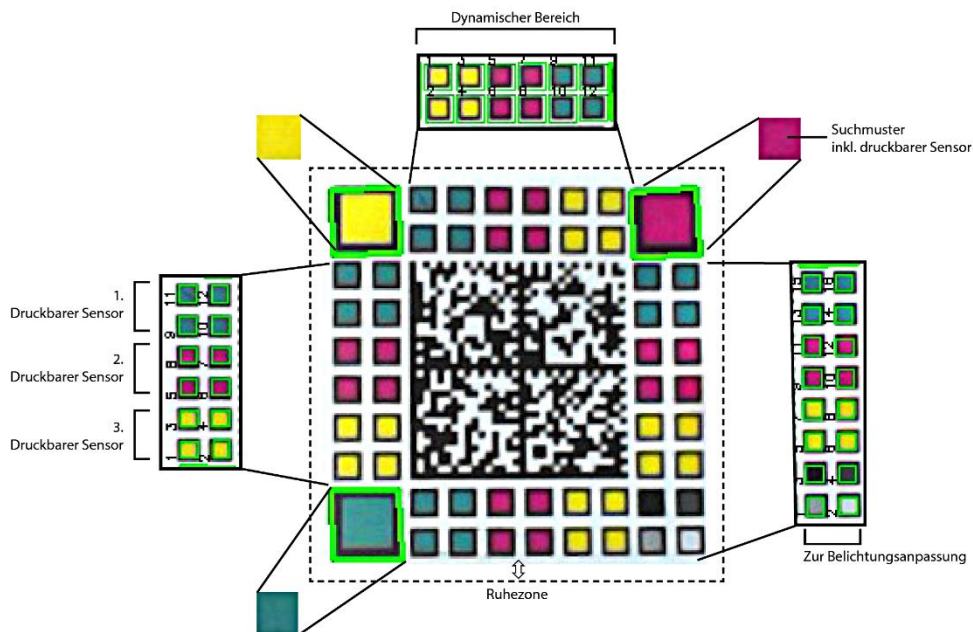


Abbildung 88 Bestandteile des intelligenten Codes

In der Abbildung sind die separierten Bestandteile des Codes hervorgehoben. Im Einzelnen sind dies die dynamischen Bereiche, der statische DataMatrix Code und die Suchmuster. Die jeweiligen Bestandteile des Codes werden ebenfalls mittels einer Konturerkennung identifiziert. Dabei werden die jeweiligen Farbinformationen aus den einzelnen quadratischen Feldern entnommen (in der Abbildung nummeriert zu erkennen). Sobald das Verfahren abgeschlossen ist, werden alle Daten in einem dafür vorgesehenen Datenbankeintrag hinterlegt.

Um die Auswertung durchzuführen, wird der Ausschnitt des intelligenten Codes zunächst vom Server abgerufen und mit der Funktion `cv.imread` geladen (Snippet 18):

```
1 cv.imread(code.jpg')
```

Snippet 18 Laden des intelligenten Codes

In diesem Abschnitt werden die Algorithmen aus dem *Kapitel 6.6* eingebunden, auf die an dieser Stelle nicht weiter eingegangen wird. Der im Kontrast und Helligkeit angepasste, farblich korrigierte und bei Bedarf mittels Colormanagement farblich angeglichenen intelligenten Code wird im nächsten Schritt zunächst in ein Graustufenbild konvertiert und über ein dynamisches Schwellenwertverfahren (*Kapitel 6.1.2*) in ein Binärbild umgewandelt. Das Binärbild dient als Grundlage für die mit der Funktion

`cv.findContours` identifizierten Konturen. Auf die Koordinaten des vorherigen Erkennungsprozesses (*Kapitel 6.3*) wird aufgrund präziserer Identifizierungsverfahren verzichtet, da eventuell auftretende Abweichungen zu Problemen bei der Separierung des intelligenten Codes führen können. Die identifizierten Konturen werden nachfolgend mit der Funktion `cv.contourArea` nach Konturgröße absteigend sortiert (Snippet 19):

2	<code>cv.cvtColor(code, cv.COLOR_BGR2GRAY)</code>
3	<code>cv.threshold(code, 100, 200, cv.THRESH_BINARY)</code>
4	<code>cv.findContours(code, cv.RETR_TREE, cv.CHAIN_APPROX_SIMPLE)</code>
5	<code>sorted(konturen, key=cv.contourArea, reverse=True)[1:5]</code>

Snippet 19 Schwellenwertverfahren und identifizieren von Konturen

Daraufhin werden die sortierten Konturen zunächst über die Funktion `cv.approxPolyDP` approximiert und soweit es möglich ist genähert. Die Funktion `cv.minAreaRect` gibt eine sogenannte Box2D-Struktur zurück, die Angaben über die Koordinaten (x, y), Breite, Höhe und Drehwinkel, beinhaltet. Die Funktion `cv.boxPoints` findet dazu die vier Eckpunkte eines gedrehten Rechtecks. Somit ist die Funktion nützlich, um gedrehte Suchmuster aufzufinden und die Koordinaten für den darauffolgenden Schritt der Ausrichtung und Segmentierung des intelligenten Codes zu verwenden (Snippet 20):

6	<code>cv.approxPolyDP(knt, 0.03 * cv.arcLength(knt, True), True)</code>
7	<code>(x, y, w, h) = cv.boundingRect(approx)</code>
8	<code>cv.minAreaRect(approx)</code>
9	<code>cv.boxPoints(rect)</code>
10	<code>np.int0(box)</code>

Snippet 20 Formerkennung und Suchmuster Identifizierung

Die zuvor identifizierten Suchmuster (*marker1–3*) werden folgend Variablen zugeordnet (Snippet 21):

11	<code>marker1 = knt [0]</code>
12	<code>marker2 = knt [1]</code>
13	<code>marker3 = knt [2]</code>

Snippet 21 Zuordnung der entnommenen Konturen

Anhand der Suchmuster und ihrer Koordinaten wird der intelligente Code mit seinen Suchmustern über die Funktion `cv.getRotationMatrix2D` und `cv.warpAffine` in eine bestimmte Ausgangsposition gedreht (*Kapitel 6.1.9*). Da der intelligente Code vom Benutzer aus unbekannter Perspektive erfasst wurde, muss der Code im Hinblick auf eine Vereinfachung für den in diesem Abschnitt behandelten Algorithmus, auf eine definierte Position (*Kapitel 6.1.9*) transformiert werden (Snippet 22):

14	<code>cv.getRotationMatrix2D(mitte, grad, faktor)</code>
15	<code>cv.warpAffine(code.copy(), Mom, (h, b))</code>

Snippet 22 Ausrichten des intelligenten Codes

Entsprechend der Koordinaten der jeweiligen Suchmuster, können mit der folgenden Syntax, die RGB-Farbwerte der einzelnen Suchmuster aus dem intelligenten Code entnommen werden. So werden die Suchmuster ebenfalls jeweils als Bild abgespeichert und auf dem Server hinterlegt (Snippet 23):

16	<code>marker1 = code[ y : y + höhe, x : x + breite]</code>
17	<code>marker2 = code[ y : y + höhe, x : x + breite]</code>
18	<code>marker3 = code[ y : y + höhe, x : x + breite]</code>

Snippet 23 RGB-Farbwerte aus den Suchmustern entnehmen

Die Koordinaten der drei identifizierten Suchmuster dienen der Separation der Einzelbestandteile des intelligenten Codes, die sich aus vier dynamischen Bereichen (*smartbereichA – D*) und einem statischen DataMatrix Code (*DataMatrix*) zusammensetzen. Diese Einzelbestandteile werden ebenfalls, vor ihrer Auswertung zunächst zur Sicherung als Bilder auf dem Server gespeichert. Der statische DataMatrix



Code wird dabei gleichzeitig zum Dekodieren der *Pylibdmtx* Programm-bibliothek übergeben. Das Ergebnis wird ebenfalls in der Datenbank hinterlegt. Im drauffolgenden Schritt der Konturerkennung, werden die jeweiligen Farbinformationen aus den einzelnen quadratischen Feldern der dynamischen Bereiche entnommen:

```

19 smartbereichA = code[ y : y + höhe, x : x + breite]
20 smartbereichB = code[ y : y + höhe, x : x + breite]
21 smartbereichC = code[ y : y + höhe, x : x + breite]
22 smartbereichD = code[ y : y + höhe, x : x + breite]
23 DataMatrix = code[ y : y + höhe, x : x + breite]

```

*Snippet 24 Separation der Bestandteile des intelligenten Codes*

Die vier dynamischen Bereiche werden anschließend einer Konturerkennung unterzogen (Snippet 19) und über eine entsprechende Sortierfunktion (Zeile 27) absteigend geordnet. Die Ordnung der einzelnen quadratischen Felder (*smartdots*) dient der im drauffolgenden Schritt notwendigen RGB-Farbwertentnahme (Snippet 25):

```

24 smartdots = sorted(smarddots, key=cv.contourArea, reverse=True)[1:13]
25 boundingBoxes_smartdots = [cv.boundingRect(c) for c in smartdots]
26 (sort_smartdots, boundingBoxes_smartdots) = zip(
27     *sorted(zip(smarddots, boundingBoxes_smartdots), key=lambda b: [b[1], b[0]]))

```

*Snippet 25 Einzelnen quadratischen Felder werden sortiert*

In diesem Abschnitt werden die einzelnen quadratischen Felder des dynamischen Bereichs (*smartbereichA*) innerhalb einer sogenannten for-Schleife ausgegeben. Die 12 quadratischen Felder werden dabei jeweils einzeln als Bild gespeichert, auf dem Server hinterlegt und mit ihren jeweiligen RGB-Farbwerten in der Datenbank eingetragen (Snippet 26):

```

28 sdt_count = 0
29 for sdt in sort_smartdots:
30     approx = cv.approxPolyDP(sdt, 0.02 * cv.arcLength(sdt, True), True)
31     (x, y, w, h) = cv.boundingRect(approx)
32     cv.rectangle(smardbereichA, (x, y), (x + w, y + h), (0, 255, 0), 1)
33     smardbereichA [y + h // 2: y + h // 2, x + w // 2: x + w // 2]

```

*Snippet 26 Farbwertentnahme der quadratischen Felder eines exemplarischen dynamischen Bereichs*

Entsprechend werden nach jedem Scan die jeweiligen Farbinformationen der Sensoren zusammen mit einem Zeitstempel versehen gesichert. Bei einer Historienbetrachtung werden die einzeln extrahierten Farbwerte mittels einer Auswerteeinheit gegenübergestellt und Informationen über die zeitliche Entwicklung der jeweiligen Sensordaten ausgegeben.

## 6.5 Auswerteeinheit

Das vorliegende Kapitel setzt sich mit der Berechnung und Auswertung der Zustände der Sensordaten auseinander, die in voneinander variierenden Zeitabständen erfasst wurden. Die RGB-Farbwerte werden als hexadezimale Werte in der Datenbank hinterlegt, da sie so einfacher zu verarbeiten sind. Im nächsten Schritt wird die zeitliche Entwicklung der Sensorinformationen in Form von Farbabstandswerten aufgezeigt. Die gesamte Programmstruktur kann im Anhang (E) eingesehen werden. Aufgrund der Komplexität der gesamten Programmstruktur werden auch hier lediglich die relevanten Funktionen der Auswerteeinheit, in Form von Snippets, vorgestellt und näher erläutert.

Beginnend wird die Datenbankanbindung vorgestellt. Anhand der Funktion `MySQLdb.connect` wird eine Verbindung zur MySQL Datenbank aufgebaut (Snippet 27):

1	<code>db = MySQLdb.connect(host="localhost", user="xxx", passwd="xxx", db="xxx")</code>
2	<code>db.autocommit(True)</code>
3	<code>cur = db.cursor(MySQLdb.cursors.DictCursor)</code>
4	<code>sql_get_latest_difference = "select * from codereader_scandifference order by id desc limit 1;"</code>
5	<code>cur.execute(sql_get_latest_difference)</code>
6	<code>sql_get_scans = "select * from codereader_scan order by datetime desc limit 2;"</code>
7	<code>cur.execute(sql_get_scans)</code>

Snippet 27 Datenbankanknüpfung herstellen

Die hexadezimalen Werte der Sensorinformationen werden in einem ersten Schritt über die Funktion `hex2rgb` in RGB Farbwerte umgewandelt und über die Funktion `color.rgb2lab` in den  $L^*a^*b^*$  Farbraum transformiert. Die entsprechenden Grundlagen der Farbvalenzen und Farbabstände können hierzu aus dem Kapitel 4.1 entnommen werden. So ist im Folgenden zur Veranschaulichung lediglich ein Ausschnitt der Farbkonvertierung zu erkennen (Snippet 28):

8	<code>color.rgb2lab(hex2rgb(scan1[quadratisches Feld]))</code>
---	--

Snippet 28 Konvertierung Hex zu RGB zu  $L^*a^*b^*$

Die drei sensorischen Farben (Kapitel 4) zeigen jeweils eine reaktionsbedingte Farbveränderung an. Gemeinsamkeiten weisen alle Sensoren beim  $L^*$ -Wert auf. Dieser beschreibt die Helligkeit und steht als Kenngröße für den Entfärbungsprozess des hydrochromen Sensors sowie die Farbveränderung der halochromen und photochromen Sensoren. Entsprechend werden die  $\Delta L^*$ ,  $\Delta a^*$  und  $\Delta b^*$  Werte zwischen zwei Messzeitpunkten berechnet – dies wird als `delta_marker1` bezeichnet – und gegenübergestellt (Snippet 29):

9	<code>delta_marker1 = (scan2_marker1_lab[0] - scan1_marker1_lab[0],</code>
10	<code>scan2_marker1_lab[1] - scan1_marker1_lab[1],</code>
11	<code>scan2_marker1_lab[2] - scan1_marker1_lab[2],</code>

Snippet 29 Berechnung der Abweichungen von  $\Delta L^*$ ,  $\Delta a^*$  und  $\Delta b^*$

Anschließend werden die euklidischen Farbabstände  $\Delta E_{76}$ , mit der Funktion `color.deltaE_cie76`, und  $\Delta E_{00}$ , mit der Funktion `color.deltaE_ciede2000`, ebenfalls zwischen zwei Messzeitpunkten berechnet (Snippet 30):

12	<code>color.deltaE_ciede2000(scan2_marker1_lab, scan1_marker1_lab),</code>
13	<code>color.deltaE_cie76(scan2_marker1_lab, scan1_marker1_lab))</code>

Snippet 30 Berechnung der euklidischen Farbvalenzabstände

Die Ergebnisse werden über die Funktion `cur.execute` in die Datenbank eingetragen und die Verbindung zur Datenbank mit der Funktion `db.close` beendet (Snippet 31).

14	<code>cur.execute(sql_insert)</code>
15	<code>db.close()</code>

Snippet 31 Eintragen der Ergebnisse in die Datenbank und beenden der Datenbank-Verbindung

Damit können zum einen auf Basis der Farbabstandswerte zwei aufeinanderfolgende Messzeitpunkte gegenübergestellt werden. Zum anderen können bspw. die  $L^*$ -Farbwerte der jeweiligen Farbveränderungen der Sensoren zum Eingrenzen des Datensatz herangezogen werden. Anhand des Beispiels der am photochromen Sensor (Abbildung 57) ermittelten  $L^*a^*b^*$  Farbwerte (Tabelle 22) können die folgenden Eingrenzungen des Datensatzes veranschaulicht werden:

### I. Ober- und Untergrenze

Die stufen- oder verlaufweisen Farbveränderungen und ihre zugehörigen Farbwerte können mit geeigneten Schwellenwerten eingegrenzt werden. Vergleichbar mit einem sensorischen Helligkeitsschalter wechselt hier ab einem vorgegebenen Farbwert das programmierte System der Farbauswertung vom zuvor inaktiven in den aktiven Bereich und signalisiert eine Überschreitung.

### II. Ampelsystem

Ein Ampelsystem wäre eine weitere Möglichkeit der Eingrenzung eines Datensatzes. Dabei ist der Datensatz in drei, nicht zwingend gleiche Teile geteilt. Hierdurch entstehen zwei Schwellenwerte, die jeweils eine Überschreitung anzeigen. Eine solche Teilung bedarf jeweils einer anwendungsspezifischen Interpretation (z. B. gut, mittel, schlecht; Warenklasse A, B oder C).

### III. Quartile und Quintile

Bei einem Quartil wird der Datensatz in vier Teile gleicher Größe aufgeteilt. Daraus resultierend können stufenweise Warnungen oder Hinweise anhand der Farbveränderung ausgegeben werden. Jedoch eignen sich auch Quintile, die eine fünffache Eingrenzung aufweisen und damit eine noch kleinschrittigere Aufteilung ermöglichen.

Entsprechend der zuvor vorgestellten Varianten (I – III) lassen sich die Datensätze eingrenzen. Eine solche Abstufung lässt sich solange und so tief detaillieren, wie ein gedruckter Sensor, unter Einbeziehung der gebotenen Toleranzen, diskrete Farbabstände anzeigt.

Die von der Auswerteeinheit ermittelten Daten werden nachfolgend für eine Eingrenzung des Datensatzes herangezogen, in dem die im *Kapitel 6.5* beschriebene Methode des Ampelsystems zur Anwendung kommt. In *Abbildung 89* ist der zu gleichen Anteilen (33,33) dreigeteilte L\*-Wert (100) mittels Signalfarben visualisiert. Dazu wird entsprechend einer Überschreitung des jeweilig festgelegten Schwellenwerts (T) die Hintergrundfarbe grün ( $T < 33,33$ ), gelb ( $T \geq 33,33$  bis  $T \leq 66,66$ ) oder rot ( $T > 66,66$ ) gefärbt. Somit kann der mittlere Bereich (gelb) als grenzwertiger Übergangsbereich gekennzeichnet werden, in dem bspw. ein Produkt, wie ein Nahrungsmittel, noch unter kontrollierten Voraussetzungen als verzehrbar angezeigt wird.






Marker1 - Color (Hex / Lab)	 [46.759, -21.826, 18.468]	Marker1 - Color (Hex / Lab)	 [16.084, -8.143, -12.204]
Marker2 - Color (Hex / Lab)	 [63.323, -5.827, 55.367]	Marker2 - Color (Hex / Lab)	 [52.459, 18.931, 24.568]
Marker3 - Color (Hex / Lab)	 [67.273, -36.22, -12.435]	Marker3 - Color (Hex / Lab)	 [77.649, -7.481, 1.418]

Abbildung 89 Ampelsystem fürs Eingrenzen des Datensatzes

## 6.6 Beseitigung von Störparametern

In diesem Kapitel werden unterschiedliche Algorithmen vorgestellt, die der Kompensation und Beseitigung verschiedener unerwünschter Störeinflüsse dienen. In einem Vorverarbeitungsprozess werden die Sensorinformationen um die Störeinflüsse bereinigt, um anschließend ausgewertet werden zu können. Störfaktoren wie Farbabweichungen durch Umgebungslicht während einer Kameraaufnahme, ungeeignete Belichtungseinstellungen, Helligkeits- und Kontrastschwankungen, Farbigkeit des Substrats etc. können die Farbinformationen der sensorischen Farben überlagern, beeinflussen und verfälschen. Das *Kapitel 3.7.1* hebt die methodische Herangehensweise der Beseitigung geräteabhängiger Einflussfaktoren der Smart Device Kameras nach einer Aufnahme durch eine nachfolgende Verarbeitung durch das Operating System (OS) hervor.

### 6.6.1 Farbanpassung

Im Folgenden wird das Konzept eines sogenannten Umgebungslicht-Assistenten (Surround-Light-Assistent) zur Farbanpassung vorgestellt. Seine Funktion ist es, die vorherrschende Lichtsituation der auszulesenden Sensoren entsprechend der Normlichtart D65 anzupassen. Dies hat den Hintergrund, da die in den Smart Devices implementierten Kamerafarprofile vorzugsweise die übliche anwendungsorientierte Beleuchtungssituation berücksichtigen, bspw. im Fotostudio. Sobald sich das Umgebungslicht dynamisch verändert und damit eine andere Lichtbedingung vorherrscht, muss in der Regel ein neues Farbprofil erstellt werden. Um eine solche Situation zu vermeiden, wird der Surround-Light-Assistent eingesetzt. Das entsprechende programmiertechnische Script für den Surround-Light-Assistenten kann im Anhang (F) eingesehen werden. Nachfolgend wird die Algorithmus-Struktur des Surround-Light-Assistenten vorgestellt (Abbildung 90).

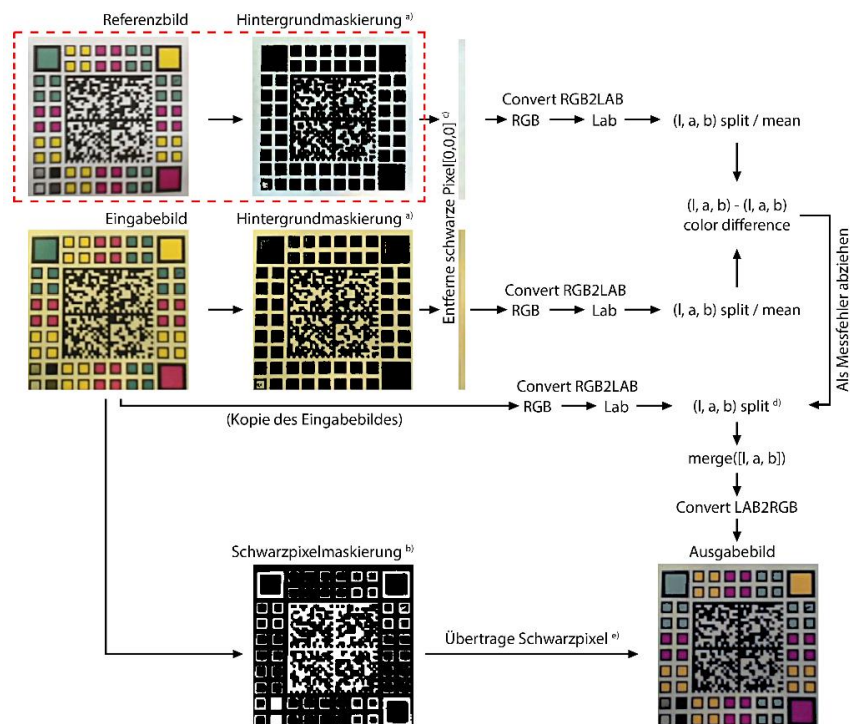


Abbildung 90 Funktionsweise des Umgebungslicht-Assistenten

Dieser Algorithmus berechnet die Farbwerte eines Eingabebildes mittels Abgleich mit den Farbwerten eines Referenzbildes und transformiert diese in ein Ausgabebild. Das Ergebnis ist ein Ausgabebild, das frei von der Farbtemperatur der während der Aufnahme vorherrschenden Lichtsituation ist.

Innerhalb des Algorithmus findet sich ein Referenzbild eines intelligenten Codes, das unter einer D65 Normlichtart aufgenommen (*Kapitel 3.5*) und in Form einer Datei auf dem Server hinterlegt wurde. Die Datei (Snippet 32) wird durch den Algorithmus für die Schritte der Farbtransformation benötigt. Dies wird anhand des Programmierabschnittes der folgenden Algorithmus-Struktur deutlich:

```
1 import numpy as np
2 d65 = np.load('d65.npy')
```

*Snippet 32 D65 Referenzdatei*

Das methodische Vorgehen des Colormanagementprozesses wurde in *Kapitel 3.7.5* beschrieben. Dabei werden mittels einer Pixeltransformation die Pixelwerte des Papierweiß aus dem Eingabebild entnommen und unabhängig von den Nachbarpixeln durch separat in einer Datei hinterlegte Referenzpixelwerte korrigiert (Pixeltransformation).

Die Programmierschritte werden im Folgenden detailliert dargestellt. Im folgenden Szenario wurde der intelligente Code seitens eines Benutzers unter einem für den späteren Analyseprozess ungeeigneten Glühbirnenlicht (A) aufgenommen (*Kapitel 3.5*). So muss das gelblich wirkende Eingabebild farblich korrigiert werden. Dazu wird in einem ersten Schritt vom Eingabebild als Sicherung eine Kopie erstellt und der Hintergrund des originalen Eingabebildes für eine anschließende Pixelentnahme über ein Schwellenwertverfahren (*Kapitel 6.1.2*) maskiert (Snippet 33):

```
3 lower_white = np.array([100, 100, 100], dtype = np.uint8)
4 upper_white = np.array([255, 255, 255], dtype = np.uint8)
```

*Snippet 33 Schwellenwert Hintergrundpixel*

Der in der Abbildung rot gestrichelte Bereich samt Programmierablauf, ist durch das Laden der D65 Referenzdatei nicht mehr notwendig, da die Referenzdatei bereits zuvor unter Anwendung einer Hintergrundmaskierung (Abbildung 90, a) erzeugt wurde, mit dem Ziel, Verarbeitungszeit einzusparen. Im nächsten Schritt war es notwendig, alle Schwarzpixel aus dem Eingabebild über ein Schwellenwertverfahren zu maskieren (Snippet 34), da diese in einem späteren Schritt für eine Kontrasterhöhung erneut dem Ausgabebild hinzugerechnet werden:

```
5 lower_black = np.array([0, 0, 0], dtype = np.uint8)
6 upper_black = np.array([90, 90, 90], dtype = np.uint8)
```

*Snippet 34 Schwellenwert Schwarzpixel*

Als nächstes werden alle Schwarzpixel aus den maskierten Bereichen (Abbildung 90, c) des Eingabebildes entfernt, um resultierend nur die im Hintergrund enthaltenen Farbpixel zu erhalten (Snippet 35):

```
7 np.delete(masked, np.argwhere(masked == [0,0,0]), axis=0)
```

*Snippet 35 Schwarzpixel werden entfernt*

Für eine gezielte Pixeltransformation war es notwendig, die Farbpixel aus dem Hintergrund der Referenzdatei, als auch die des Eingabebildes und ihrer Kopie vom RGB-Farbraum in den L\*a\*b\* Farbraum zu konvertieren (Snippet 36):

```
8 cv.cvtColor(Datei, cv.COLOR_RGB2LAB)
```

*Snippet 36 Farbkonvertierung*

Alle drei Hintergrundfarbpixel Datensätze (Ein- und Ausgabebild, Kopie) wurden in einem nächsten Schritt in einzelne L\*, a\*, b\*-Farbkanäle separiert (Abbildung 90, d), um gezielt auf die Farbkanäle zuzugreifen (Snippet 37):

```
9 (l, a, b) = cv.split(Datei)
```

*Snippet 37 Farbkanal Separation*

Aus den einzelnen Farbkanälen der beiden jeweiligen Farbpixel (ohne Kopie des Eingabebildes) wurde ein arithmetisches Mittel berechnet (Snippet 38):

```
10 (l, a, b), np.mean()
```

*Snippet 38 Arithmetisches Mittel*

Die arithmetischen Mittel der jeweiligen L\*, a\*, b\*-Farbkanäle der beiden Farbpixel wurden dabei in einem nächsten Schritt subtrahiert und die Farbdifferenz in Form eines Messfehlers für jeden L\*-, a\*-, b\*-Farbkanal des zuvor kopierten Eingabebildes abgezogen (subtrahiert). Anschließend wurden die einzelnen L\*, a\*, b\*-Farbkanäle wieder zusammengeführt (Snippet 39):

```
11 cv.merge([l, a, b])
```

*Snippet 39 Zusammenfügen der separierten Farbkanäle*

und anschließend erneut in den RGB-Farbraum konvertiert. Zuletzt wurden die Schwarzpixel, die zuvor in (Abbildung 90, b) (weiß) maskiert wurden, dem kopierten Eingabebild für eine Kontrasterhöhung übertragen (Abbildung 90, e) (Snippet 40):

```
12 transfer[np.where(mask_img == 255)] = img[np.where(mask_img == [255])] = [0]
```

*Snippet 40 Schwarzpixelübertragung*

Das Ziel, die vorherrschende Lichtsituation entsprechend der Normlichtart D65 anzupassen, wurde erreicht. Das unter einem Glühbirnenlicht (A) aufgenommene Bild eines intelligenten Codes wurde in eine D65-Normlichtart überführt und anschließend als Ausgabebild exportiert.

## 6.6.2 Helligkeits- und Kontrastanpassung

Entsprechend der im *Kapitel 3.7.4* festgelegten methodischen Herangehensweise, wird die programmtechnische Umsetzung des Algorithmus zur Helligkeits- und Kontrastanpassung durch eine Bildtransformation, genauer einem Punktoperator (Pixeltransformationen), vorgestellt.

Um die Helligkeits- und Kontrastanpassung bei der Leseapplikation automatisiert auszuführen, wird ein Histogramm<sup>15</sup> ausgehend von einem oder mehreren Arrays, in diesem Fall aus einem Eingabebild berechnet. Histogramme können sinnvoll dazu eingesetzt werden, um die Eigenschaften eines Bildes, im Zusammenhang mit der Helligkeit, dem Kontrast, der Dynamik sowie der Über- und Unterbelichtung, zu ermitteln. Das Histogramm wird anschließend, um eine Helligkeitsveränderung zu bewirken, zu 5% beidseitig beschnitten (clip). In diesem Abschnitt werden, die Konstanten der Verstärkungsparameter ( $\alpha$ ) und Verzerrungsparameter ( $\beta$ ), welche jeweils den Kontrast und die Helligkeit steuern, ermittelt, woraus resultierend eine Helligkeits- und Kontrastanpassung des Eingabebildes erfolgt.

Im Folgenden wird die programmtechnische Umsetzung vorgestellt. Beginnend wird ein Perzentil für den Anteil einer beidseitigen Histogrammbeschneidung (Verteilung) definiert (Snippet 41):

```
1 clip_prozent = 5
```

*Snippet 41 Prozentuale Definition für die Beschneidung*

Daraufhin wird das Eingabebild in ein Graustufenbild konvertiert, um einen Graustufenkanal (mit 256 Intensitätsabstufungen) für die Erzeugung eines Histogramms vorzubereiten (Snippet 42):

```
2 cv.cvtColor(img, cv.COLOR_BGR2GRAY)
```

*Snippet 42 Graustufenkonvertierung*

Anschließend wird ausgehend vom Eingabebild ein Histogramm (*cv.calcHist*) und zusätzlich die Größe (*hist\_size*) berechnet (Snippet 43):

```
3 hist = cv.calcHist([grau],[0],None,[256],[0,256])
4 hist_size = len(hist)
```

*Snippet 43 Histogramm Berechnung*

Als nächstes wird die kumulative Summe des Histogramms berechnet (Snippet 44), um zu bestimmen, wo die Farbhäufigkeit unter einem bestimmten Schwellenwert liegt:

```
5 cumsum = np.cumsum(hist)
```

*Snippet 44 kumulative Summe des Histogramms*

Es folgt die Lokalisierung der anschließend zu beschneidenden Punkte (Snippet 45):

```
6 maximum = cumsum[-1]
7 clip_prozent *= (maximum/100.0)
```

*Snippet 45 Beschneidungspunkte lokalisieren*

Abschließend erfolgt die Lokalisierung der linksseitigen Beschneidung (Snippet 46):

```
8 minVal = 0
9 while cumsum[minVal] < clip_prozent:
10 minVal += 1
```

*Snippet 46 linksseitige Beschneidung*

<sup>15</sup> Dient der Darstellung eines Bildes in Form einer Grafik, mit einer Anzahl von Intensitätswerten im Bereich von 0 – 255.

Daraufhin fällt die Lokalisierung der rechtsseitigen Beschneidung an (Snippet 47):

11	<code>maxVal = hist_size - 1</code>
12	<code>while cumsum[maxVal] &gt;= (maximum - clip_prozent):</code>
13	<code>maxVal -= 1</code>

*Snippet 47 rechtsseitige Beschneidung*

Anschließend wird der Verstärkungsparameter ( $\alpha$ ) berechnet (Snippet 48). Hierzu wird nach dem Beschneiden der minimale und maximale Intensitätsbereich entnommen und durch den Ausgabebereich von 255 geteilt:

14	<code>alpha = 255.0/(maxVal - minVal)</code>
----	--

*Snippet 48 Verstärkungsparameter ( $\alpha$ ) berechnen*

Anschließend wird der Verzerrungsparameter ( $\beta$ ) berechnet (Snippet 49):

15	<code>beta = -minVal * alpha</code>
----	-------------------------------------

*Snippet 49 Verzerrungsparameter ( $\beta$ ) berechnen*

Beide berechneten Parameter  $\alpha$  und  $\beta$  fließen als Skalierungsfaktoren in die folgende Funktion ein, welche das Eingabebild (*img*) skaliert und dazu die absoluten Werte berechnet. Resultierend wandelt die Funktion die berechneten Werte in 8-Bit, in Form eines Ausgabebildes  $f'(f(i, j))$ , um (Snippet 50):

16	<code>cv.convertScaleAbs(img, alpha=alpha, beta=beta)</code>
----	--

*Snippet 50 ConvertScaleAbs*

### 6.6.3 Geräteerkennung

Für den nachfolgenden Colormanagement Prozess werden ICC-Profile (International Color Consortium) der gängigsten Smartphones bereitgestellt, um die Mindestanforderungen an eine geeignete Kamera für die zu erzielende Codeerkennung und Codeanalyse zu erfüllen. Die im Methodenteil (*Kapitel 3.7.3*) aufgelisteten Smartphones dienen der zusätzlichen Verbesserung der Leseapplikation. In diesem Kapitel soll eine Geräteerkennung vorgestellt werden, die nahezu automatisiert die in der Liste verfügbaren Smartphones erkennt und das dazugehörige ICC-Profil für den anschließend zu behandelnden Colormanagement Prozess zur Verfügung stellt. Das Browser Object Model (BOM) ermöglicht es JavaScript mit dem clientseitigen Browser zu kommunizieren und somit nähere Informationen über das Endgerät zu erhalten (Jennings et al., 2020, §9.2). Entsprechend dieser Methode können mögliche Kriterien entwickelt werden, um das passende Endgerät und im aktuellen Fall das Smartphone mit einer hohen Wahrscheinlichkeit einzugrenzen. Anschließend kann der Benutzer aus einer möglichen Liste (Tabelle 12) an Smartphones seines auswählen. Somit kann das dazugehörige ICC-Profil aufgerufen und zielgerichtet eingesetzt werden. Zunächst werden die Codeabschnitte, die zugehörigen Kriterien und darunter die Ausgabe der jeweiligen Funktionen vorgestellt:



### I. Fenstergröße

Um die Darstellung der Webapplikation dynamisch im Sinne eines responsive Interfacedesign an das jeweilige Endgerät anzupassen, wird die innere Höhe und Breite des Browserfensters (ohne Symboleisten / Scrollbars) in Pixel bestimmt. Hierzu werden zwei Funktionen eingesetzt (Snippet 51):

1	window.innerHeight
2	window.innerWidth

*Snippet 51 Fenstergröße*

Ausgabe: (1) 962, (2) 932.

### II. Bildschirmauflösung

Bei der Funktion der Bildschirmauflösung werden unter anderem Browserinformationen über die Bildschirmhöhe und -breite sowie die Farbtiefe bereitgestellt (Snippet 52):

3	screen.width
4	screen.height
5	screen.colorDepth

*Snippet 52 Bildschirmauflösung*

Ausgabe: (3) 1920 Pixel, (4) 1080 Pixel (5), 24-bit.

### III. Browseranwendung

Die Funktion der Browseranwendung gibt unter anderem den Anwendungsnamen, Browser-Codenamen, die sogenannte Browser-Engine, die Version des Browsers und den vom Browser an den Server gesendeten User-Agent-Header zurück (Snippet 53):

6	navigator.appName
7	navigator.appCodeName
8	navigator.product
9	navigator.appVersion
10	navigator.userAgent

*Snippet 53 Browseranwendung*

Ausgabe: (6) Netscape, (7) Mozilla, (8) Gecko, (9) 5.0 (Windows), (10) Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0.

Anhand der vorgestellten Kriterien können Smartphones von anderen Geräten abgegrenzt werden. So ist eine Differenzierung zwischen Smartphones, Tablets, Laptops und Desktop-Computern bereits über die Bildschirmauflösung möglich (Snippet 52, Zeile 1 und 2). Eine zusätzliche Einteilung der Endgeräte kann bspw. über das Betriebssystem iOS und Android erfolgen (Snippet 53, Zeile 9). Endgültig muss jedoch der Benutzer, sofern vorhanden, aus einer möglichen Auswahl sein Smartphone auswählen.

## 6.6.4 Colormanagement

Die Farbanpassung der Smartphone Kameras und ihr Workflowprozess mittels eines Colormanagementsystems (CMS) wird in diesem Abschnitt behandelt. Das methodische Vorgehen, vor allem im Zusammenhang mit der eingesetzten Profile-to-Profile Transformation, kann aus dem *Kapitel 3.7.2* entnommen werden. Je nach gerätespezifischen Einstellungen unterscheiden sich das auf dem Monitor angezeigte Bild, die Druckausgabe eines Druckers oder auch das von der Kamera erfasste Bild erheblich in der Farbgebung. Wenn dasselbe Bild auf verschiedenen Anzeigeräten betrachtet wird, kann ein unkalibrierter Bildschirm zu unterschiedlichen Farbwahrnehmungen seitens der Betrachter führen. Dasselbe gilt für die meisten Komponenten zur Verarbeitung von Farbinformationen, insbesondere für Kameras. Denn auch die Farben eines aufgenommenen Bildes einer Kamera unterscheiden sich so, dass die Farben desselben Objekts von unterschiedlichen Geräten fotografiert verschieden wahrgenommen werden können. Um ein farbgetreues Bild zu erhalten, können die charakteristischen Farbeinstellungen der Endgeräte durch Farbprofilierungen, mittels ICC-Profilen nahezu angeglichen werden. Dies ist ebenfalls wichtig für die Beurteilung und Analyse von sensorischen Farben. Je nach äußerem Einflussfaktor und Stimulus verändert sich die Farbe der sensorischen Farben, irreversibel von einem Farbzustand in den anderen. Häufig erfolgt die Farbveränderung nicht spontan und diskret, sondern entsprechend der Intensität einer Kontamination. Deswegen ist für eine Analyse eine genaue Farberfassung unabdingbar. Im methodischen Teil (*Kapitel 3.7.3*) wurden bereits Smartphones ausfindig gemacht, die referenziert wurden, indem ein standardisiertes Target (Colorchecker Classic) der Firma X-Rite erfasst wurde. Zur Charakterisierung der Kameras können ICC-Profile eingesetzt werden, die definieren, wie die Farbinformationen der fotografierten Bilder relativ zum vorliegenden Referenzfarbraum zu interpretieren sind. Zu diesem Zweck werden ICC-Farbprofile verwendet, um leichte Farbabweichungen der Smartphone Kameras, zu kompensieren. Auf diese Weise lässt sich ein ICC-Konvertierungsprofil erstellen, das Konvertierungseinstellungen enthält, die zu einer möglichst geräteneutralen Bilddatei führen sollten. Entsprechend beschreiben ICC-Farbprofile den Farbraum eines Farbeingabe- oder Farbausgabebeugerätes (Green, 2010, S. 19 ff.). Damit die resultierenden Farben der Bilddatei weitestgehend angenähert oder angepasst werden, wandelt ein Farbtransformationsprozess die Farbinformationen aus dem Quellprofil (das an das Bild angehängt ist) in das Zielprofil um. Die Transformation und somit die mit ihr eingehende Umrechnung über die Rendering Intents erfolgt dabei über einen ausgedehnten und geräteneutralen Schnittstellenfarbraum, den Profile Connection Space (PCS). Der PCS gleicht verschiedene Farbräume einander an. In vielen Fällen werden die Kameraaufnahmen allerdings nicht mit einem Farbprofil (ICC-Profil) versehen, sondern basieren auf einem Standard sRGB Farbprofil im JPG-Format. So ignorieren viele Endgeräte, die von anderen Endgeräten aufgenommenen Farbprofile und betrachten diese lediglich im sRGB Farbraum. In der Vergangenheit verwendete Apple für seine iPhones (iOS) den Standard RGB Farbraum (sRGB). Aktuell setzt Apple den DCI-P3 Farbraum ein, der als ICC-Profil den aufgenommenen Bildern angehängt wird. Der Vergleich der Farbräume und ihr Farbfassungsvermögen kann aus Abbildung 91 entnommen werden. Mit der Version Android 8.0 und höher von Google werden alle das Betriebssystem verwendenden Smartphones mit einem Colormanagementsystem ausgestattet. Nachfolgend wird ein serverbasierter Colormanagementprozess vorgestellt, der in der zuvor vorgestellten Leseapplikation integriert ist und vordergründig für die Farbangleichung der Smartphone Kameras verantwortlich ist.

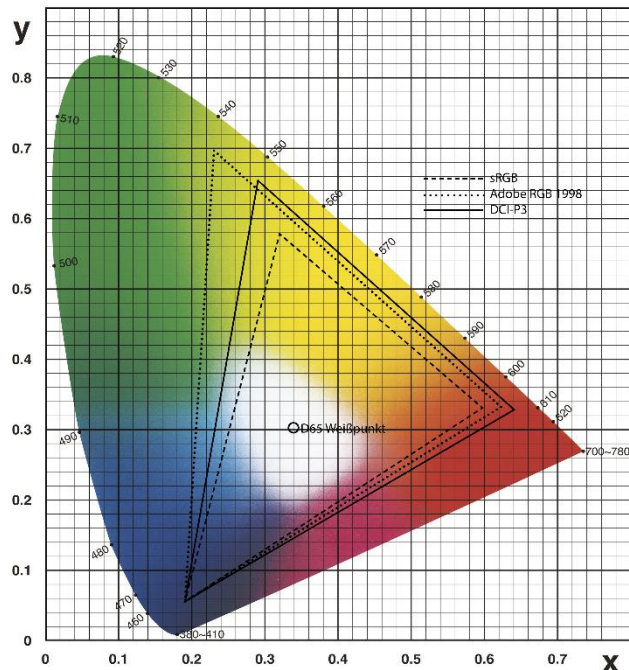


Abbildung 91 sRGB, Adobe RGB (1998) und DCI-P3 im Vergleich (Konica Minolta, 2020)

Der Colormanagement Workflowprozess ist ein integrativer Bestandteil der zuvor behandelten Leseapplikation und wurde primär über die Python 3 PyCMS Programmbibliothek realisiert. Entsprechend wird auf die Programmbibliothek über eine Python Imaging Library (PIL) zugegriffen. Die PIL ermöglicht das Öffnen, Manipulieren und Speichern von Bilddateiformaten. Sie wird verwendet, um Smartphone Eingabebilder, mit entsprechendem ICC-Profil, in ein nahezu neutralisiertes Ausgangsbild zu konvertieren.

Bevor das Bild eines intelligenten Codes ausgewertet werden kann, sollten die Bilder verschiedener Kameras, hier insbesondere von Smartphones, mit einem Colormanagementsystem (CMS) neutralisiert werden, um eine möglichst einheitliche Farbgenauigkeit für verschiedene Smartphones zu erreichen. Die Funktionalität des hier behandelten Colormanagement Algorithmus wurde von Xie (2020) anhand der in Tabelle 12 aufgelisteten Smartphones untersucht. Abbildung 92 stellt das Konzept des serverseitigen Colormanagement Workflowprozesses, in Form eines Flussdiagramms, dar. Das entsprechende Skript kann aus Anhang (G) entnommen werden. Zu Beginn (Start) werden die zuvor erstellten ICC-Profile der ausgewählten Smartphones (*Kapitel 3.7.3*) in Form eines Dictionary (Wörterbuch) aufgelistet. Als nächstes wird das Smartphone des Benutzers, der die Leseapplikation anfordert, durch den zuvor beschriebenen nahezu automatisierten Geräteerkennungprozess (*Kapitel 6.6.3*) erkannt. Dieser sucht innerhalb des Dictionary nach dem Smartphone und wartet auf eine Eingabe (Input) des Benutzers, der sein Smartphone an dieser Stelle auswählen muss. Wie bereits zuvor erwähnt, basieren viele Bilder auf den Farbräumen sRGB und einige jedoch auf dem Farbraum DCI-P3. Um ein Ausgabeprofil zu bestimmen, wird eine entsprechende Abfrage ausgeführt, die die Kriterien für das Smartphone prüft. Im nächsten Schritt wird eine Profile-to-Profile-Transformation durchgeführt. Diese basiert auf dem im *Kapitel 3.7.2* beschriebenen RI „relativ farbmtrisch“ (farbmtrisch exakte Konvertierung mit Anpassung des Weißpunktes). Nach Abschluss der Transformation wird das korrigierte Bild angezeigt, entsprechend auf dem Server hinterlegt und steht für den nächsten Verarbeitungsprozess zur Verfügung.

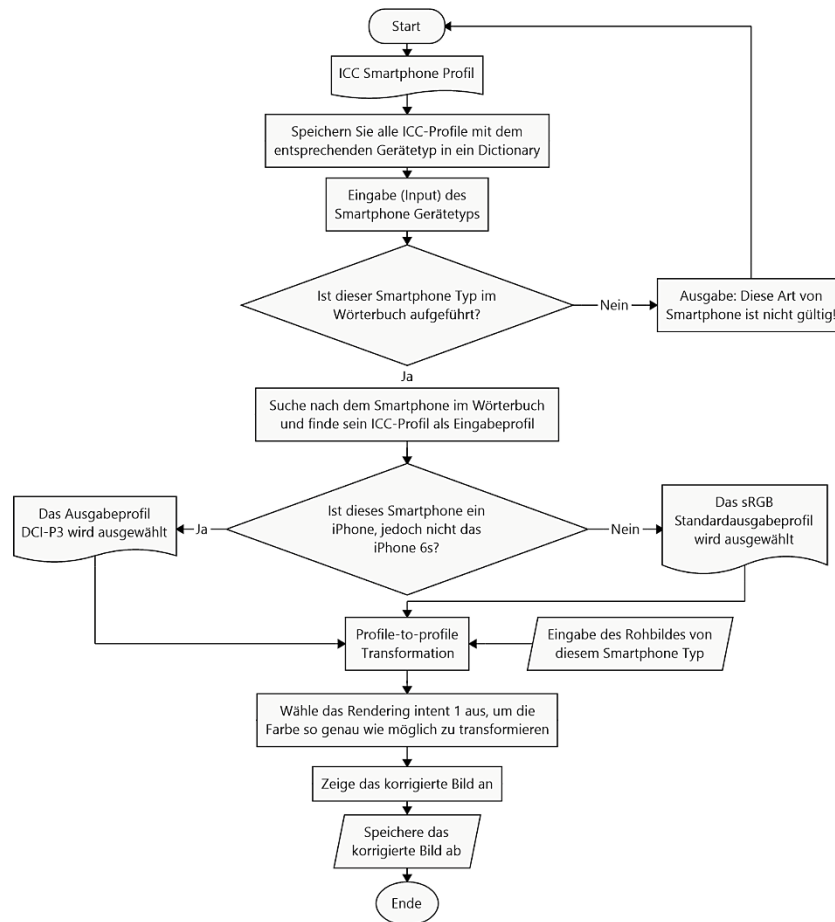


Abbildung 92 Flussdiagramm Colormangement

Die Profile-to-Profile Transformation (RI 1) wurde mittels einer ICC-Profilierung ausgewählter Smartphones durchgeführt. Die Auswertung der Farbdifferenz ( $\Delta E$ ) zwischen den verschiedenen Geräten vor und nach einer Farbtransformation ist in Tabelle 26 aufgeführt. Für die Durchführung des Colormangementprozesses wurde die Colormangement Software i1 von X-Rite verwendet. Anhand der Ergebnisse ist zu erkennen, dass alle mit den Smartphones aufgenommenen Bilder eine signifikante Veränderung nach dem Colormangement Workflowprozess zeigen. Somit konnten alle Smartphone Bilder aufeinander abgestimmt und angenähert werden. Die charakteristischen Farbeinstellungen der Kameras wurden somit harmonisiert.

Marke	Typ	$\Delta E_{76}$ (vorher)	$\Delta E_{00}$ (vorher)	$\Delta E_{76}$ (nachher)	$\Delta E_{00}$ (nachher)
Apple	iPhone 6S	9,8	5,6	1,6	1,0
Apple	iPhone 7 Plus	11,8	5,5	7,8	4,5
Apple	iPhone 8 Plus	11,5	5,5	7,2	5,1
Apple	iPhone X	13,1	6,3	10,1	4,2
Apple	iPhone 11	18,1	10,2	7,3	4,0
Samsung	Galaxy S7	9,0	5,9	3,1	1,9
Samsung	Galaxy S8	13,3	8,8	3,4	2,6
Samsung	Galaxy S9	13,4	9,4	6,4	3,6
Huawei	Honor 6X	10,2	6,0	1,7	1,1
Huawei	Honor7i	12,4	9,3	2,5	1,6
Huawei	P20 Lite	17,5	10,1	2,5	1,5
Huawei	P30 Pro	10,5	5,4	2,3	1,4

Tabelle 26 Farbunterschiede der Testchart (Kapitel 3.7.3) verschiedener Smartphones

## 6.7 Kommunikationstechnologien

Die Kommunikationstechnologie baut, wie bereits im *Kapitel 3.8* methodisch erläutert auf dem Client-Server-Modell auf. Das hierbei zugrundeliegende System der Soft- und Hardwarearchitektur basiert auf einem mehrdimensionalen intelligenten Code, auf Basis von sensorischen Farben. Der intelligente Code wird über eine auf dem Endgerät (Smart Device) vorinstallierte Client Applikation (Thin Client) erfasst, welche eine geräteunabhängige Kommunikation initiiert und ein Bild des intelligenten Codes an einen Server zur Analyse übermittelt. Die Resultate der auf dem Server durchgeführten Analyse werden als Ergebnisbericht an das Endgerät zurückgesandt. Durch die informationstechnologische Vernetzung mehrerer intelligenter Codes über das Internet der Dinge können feinere Informationsgeflechte ausgewertet, evaluiert und autonom überwacht werden (*Kapitel 6.7.2*). Daher setzt sich das vorliegende Kapitel mit den Grundlagen der Internettechnologie, dem Internet der Dinge und dem Themenfeld der Datensicherheit und Verschlüsselung auseinander.

### 6.7.1 Internettechnologien

Im Folgenden werden zum Verständnis des vorliegenden Kapitels der Internettechnologien Grundlagen vorgestellt, die für die Kommunikation mit dem Internet notwendig sind. Ein wesentlicher Aspekt ist hierbei das TCP/IP-Referenzmodell (Transmission Control Protocol / Internet Protocol). So ist TCP ein verbindungsorientiertes Transportprotokoll, bei dem Datagramme strukturiert als Header mit Quell- und Ziel-IP-Adressen, inkl. Portnummer und Prüfsumme, zusammen mit dem Payload (Daten eines Datenpakets) an ein Netzwerk übergeben werden (Ernst, 2003, S. 725). Beim Erhalt des Payloads seitens des Empfängers wird eine Bestätigung innerhalb einer vorgegebenen Zeit erwartet. Wird dieser Vorgang nicht erfüllt, wird das Datagramm erneut gesendet und im ungünstigsten Fall eine Fehlermeldung generiert, wenn der Empfänger nicht reagiert. TCP erfüllt nicht unbedingt die Voraussetzung für verzögerungsfreie Echtzeitübertragungen bspw. bei einer Audio- oder Videoübertragung (ebd.), da der Empfang jedes Pakets vorher geprüft und bestätigt werden muss. IP hingegen ist ein verbindungsloses Netzwerkprotokoll, das durch Auswertung einer Adressierung (IP-Adresse) und dem daraus resultierenden Routing (Wegewahl) für das Versenden und Weiterleiten von Datagrammen zuständig ist (ebd.). Für den Benutzer sind DNS-Namen (Domain Name System), wie bspw. [www.mustafabilgin.de](http://www.mustafabilgin.de), komfortabler, da sie einfacher einzuprägen sind als IP-Sequenzen, wie 178.202.190.114. Aus diesem Grund sind DNS-Dienste verantwortlich für die Beantwortung von Anfragen zur Namensauflösung.

OSI-Schicht	TCP/IP-Schicht	Beispiel
Anwendung Darstellung Sitzung	Anwendung	HTTP(S)
Transport	Transport	TCP / UDP
Vermittlung	Internet	IP (IPv5, IPv6)
Sicherung Bitübertragung	Netzzugang	Ethernet (IEEE 802.3) WLAN (IEEE 802.11)

*Tabelle 27 TCP/IP und ISO/OSI Referenzmodell im Vergleich*

Entgegengesetzt eignet sich das verlustbehaftete, unsichere und verbindungslose UDP (User Datagram Protocol) Transportprotokoll. Hierbei trägt das jeweilige Datagramm eine Portnummer des Empfängers, ohne, dass eine Bestätigungsanforderung gesendet wird. Beim UDP-Transportprotokoll findet kein Manipulations- oder Fälschungsschutz statt, der die übertragenen Daten vor Angriffen schützt. Zur Verschlüsselung von Medien und Datenströmen auf Basis von UDP, greift WebRTC unter anderem auf den DTLS (Datagram Transport Layer Security) zu, das einen anfänglichen Handshake für die jeweilige Medienkomponente realisiert und auf TLS basiert. DTLS soll das Abhören oder die Manipulation einer Kommunikation, die beim Transport von Datagrammen, die vom UDP bereitgestellt werden, verhindern (Kurose & Ross, 2013, S. 5 f.). Hierbei findet der Schlüsselaustausch in der Medienebene statt. Sobald der DTLS Handshake abgeschlossen ist, werden die Schlüssel exportiert und verwendet, um SRTP (Secure Real-Time Transport Protocol) für die verwendeten Medienkanäle zu verschlüsseln. Dieser Prozess gewährleistet die Unversehrtheit (Integrity) der gesendeten Pakete, die Beglaubigung beider Peers (Authentication) und den Datenschutz (ebd.).

## 6.7.2 Internet der Dinge

Der Terminus *Internet of Things* (IoT) – zu Deutsch *Internet der Dinge* – wurde 1999 von Ashton eingeführt. Hierzu ergänzt er später (2015) folgendes: „[...] because of the Internet of Things, computers can sense things for themselves. It’s only been a few years, but we already take networked sensors for granted“. Damit ermöglicht IoT die Konnektivität von verschiedenen Sensoren mit der physischen Welt über das Internet. Brand et al. (2009, S. 11) wiederum beschreiben Internet der Dinge folgendermaßen: „Objekte mit eigener dezentraler Intelligenz vernetzen sich, tauschen Informationen aus und bewegen sich autonom in ihrer Umgebung“. Heute ist der Begriff *Internet der Dinge* jedoch ein Sammelbegriff für verschieden miteinander vernetzter Objekte geworden, der je nach Autor abweicht.

Die Kombination sensorischer Farben, intelligenten Codes, Smart Devices und einem zentralen Server ermöglicht den Aufbau des Internet der Dinge. Entsprechend überwachen sensorische Farben als integrierter Bestandteil intelligenter Codes, aufgebracht an Objekten, autonom ihre Umgebung. Nach jedem Scanvorgang über eine serverseitige Leseapplikation aufgerufen von einem Smart Device, werden die jeweiligen Informationen der intelligenten Codes unidirektional in eine serverseitige Datenbank eingetragen und mit einem Zeitstempel versehen. Durch die Anzahl an intelligenten Codes auf verschiedenen Waren, wie bspw. Verpackungen, können nach jedem Scanvorgang feinere Informationsgeflechte serverseitig gesichert, ausgewertet und evaluiert werden.

## 6.7.3 Datensicherheit und Verschlüsselung

Gerade bei der Entwicklung von Webanwendungen steht die Relevanz des Datenschutzes und der Datensicherheit außer Frage. Sie ist ein wesentlicher Faktor bei der Webentwicklung. Die hier vorgestellten präventiven Maßnahmen sind dabei in der Webapplikation integrierten. Im Fokus der WebRTC-Sicherheit stehen sowohl der Benutzer als auch der Browser. Vor dem Medienzugriff muss der Browser immer eine Benutzerberechtigung über die zu verwendenden Mediengeräte wie Kamera und Mikrofon über die Funktion *getUserMedia* (Snippet 3) einholen. So wird der Zugriff auf die Kamera, bspw. im Firefox Browser (Abbildung 93), mit einem roten Kamerasymbol angezeigt.

Der Benutzer kann den dabei temporär erlaubten Zugriff jederzeit beenden. Bei unsicheren Zugriffsbedingungen und -umgebungen, wie bei einer fehlenden Zertifizierung, bleibt die Funktion *navigator.mediaDevices* undefiniert, verhindert den Zugriff auf *getUserMedia* und führt zu einem *TypeError*.

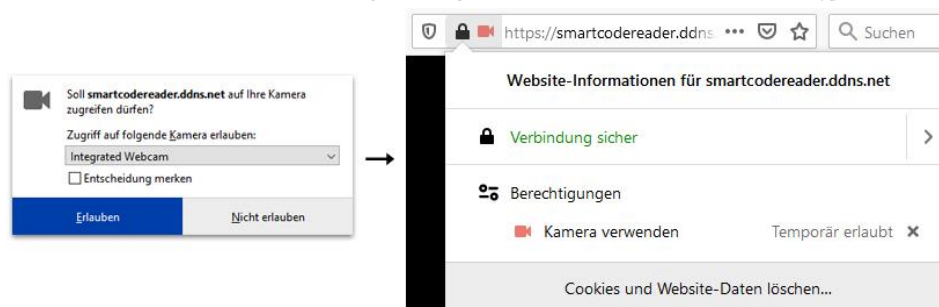


Abbildung 93 Zugriff auf Mediengeräte

Dabei kann nur ein Dokument der obersten Ebene (*Index*) eines Fensters die Erlaubnis zur Verwendung der Funktion *getUserMedia* anfordern. Entsprechend können die Browser nur eine einmalige Erlaubnis pro Domain einholen. Eine zertifikatbasierte Verbindung erlaubt, basierend auf dem Netzwerkprotokoll HTTP(S) / TLS (Transport Layer Security) oder auch auf dem lokalen Aufruf (Localhost), den Zugriff auf die Mediengeräte des Clients.

HTTPS (Hypertext Transfer Protocol Secure) ist ein Kommunikationsprotokoll, das die Protokolle HTTP und SSL (Secure Sockets Layer) / TLS verwendet, um eine verschlüsselte Kommunikation und sichere Identifizierung eines Webservers zu ermöglichen. Somit wird eine relativ sichere Verbindung innerhalb eines unsicheren Netzwerks geschaffen, um gerade Lauschangriffen Dritter, wie Man-in-the-Middle Angriffen, durch eine asymmetrische Verschlüsselung<sup>16</sup> entgegenzuwirken. HTTP selbst ist unverschlüsselt und durch Dritte im Netzwerk mit Leichtigkeit auszulesen. Aus diesem Grund werden sogenannte SSL Zertifikate eingesetzt, die serverseitig bereitgestellt werden. Letztere sind ebenfalls in der Abbildung mit dem grünen Schriftzug „Verbindung sicher“ zu erkennen, der auf eine gültige Zertifizierung hinweist. Dabei arbeitet das HTTP auf der höchsten Schicht des TCP / IP-Schichtmodells, der Anwendungsschicht, ebenso wie das Sicherheitsprotokoll SSL / TLS, das sich auf der Transportschicht verortet.

Ein weiterer Mechanismus zur Sicherung der Webanwendung ist die Cross-Site-Request-Forgery (CSRF), was auf Deutsch so viel wie „seitenübergreifende Anfragefälschung“ bedeutet. Beim CSRF versuchen Angreifer eine vom Benutzer autorisierte Session (Session Riding), bei der eine Verbindung zwischen dem Client und dem Server besteht, zu übernehmen und damit über HTTP-Request schadhafte Aktionen durchzuführen. Dies könnte bspw. ein vergessener Log-out auf einem online Shop-Konto sein, bei dem der Benutzer für eine Dauer der Session angemeldet bleibt. Beim Besuchen diverser schadhafter Webseiten, wie bspw. *www.evil.com*, kann dieser versehentlich eine URL (*Uniform Resource Locator*) betätigen, der mit der Identität des Benutzers einen HTTP-Request an die noch aktive Session des Online Shops sendet. Dieser erkennt die vermeintliche Aktion des Benutzers aufgrund der entsprechenden Cookies, welche die Session als aktiv und als legitim erkennen, an und gewährt somit den Zugriff auf Betätigungen diverser Einkäufe. Eine andere Möglichkeit ist die Durchführung von Cross-

<sup>16</sup> Verschlüsselt wird über einen öffentlichen Schlüssel (Public Key) und entschlüsselt wird über einen privaten Schlüssel (Private Key).

Site-Scripting (XSS), die auch als HTML-Injection bezeichnet werden, bei der durch bspw. Formularfelder oder vom Angreifer versteckte Hintertüren in vorgefertigten, auch nützlichen Skripten (vermeintlich als Hilfe oder Effekt getarnt) einer bestehenden Website untergejubelt werden.

Der im *Kapitel 6.3* genauer unter dem Snippet 20, erfolgte Schritt der Funktion *sendCodeToServer*, welcher den intelligenten Code an den Server sendet, wird hier fortgesetzt. Um serverseitige Abwehrmaßnahmen gegen CSRF vorzunehmen und somit CSRF-Angriffen entgegenzuwirken, werden unter der Funktion Maßnahmen unternommen. Hierzu wird jede serverseitig ausgeführte Transaktion der Webanwendung mit einer geheimen Information, dem Synchronizer Token Pattern (STP), versehen. Unsichere Anfragemethoden, wie bspw. POST, PUT und DELETE, werden vom vorzustellenden Algorithmus mit einem *csrfmiddlewaretoken* übergeben, welcher pro Benutzer und Sitzung gültig ist. Beim STP wird ein Token, meistens eine Zahlenkombination oder eine Zeichenkette, in einem versteckten Feld auf der Seite eingebunden und ist in der Regel für den Angreifer nicht auslesbar. Die Funktion *sendCodeToServer* sendet dazu den gefunden intelligenten Code (*foundCode*) zusammen mit dem Token (*csrfmiddlewaretoken*) mit der Zeichenkette „Efc22Hewjxkw...“ an den Server:

1	function sendCodeToServer() {
2	var dataURL = document.getElementById("foundCode").toDataURL("image/png");
3	var post_data = {
4	'csrfmiddlewaretoken': "Efc22Hewjxkw72yae4hiq8cPerDUWknoYk4mFKH6uTdc7VongtiSb1zO",
5	imageBase64: dataURL, };

*Snippet 54 Maßnahme gegen Cross-Site-Request-Forgery*



## 6.8 Serverarchitektur

Die Serverarchitektur wird auf Basis eines Einplatinencomputers dem, Raspberry Pi, aufgebaut. In der gewählten Konfiguration ist der Raspberry Pi so hinreichend ausgestattet und performant, dass er als Webserver und Datenbank in einer Laborumgebung die Leseapplikation vollumfänglich sowie sicher und stabil ausführen kann. Die wesentlichen implementierten Basissysteme sind Linux als Betriebssystem, eine Webserverstruktur, ein DNS Client sowie die höhere Programmiersprache Python, ergänzt um eine Vielzahl von Programmbibliotheken. Die Leseapplikation wurde prototypisch entwickelt, ist jedoch in ihrer Funktionalität, in den Sicherheitsstrukturen und im Aufbau hinreichend für eine professionelle Implementation ins offene Netz, das Internet. Um die Performanz zu steigern, wurden verschiedene und nicht verwendete Dienste deaktiviert, der Standardbenutzer für Root-Zugriffe entsprechend umbenannt und die Zugriffsrechte des Linux Systems angepasst. Ein weiterer Schritt war dabei die Netzwerk-Konfiguration, die Freigabe entsprechender Ports und das Einrichten und Bestimmen einer statischen IPv4 Adresse aufseiten des Routers. Der Dienst eines dynamischen DNS inklusive eines Hostnamens wurde ebenfalls in Anspruch genommen. Hierzu sendet der DNS Client bei einer entsprechenden Veränderung der IP Adresse seitens des Routers, dem DNS Dienst über HTTP automatisiert die aktualisierte IP Adresse und ermöglicht damit den stetigen Zugriff aus dem Internet. Abbildung 94 zeigt die Struktur der serverseitigen Hardware sowie mögliche Verbindungen zu verschiedenartigen Clients.

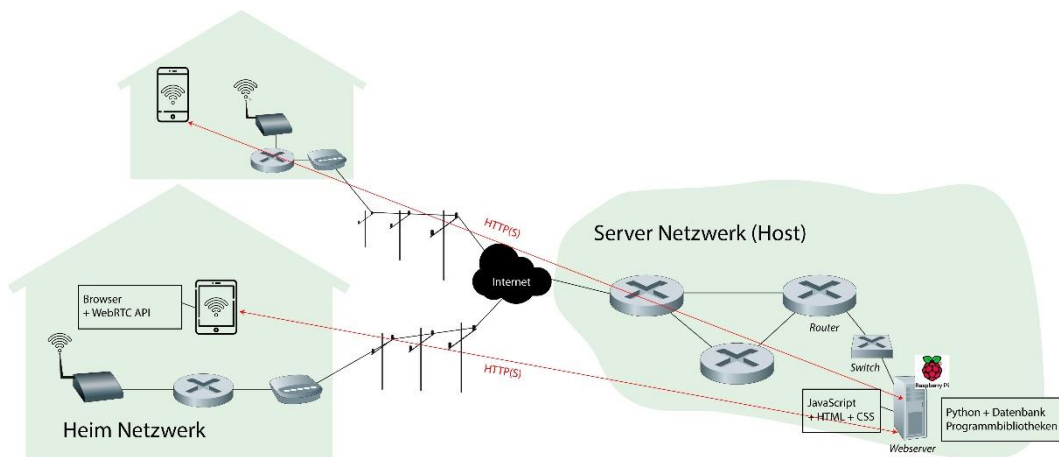


Abbildung 94 Hardwarearchitektur angelehnt an Kurose und Ross (2013, S. 85)

## 6.9 Praxisnahe Umsetzung

Die praxisnahe Umsetzung der Leseapplikation wird im Folgenden für eine browserbasierte Oberfläche z.B. eines Laptops oder eines Desktop PCs dargestellt. Bei der Eingabe der Domain <https://smartcodereader.ddns.net/> wird das Interface samt Leseapplikation aufgerufen und der Benutzer bezüglich der Zugriffsberechtigung auf die Mediengeräte, hier insbesondere auf die Kamera des Clients, gefragt (Abbildung 95). Bei einer Zugriffsfreigabe wird die Leseapplikation gestartet. Daraufhin erscheint ein aktives Kamerasymbol in der oberen Mitte des Firefox Browsers, welches auf eine aktive Session hinweist (Abbildung 96).

Ebenfalls ist in der Abbildung rechts eine Menüleiste zu sehen, die als oberstes Symbol für eine Scanübersicht steht, welche die bisherigen Scans der Historie des intelligenten Codes zur Verfügung stellt. Das Symbol in der Mitte steht für verschiedene Statussymbole, auf die später näher eingegangen wird. Das unterste Symbol steht für eine Vollbilddarstellung.

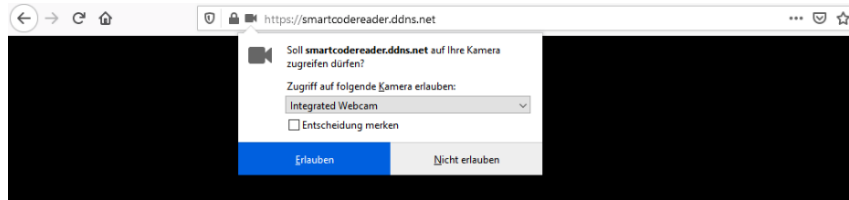


Abbildung 95 Wartet auf Zugriffsberechtigung

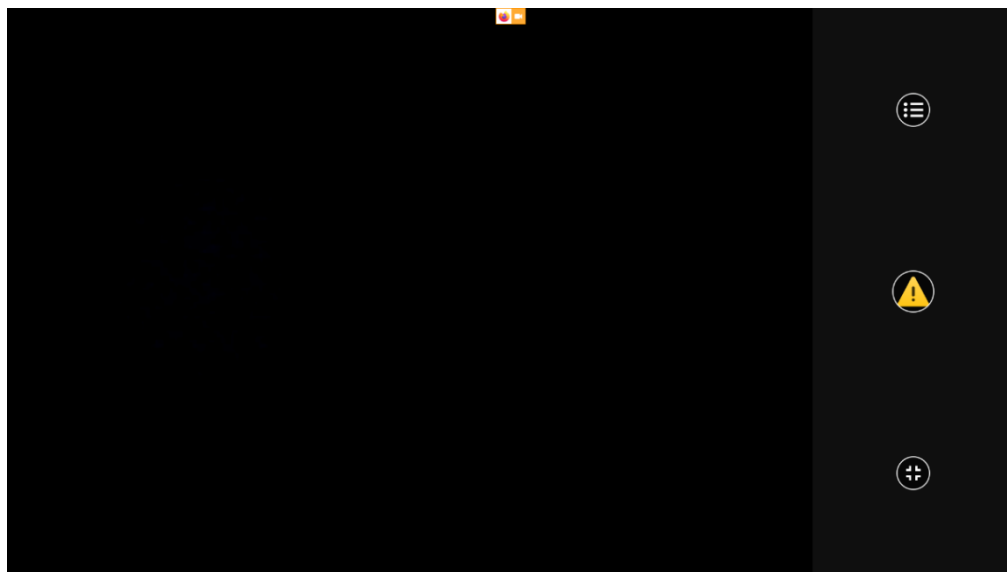


Abbildung 96 Zeigt aktive Kamera und Interface im Browser Firefox an

Beim Betätigen des obersten Symbols der Menüleiste wird eine Scanübersicht aufgerufen (Abbildung 97), die, wie zuvor erwähnt, die bisherigen Aufnahmen des intelligenten Codes und dessen Sensordaten, und entsprechend den Zielforderungen deren Historie, zur Verfügung stellen. In der Scanübersicht findet sich zum einen die Detailansicht mit den Einzelfarbwerten der Sensoren und zum anderen ein Vergleich zu zwei Messzeitpunkten.

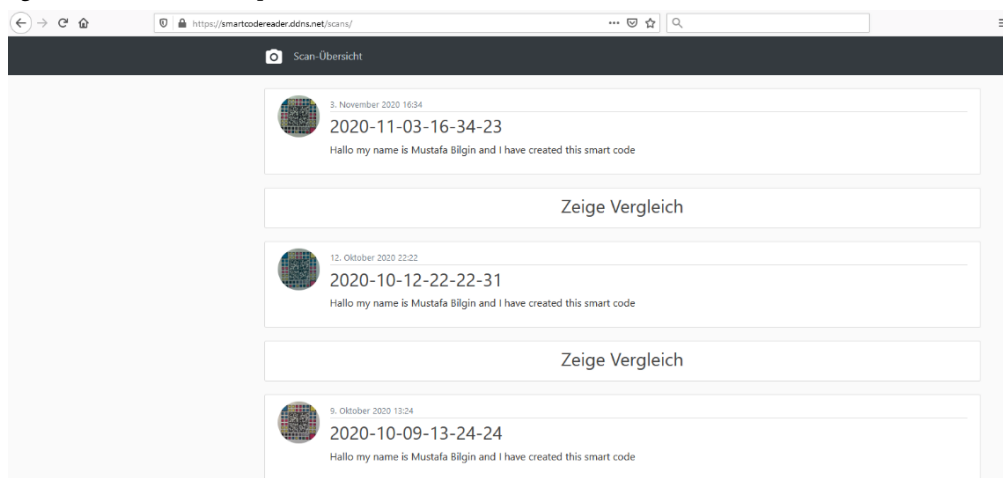


Abbildung 97 Scanübersicht mit Historie des intelligenten Codes

Beim Betätigen der Verlinkung, hinterlegt auf dem jeweiligen Zeitstempel der Aufnahmen oder der Verlinkung „zeige Vergleich“, wird die Detailansicht (Abbildung 98) aufgerufen. Diese basiert auf der für die Gegenüberstellung angepassten Auswerteeinheit (Kapitel 6.5). Hierin werden die Bilder des Codes, vor und nach einer Stimuli induzierten Farbveränderung, gegenübergestellt dargestellt.

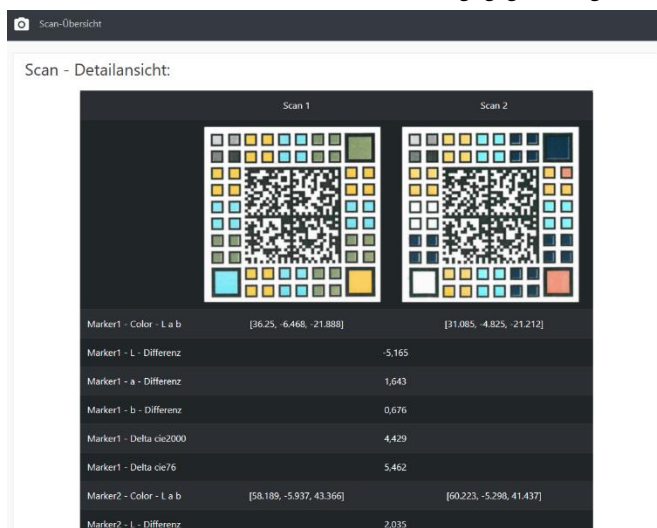


Abbildung 98 Detailansicht eines Vergleichs zwischen zwei Messzeitpunkten

In Gegensatz zu der oben beschriebenen Darstellung bei einer browserbasierten Oberfläche erkennt das implementierte responsive Webinterface beim Zugriff auf die Webapplikation mit einem Smart Device, hier insbesondere einem Smartphone, das jeweilige Endgerät sowie seine Parameter (Kapitel 6.6.3). Es passt die Ausgabe automatisch an die Bildschirmauflösung des Endgeräts an. Das Symbol in der Mitte steht für verschiedene Statussymbole (Abbildung 99). Das gelbe Symbol weist den Benutzer darauf hin, sein Smartphone ruhig zu halten. Das rote Symbol weist den Benutzer hin, den Aufnahmeprozess des intelligenten Codes zu wiederholen, da bei dessen Verarbeitung ein Fehler aufgetreten ist. Das letzte Symbol, die Sanduhr, weist auf die Verarbeitung des intelligenten Codes hin.

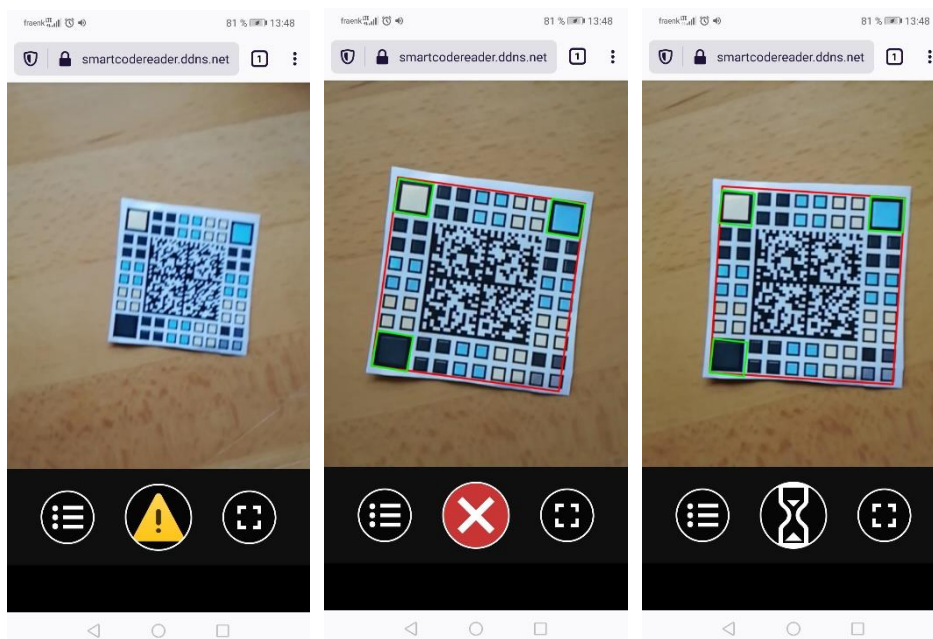


Abbildung 99 Responsive Webinterface und Codeerfassung samt Statussymbole

## Kapitel 7

# Validierung

Dieses Kapitel setzt sich mit der Validierung und somit der Nachweisführung, insbesondere der Erfüllung der spezifizierten Zielforderungen (*Kapitel 1.1*), auseinander. Dabei wird Validierung als ein Prozess der Beurteilung eines Systems oder einer Komponente am Ende eines Entwicklungsprozesses betrachtet (IEEE, 1990).

Die übergeordneten Ziele dieser Dissertation bestanden darin:

- I. Instrumente zu entwickeln, mit der die Auswirkungen von Umwelteinflüssen auf ein Objekt kontrolliert, erfasst und validiert werden können.
- II. Die Reaktion gedruckter sensorischer Farben auf Umwelteinflüsse zu untersuchen und herauszuarbeiten, inwieweit die mit den Farben verbundenen Informationen eine Möglichkeit darstellen, eine valide Aussage über die Umwelthistorie zu treffen.

Um das erste Ziel zu realisieren, wurden zwei Instrumente erarbeitet, die im Folgenden validiert werden und somit zu dem in den Zielforderungen definierten und vom Konzept (*Kapitel 1.3*) dargestellten Erwartungshorizont führen. Das erste Instrument basiert auf einem intelligenten Code (*Kapitel 5.3*). Dieser dient einerseits der Überwachung und Sicherung der Auswirkungen von Umwelteinflüssen, wie hier speziell Wasser / Feuchte (Hydrochrome), pH-Wert Änderung (Halochrome) und UV-A Lichteinfluss (Photochrome). Zudem besitzt der Code invariante Bereiche, in denen statische bspw. kennzeichnende und administrative Informationen gespeichert werden können. Der Entwicklungsprozess von sensorischen Farben (*Kapitel 4.6*) und deren Weiterverarbeitung zu sensorischen Tinten (*Kapitel 4.7*) wurde gleichsam in den entsprechenden Kapiteln behandelt. Mittels Inkjetdruckverfahren (*Kapitel 3.3*) wurden intelligente Codes auf geeignetes Haftpapier gedruckt (*Kapitel 4.8*). Dabei substituieren die sensorischen Tinten die Buntfarben (CMY) des Druckers. Gleichzeitig resultieren geeignete Einstellungen des Farbmanagementsystems darin, dass annähernd keine Mischfarben gedruckt werden. Das Ergebnis eines funktionalen intelligenten Codes zeigt Abbildung 100 in zwei Reaktionszuständen, jeweils vor (links) und nach (rechts) einem Reiz-Reaktionsprozess entsprechend mit den damit korrespondierend hinterlegten Farbwerten. Somit bilden die Farbzustände der hydrochromen (türkis – transparent), halochromen (gelb – rot) und photochromen (grün – dunkelblau) sensorischen Farbe den jeweiligen Reaktionszustand ab.

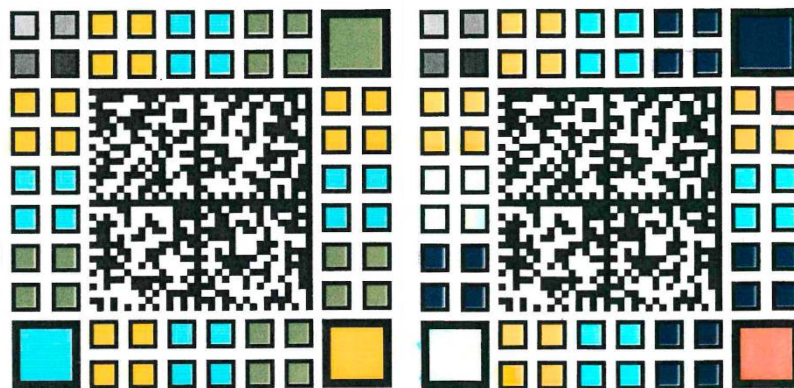


Abbildung 100 Vollständig gedruckter intelligenter Code, links: nicht reagiert und rechts: reagiert

Somit konnte das erste Instrument, der intelligente Code, validiert werden. Damit dieser jedoch erfasst, analysiert und ausgewertet werden kann, wird entsprechend das zweite Instrument, eine Leseapplikation, benötigt. Die Softwarearchitektur (Kapitel 6.2), die der Leseapplikation zugrunde liegt, basiert auf mehreren Algorithmen. Die Basis des kohärenten Systems bilden unter anderem der Erkennungsprozess (Kapitel 6.3) und die Separierung (Kapitel 6.4) des intelligenten Codes. Das System der Leseapplikation wird dabei durch qualifizierte Algorithmen ergänzt, welche verschiedene Störeinflüsse durch eine Farb-anpassung einen Colormanagementprozess sowie eine Helligkeits- und Kontrastanpassung beseitigen (Kapitel 6.6). Zum Vermeiden von unscharfen und verwackelten Aufnahmen wurde zusätzlich eine Blur detection (Unschärfeerkennung) (Pech-Pacheco et al., 2000) in die Leseapplikation programmiert. Eine entsprechend zugeschnittene Auswerteeinheit (Kapitel 6.5), die dem Punkt II entspricht, wertet die sensorischen Felder des intelligenten Codes aus. Deren Farbinformationen werden quantifiziert und erlauben valide Aussagen über eine vorrausgegangene Kontamination des Codes. Durch Messungen in zeitlicher Folge kann schließlich die Umwelthistorie des Codes transparent gemacht werden. Die Software der Leseapplikation ist in eine Hardwarearchitektur (Kapitel 3.8) eingebettet. In diesen Kommunikationstechnologien werden zudem angemessene Vorkehrungen zur Datensicherheit, zur Sicherstellung der Datenintegrität und Verschlüsselung getroffen (Kapitel 6.7).

Durch die vollständige Realisierung der beiden Instrumente, zum einen des entwickelten intelligenten Codes samt sensorischer Farben und zum anderen einer serverbasierten Leseapplikation, konnte der Entwicklungsprozess des Gesamtsystems vollständig validiert werden.

## 7.1. Gesamtprozess

Die in den Zielanforderungen bestimmten Unterziele werden im Folgenden zusammenfassend geprüft, um den Gesamtprozess zur Erfüllung der Zielanforderungen zu validieren. Das Unterziel des Funktionstests und dessen Evaluation wurden bereits zuvor behandelt.

### 1) ENTWURF EINES GESAMTSYSTEMS

Das grundlegende Konzept, sensorische Farben als Bestandteil eines Codes zu applizieren und auszuwerten, wurde im Architekturentwurf (Kapitel 1.3) aufgegriffen und tiefergehend durch das Client-Server-Modell (Kapitel 3.8) ergänzt, um auch die Kommunikationstechnologien in eine Gesamtarchitektur zu integrieren. Die Leseapplikation basiert dabei zum einen auf der Softwarearchitektur (Kapitel 6.2)

und zum anderen auf dem Client-Server-Modell. Die beiden Komponenten der Soft- und Hardwarearchitektur bilden gemeinsam ein kohärentes Gesamtsystem. Dabei wurde zusätzlich die Funktionalität anhand eines praxisnahen Funktionstest, bei dem alle Funktionen getestet wurden, validiert.

## 2) UNTERSUCHUNG UND ENTWICKLUNG SENSORISCHER FARBEN

Der Schwerpunkt dieses Unterziels lag darin, geeignete irreversible sensorische Farben zu entwickeln und zu untersuchen, da zum Zeitpunkt der Niederschrift dieser Dissertation einerseits keine inkjetfähigen und irreversiblen sensorischen Farben zu erwerben und aufzufinden waren und andererseits in diesem Bereich Forschungslücken zu verzeichnen waren. Eine weitreichende Recherche zu dem Themenbereich der sensorischen Farben, insbesondere zu den gegenwärtigen Erkenntnissen des Forschungsstands, erfolgte dabei im *Kapitel 2 Stand der Forschung*. Innerhalb dieser Untersuchung wurden unter anderem Forschungsbedarfe für die Entwicklung irreversibler sensorischer Farben (*Kapitel 2.1*) identifiziert, da aufgrund der vergleichsweise aufwendigen Zusammensetzung der entwickelten Inkjettinten, kaum, bis keine, für den gesuchten Einsatzbereich und die gesuchte Funktionalität, auffindbar waren. Dies veranlasste die Untersuchung und anschließende Entwicklung dreier exemplarischer sensorischer Farben (*Kapitel 4.6*), die unter dem Einfluss eines externen Stimulus ihre Farbe irreversibel von einem farblichen Zustand in einen anderen ändern und in diesem unumkehrbar verbleiben.

## 3) ENTWICKLUNG UND UNTERSUCHUNG SENSORISCHER TINTEN

Die zuvor identifizierten und entwickelten sensorischen Farben wurden in einem weiteren Entwicklungsschritt hinsichtlich ihrer Eignung als Farbstoffadditiv (*Kapitel 3.1.1*) innerhalb einer speziell zugeschnittenen Tintenmatrix untersucht und für den Einsatzbereich als Tinte, für das eingesetzte piezoelektrische Inkjetsystem (*Kapitel 3.3*), angepasst. Entsprechend wurden die Tintenanforderungen (*Kapitel 3.2*) der sensorischen Tinten bestimmt und dem eingesetzten Inkjetsystem iterativ angepasst. Die entwickelten sensorischen Tinten erfüllen die Parameter der Viskosität, Oberflächenspannung und Partikelgrößenverteilung, die in der einschlägigen Fachliteratur für Inkjetdruckverfahren angegeben wurden (Hutchings & Martin, 2013, S. 24; Magdassi, 2010, S. 35; Zapka, 2018, S. 12). Die sensorischen Tinten wurden abschließend einem mehrstufigen Filtrationsprozess (*Kapitel 3.2.4*), ohne den Verlust oder die Einschränkungen der Funktionalität, unterzogen. Dadurch konnte die in der einschlägigen Literatur vorgeschlagene Partikelgröße eingehalten werden (Magdassi, 2010, S. 34). Dass diese für eine Laborumgebung formulierten Tinten den Anforderungen von Industriedruckern genügen, ist vorerst anzunehmen, da sie die grundlegenden Anforderungen an Inkjettinten erfüllen. Möglicherweise sind jedoch für industrielle Anwendungen weitere Additive hinzuzufügen, z. B. zur Stabilisierung (Vermeidung von Entmischung). Die gewählten Farbstoffkonzentrationen sind valide, da eindeutige Reaktionen der Sensoren beobachtet und messtechnisch ermittelt werden konnten.

## 4) UNTERSUCHUNG INTELLIGENTER CODES

Der Fokus dieses Unterziels lag auf der Untersuchung intelligenter Codes zur Speicherung dynamischer sowie statischer Inhalte. Hierzu erfolgte eine weitreichende Recherche aller wesentlichen Konzepte, Veröffentlichungen und Forschungsansätze, wobei Forschungsbedarfe festgestellt wurden. Dazu wurde zum einen das Themenfeld der Smart Labels (*Kapitel 2.3*) untersucht und zum anderen wurden ausgewählte intelligente Codes (*Kapitel 2.2*), welche auf ein Lesegerät bzw. auf eine Leseapplikation und das Inkjetdruckverfahren zurückgreifen, näher untersucht. Gegenüber bisher bekannten Ansätzen (Aida et al., 2020; Gligoric et al., 2019; Heilmann et al., 2008; Prusik et al., 2020; Ribi, 2012) ist der in dieser

Dissertation entwickelte intelligente Code eine neue Art und Form von Code. Da die meisten der bisherigen Verfahren einen Teilbereich eines klassischen Symbolcodes um einen oder mehrere dynamische Bereiche substituieren, jedoch dafür bspw. die Fehlerkorrektur deaktivieren, den Symbolcode teilweise beschädigen oder den Symbolcode vollständig mit sensorischen Tinten verdrucken und bei einem Einflussfaktor unleserlich machen.

#### 5) ENTWICKLUNG EINES INTELLIGENTEN CODES

Die Erkenntnisse der Untersuchung im Stand der Forschung (*Kapitel 2*) wurden bei der Entwicklung eines neuartigen intelligenten Codes berücksichtigt. Entsprechend wurden zusätzlich Kriterien (*Kapitel 5.2*) für die Entwicklung eines intelligenten Codes synthetisiert, die bei der Planung und bei dem Entwurf intelligenter Codes richtweisend Verwendung finden können. So wurden alle Funktionalitäten und Komponenten, die für die Entwicklung eines intelligenten Codes notwendig sind, in den entsprechenden Grundlagen (*Kapitel 5.1*) behandelt und dargestellt. Die Grundlagen flossen dabei in den Aufbau des entwickelten intelligenten Codes ein, der auf Basis eines Gestaltungsrahmens konstruiert (*Kapitel 5.3*) wurde. Der daraus resultierend entwickelte intelligente Code beinhaltet einen zentralen statischen Bereich, welcher von mehreren dynamischen Bereichen umrandet wird. So konnte ein neuartiger intelligenter Code entwickelt werden, der auf drei sensorischen Farben basiert, dennoch für weitere sensorische Farben erweitert werden kann und zentral positioniert einen DataMatrix Code, samt Fehlerkorrektur, für statische Informationen, beinhaltet. Die sensorischen Felder sind redundant an verschiedenen Stellen des Codes angebracht, sodass eine Auswertung nicht von einem – möglicherweise beschädigten oder partiell kontaminierten – Bereich abhängt, sondern sich auf mehrere, hier 12 Felder je 4 dynamische Bereiche stützen kann. Der intelligente Code weist ausschließlich quadratische Formen und für die dynamischen Bereiche besonders diskrete Kanten auf, so dass der Code sowie die dynamischen Bereiche beim Ausleseprozess schnell und sicher zu identifizieren sind. Die experimentellen Untersuchungen haben sowohl die uneingeschränkte Druckbarkeit des Codes, selbst mit dem eingesetzten Drucksystem, bestätigt. Aber auch die eindeutige Lesbarkeit des Codes wurde experimentell nachgewiesen. Dabei konnten zudem die graduellen Unterschiede der dynamischen Bereiche aufgrund unterschiedlicher Intensität bzw. Dauer der Kontamination klar differenziert werden.

#### 6) DRUCKTECHNISCHE UMSETZUNG UND AUSGABE EINES INTELLIGENTEN CODES

Die für das Inkjetsystem entwickelten sensorischen Tinten (*Kapitel 4.7*) samt intelligentem Code, konnten erfolgreich auf ein dafür identifiziertes Haftpapier (*Kapitel 3.1.2*) verdruckt werden. – Auch, wenn der Zugriff auf den Mikrokontroller und auf die softwareseitige Steuerungseinheit des verwendeten Drucksystems verwehrt war. So konnte der Laborumgebung, insbesondere dem in sich geschlossenen Drucksystem geschuldet, nicht dem Prozess der Droplet Formation (Tropfenbildung) mittels Bildgebungsverfahren oder der Variation der Ausstoßgeschwindigkeit, nachgegangen werden. So wäre eine Analyse, insbesondere in Bezug auf die Droplet-Eigenschaften (bspw. Bildung von Satellitentropfen) in Abhängigkeit der Ausstoßfrequenz ein untersuchungswerter Faktor bei der weiteren Optimierung der sensorischen Tinten. Des Weiteren ist die chemische Beschaffenheit des Substrats als ein unbekannter Faktor anzusehen, da vor allem der Einsatz eines Primers für die Verbesserung der Haftung und Absorption von Inkjetintinen einen negativen Einfluss auf die Reaktionsfähigkeit der sensorischen Tinten hat. Bereits beim Verdrucken der sensorischen Tinten auf ein Substrat kann die Eingangsfarbe der sensorischen Farbe verändert und gar die Funktionalität beeinträchtigt werden.

## 7) ERKENNUNGSPROZESS EINES INTELLIGENTEN CODES

Die Softwarearchitektur (*Kapitel 6*) basiert auf den zugrundeliegenden Grundlagen der Bildverarbeitung, Computer Vision (*Kapitel 6.1*) und der eingesetzten Algorithmen, die in den Erkennungsprozess des Intelligenten Codes (*Kapitel 6.3*) einfließen. Die für den Erkennungsprozess geschriebene Programmierung basiert dabei auf der Skriptsprache JavaScript, die auf der ihr zugeschnittenen Computer Vision (OpenCV.js) Programmbibliothek basiert. Vor allem die Auswahl und Anpassung eines Algorithmus war ein iterativer Prozess, der mit der Entwicklung des intelligenten Codes einherging. Bereits bei der Auswahl eines geeigneten Algorithmus wurden kleine Prototypen des intelligenten Codes zur Evaluation der Funktionalität eingesetzt. Hierbei stellte sich der Algorithmus von Suzuki und Abe (1985) zur Kantenerkennung – zusammen mit dem intelligenten Code – als überaus robust dar, jedoch war dieser, wie auch die OpenCV.js Programmbibliothek, nicht ausreichend dokumentiert. Hinzuzufügen ist, dass die OpenCV Programmbibliothek zum einen plattformunabhängig und zum anderen umfangreicher als vergleichbare CV Programmbibliotheken ist und kompakt alle wesentlichen Funktionen und Algorithmen für die konzeptionierte Softwarearchitektur aufweist.

## 8) BESEITIGUNG VON STÖRPARAMETERN

Es wurden verschiedene assistierende Algorithmen entwickelt, die mögliche Störfaktoren beseitigen (*Kapitel 6.6*) und die eventuell einen Einfluss auf die spätere Analyse und Auswertung des intelligenten Codes haben könnten. Einem Teil der möglichen Störparameter konnte durch eine Farbanpassung, einen Colormanagementprozess, eine Helligkeits- und Kontrastanpassung und einen Umgebungslicht-Assistenten, entgegengewirkt werden. Um unscharfe und verwackelte Aufnahmen zu vermeiden wurde zusätzlich eine Blur detection (Unschärferkennung) programmiert. Hinzuweisen ist jedoch darauf, dass die PyCMS Programmbibliothek zum Zeitpunkt der Abfassung dieser Dissertation ebenfalls nicht ausreichend dokumentiert war, wobei sich eine einfache ICC-Profil Implementierung in PyCMS als überaus schwierig herausstellte, da ältere ICC-Profil-Spezifikationen (ICC v4; ICC.1:2010) unterstützt und neueste Spezifikationen (ICC v5; ICC.2:2019) nicht durch die Programmbibliothek unterstützt wurden. Außerdem steht ein CMS lediglich für eine eingeschränkte Anzahl an Smartphones zur Verfügung und müsste um eine breite Verfügbarkeit zu schaffen, stetig ergänzt werden.

## 9) SEPARIERUNG DES INTELLIGENTEN CODES

Nach einer entsprechenden Beseitigung der Störparameter wird der intelligente Code in seine Bestandteile separiert (*Kapitel 6.4*), damit die jeweiligen Codebestandteile für eine nachfolgende Auswertung zur Verfügung stehen. Der Vorgang der Separierung wurde durch eine Programmierung in Python gelöst, da die anschließende Analyse und Auswertung auf eben dieser Programmiersprache umgesetzt wird. Außerdem erwies sich die OpenCV Programmbibliothek für Python umfangreich und als sehr gut dokumentiert.

## 10) ANALYSE UND AUSWERTUNG SENSORISCHER FARBINFORMATIONEN

Die Farbinformationen der zuvor separierten dynamischen Bereiche konnten berechnet und ausgewertet werden. Diese Auswerteeinheit (*Kapitel 6.5*) basiert dabei auf mehreren Algorithmen und Operatoren, die in den jeweiligen Grundlagen vorgestellt wurden (*Kapitel 6.1*). Dabei wurde der DataMatrix Code im statischen Bereich dekodiert und die jeweiligen Farbinformationen aus den dynamischen Bereichen entnommen und über ein Berechnungsverfahren innerhalb der Auswerteeinheit für den Benutzer visuell



dargestellt. Ein Risikobereich, der die Robustheit der Auswerteeinheit beeinflusst, ist die der Konturerkennung, welche in seltenen Fällen einige sensorische Felder nicht erkennt und eine marginale Fehleranfälligkeit aufweist. Diese Fehleranfälligkeit ist unter anderem auf Verzerrungen der Kameras zurückzuführen, die nicht durch Kalibrierung korrigiert wurden und Aufnahmen von quadratischen Formen rundlich erscheinen lassen. Diese Konturen sind somit für die programmierte shape detection, strenggenommen keine quadratischen Formen mehr und werden nicht erkannt.

## 11) BEWERTUNG DER KOMMUNIKATIONS- UND INTERNETTECHNOLOGIEN

Die Bewertung der sicherheitstechnischen Anforderungen der eingesetzten Kommunikationstechnologien (*Kapitel 6.7*) ist das Ziel dieses Teilschritts. Hierbei wurden die Grundlagen der Internettechnologie und die Themenfelder der Datensicherheit und Verschlüsselung vorgestellt und geben gleichzeitig einen Einblick in das Internet der Dinge. Die Grenzen und Risiken des Gesamtsystems werden nachfolgend behandelt und dienen der sicherheitstechnischen Validierung.

### 7.2. Grenzen

Die möglichen Grenzen des entwickelten Gesamtsystems sind der Schwerpunkt des vorliegenden Kapitels. Die Grenzen des gesamten Systems und seine Komponenten ergeben sich aus dem Anspruch an Forschung und Entwicklung sowie den gesetzten Zielen. Es sollte gezeigt werden, dass mithilfe sensorischer Farben in codierter Form Mehrfachkontaminationen eines Packgutes nachgewiesen und qualifiziert ausgewertet sowie deren Historie aufgezeigt werden können. Hierzu waren die damit verbundenen mittelbaren und unmittelbaren Schritte und Probleme zu identifizieren, in ein zu entwickelndes Gesamtsystem einzuordnen und im Labormaßstab experimentell zu lösen. Dies konnte lückenlos erfüllt werden.

Allerdings ergeben sich über den Labormaßstab hinaus Grenzen unterschiedlicher Art:

Zwar sind einige wenige funktionelle Druckfarben (sensorische Farben) für den Sieb- und Flexodruck mit irreversibel reagierenden Farbstoffen erhältlich. Sensorische Farben, die die sehr hohen Ansprüche an Inkjettinten erfüllen, müssen jedoch entwickelt werden. Ferner setzt der gewählte Labormaßstab Grenzen. Hier wurde mit drei irreversiblen Farbstoffen experimentiert. Bei mehr als drei Farben (plus Schwarz) ist die Grenze der üblichen, auch industriellen 4C Inkjet Drucksysteme erreicht. Allerdings bieten industrielle Inkjetsysteme gegenüber den verwendeten Drucksystemen (*Kapitel 3.3*) den Vorteil, dass jeder Druckkopf einzeln angesteuert werden kann. Auf diese Weise können reinfarbige Sensoren gedruckt werden. Bei „consumer“ Drucksystemen bestimmt stets ein Farbmanagementprofil die Ansteuerung der Druckköpfe und erzeugt Mischfarben. Auch bei der gezielten Auswahl von Farbprofilen kann dieser unerwünschte Effekt zwar fast, aber nicht ganz unterbunden werden. Die im Labormaßstab gewählten Hardwarekonfigurationen sind für eine darüberhinausgehende Anwendung kaum ausreichend. Dieser Engpass wäre durch leistungsfähigere Geräte leicht zu kompensieren. Bei den Smartphones (Client) bestehen heute noch Grenzen. Zum einen steht nicht bei allen Smartphones ein ICC-Farbprofil der Kamera zur Verfügung und zum anderen operieren die meisten Smartphones heute noch mit einem begrenzten Farbraum (sRGB). Jedoch wird inzwischen bei Smartphones der höheren Preiskategorie der ICC-Farbraum den Bildern angehängt. Möglicherweise wird dies auch bei einfachen Smartphones übernommen, so dass künftig für jedes Kamerabild die neutralen Farben leicht errechnet werden können. Als Vorstufe für den Auswertungsprozess der dynamischen Bereiche des intelligenten Codes

wurden exemplarisch Verfahren zur Kompensation unerwünschter Störeffekte gezeigt: Korrektur der auftretenden Störeffekte des umgebenden Lichts (Kapitel 6.6.2) und Korrektur des auf den intelligenten Code einwirkenden Lichts und die damit einhergehende Beeinflussung der Substratfarbe (*Kapitel 6.6.1*). Je nach Anwendung gibt es hier keine Grenzen für die Beseitigung von umgebungs- und anwenderbedingten Störeffekten und Handhabungsfehlern.

### 7.3. Risiken

In diesem Abschnitt werden mögliche Risiken gegenüber der entwickelten Webapplikation und ebenfalls ausgehend von sensorischen Farben sowie sensorischen Tinten aufgezeigt. Wie bereits in *Kapitel 6.7.3* erläutert, sind bereits einige präventive Maßnahmen zum Schutze der Serverarchitektur vorgenommen worden. Ein wesentlicher Aspekt war es dabei, die Selbstbestimmung der Benutzer zu wahren, die vor einem Mediengerätzugriff ihre Zugriffserlaubnis erteilen müssen und können somit den Zugriff auf bspw. eine Kamera bei Bedarf deaktivieren. Der zertifikatbasierte Verbindungsaufbau basiert dabei auf dem Netzwerkprotokoll HTTP(S) / SSL. Hierbei beginnt der clientseitige Browser einen SSL Handshake-Prozess zu initiieren, bei dem die gesicherte (verschlüsselte) Webanwendung unter Verwendung des HTTPS-Protokolls, aufgerufen wird. Eine weitere Maßnahme ist der Schutz vor Cross Site-Request-Forgery, bei der eine Anfragefälschung seitens Angreifern angestrebt wird. Diesen gefälschten Anfragen wird serverseitig jedoch über so bezeichnete Token entgegengewirkt, die pro Session übergeben werden. Dennoch sind kein System und keine Webanwendung vollständig sicher vor Angriffen. Wie bereits in *Kapitel 3.8* erläutert, ist ein wesentlicher Nachteil der Netzwerktopologien mit Sternstruktur, dass bei einem Ausfall des zentralen Servers auch die gesamte Kommunikation und der Zugriff auf den Server unterbrochen wird. Dieser Unterbrechung kann jedoch unter anderem mit einem bereitstehenden und gespiegelten Server (Backup) entgegengewirkt werden, indem die zyklisch wiederkehrende Verfügbarkeit des Servers überprüft wird.

Dennoch führen DDoS-Angriffe (Distributed-Denial-of-Service-Angriffe) auf Server vermehrt zu Ausfällen. Hierbei werden Botnetze eingesetzt, die aus vielen verteilten und größtenteils gekaperten Privatrechnern (als Zombie-Rechner bezeichnet) bestehen und über Kontrollserver auf ihr Angriffsziel gesteuert werden. Dabei wird der anzugreifende Zielservers durch massenweise Anfragen, bei dem sehr viel Traffic (Zugriffe) entsteht, überlastet und fällt aus. So können bereits im Untergrund DDoS-Dienste seitens Angreifern gebucht werden, die gezielt Angriffe ausführen. Gegenmaßnahmen können dabei die Erhöhung der Serverbandbreite oder das Blockieren identifizierter DDoS-Traffics sein. Darüber hinaus sind bei der Entwicklung von sensorischen Farben und auch sensorischen Tinten auf mögliche Risiken bei der Materialentwicklung zu achten. So wurde bereits im *Kapitel 4.6* auf toxikologische und auch ökotoxikologische Eigenschaften verschiedener Materialien, insbesondere chemischer Komponenten, hingewiesen. Dabei wurden verschiedene Stoffe aufgezeigt, die gerade im Lebensmittel- oder Gesundheitsbereich ungeeignet sind. Dahingehend war es ein weiteres Ziel, irreversible sensorische Farben zu entwickeln, die auf umweltgefährliche Stoffe verzichten. Somit konnten abschließend mögliche Risiken aufgezeigt und ihre Gegenmaßnahmen validiert werden.

## Kapitel 8

# Ausblick

Mit den dargelegten Forschungsergebnissen konnte gezeigt werden, dass mithilfe sensorischer Farben in codierter Form Mehrfachkontaminationen eines Packgutes nachgewiesen und qualifiziert ausgewertet sowie deren Historie aufgezeigt werden können. An dieser Stelle wird weiterer Forschungsbedarf in diesem Kontext identifiziert. Im Stand der Forschung (*Kapitel 2*) konnte bereits ein Forschungsbedarf an irreversiblen sensorischen Farben aufgezeigt werden. So sind überwiegend reversible statt irreversible Materialien in diesem Forschungsbereich aufzufinden. Wird zusätzlich das zweite Kriterium der inkjetfähigen Tinte hinzugezogen, verringert sich die Liste (Tabelle 3, grau hinterlegt) der identifizierten sensorischen Farben erheblich. Anders verhält es sich bei den Siebdruckfarben, bei denen eine große Auswahl an reversiblen, aber auch nur eine geringe Anzahl an irreversiblen sensorischen Farben aufzufinden ist. Insgesamt zeigt sich ein Forschungs- und Entwicklungsbedarf an irreversiblen thermochromen, hydrochromen und weiteren Varianten sensorischer Farben, wobei erstere für die Reaktivität im gewünschten Temperaturbereich  $< 50\text{ °C}$  kaum, bis gar nicht identifiziert wurden. Von Interesse wäre es zudem, wenn die Entwicklung von inkjetfähigen sensorischen Farben vorangetrieben werden würde, da diese als sensorische Tinten oder auch als so bezeichnete „smart coatings“ für verschiedene Anwendungsfelder eingesetzt werden können. Besonders sollten Risiken bei der Materialentwicklung insbesondere in Bezug auf toxikologische oder auch ökotoxikologische Eigenschaften vermieden und Entwicklungen im Sinne einer nachhaltigen Chemie angestrebt werden.

Ein besonders attraktiver Forschungsbereich stellen Biosensoren dar, in dem insbesondere Halochrome als Detektoren, gezielt in Kombination mit Biorezeptoren, für einen Nachweis des zu bestimmenden Stoffes (Analyt) durch biochemische Reaktionen eingesetzt werden könnten. So könnten biologische Komponenten, wie bspw. isolierte Enzyme, Antikörper oder Zellen, als Biorezeptoren eingesetzt werden. Durch die gezielte Kombination der Biorezeptoren mit den Halochromen können diese immobilisiert werden, um somit ein Analyt durch biochemische Reaktionen zu identifizieren und durch einen Stimulus bedingte Farbänderung, in Form eines Messsignals, anzuzeigen. Ein weiterer Einsatzbereich für sensorischen Tinten eröffnet sich beim Sicherheitsdruck, bei dem sie als Sicherheitsmerkmale ihre Verwendung finden können. Bspw. könnten durch den Einsatz von Thermochromen, thermische Einflüsse, die bei räuberischen Bohrungen in Goldbarren entstehen, detektiert und somit farblich nachgewiesen werden. Auch bei der Optimierung der hier entwickelten Leseapplikation können die eingesetzten Algorithmen für das Erzielen präziserer Ergebnisse fortlaufend optimiert und verschiedene Szenarien von bestimmten Einflussfaktoren, wie unter anderem Licht- und Schatteneinwirkungen, und realitätsnahe Funktionstests untersucht werden. Zusätzlich kann die Art der Ergebnisdarstellung verbessert, weitere statistische Verfahren, oder bei Bedarf Elemente künstlicher Intelligenz implementiert und die Resultate benutzerfreundlich dargestellt werden. So könnten weitere Entwicklungen von Webanwendungen für das Web- Virtual Reality (WebVR) und Augmented Reality (WebAR) verfolgt werden, da in diesem Bereich noch kaum Projekte umgesetzt werden. Die aufgeführten möglichen Forschungsbereiche zeigen einen vielfältigen und differenzierten Einsatz der sensorischen Tinten und die der Leseapplikation. Dadurch zeigt sich die breite, komplexe und fassettenreiche Anwendbarkeit dieses Forschungsfeldes.

# Anhang

## A) Geräteeigenschaften und Geräteeinstellungen

### Rheometer Physica MCR 101

Bezeichnung	Werte
Antriebsprinzip	Permanentmagnet-Synchronantrieb
Minimales Drehmoment	0.5 $\mu\text{Nm}$
Maximales Drehmoment	125 mNm
Drehmomentauflösung	0.002 $\mu\text{Nm}$
Drehmomentgenauigkeit	max. (0.2 $\mu\text{Nm}$ ; 0.5%)
Maximaler Anstieg des Drehmoments	1500 Nm/s
Geschwindigkeitsbereich (CSS)	$10^{-6} \text{ min}^{-1}$ - $3000 \text{ min}^{-1}$

Tabella 28 Geräteeigenschaften Rheometer

### Blasendrucktensiometer Sita pro line t15

Bezeichnung	Werte
Messbereich	15 - 100 mN/m (dyn/cm)
Auflösung	0,1 mN/m (dyn/cm)
Regelbereich	15 ms - 20.000 ms
Auflösung	1 ms
Messbereich	0 - 100 °C
Auflösung	0,1 K
Kapillaren	Typ I und II PEEK

Tabella 29 Geräteeigenschaften Blasendrucktensiometer

### Spektralradiometer UVpad E

Bezeichnung	Werte
Spektralbereich	240 - 480 nm $\pm$ 5 nm
Spektrale Bandbreite	2 nm
Messbereich, typisch	2 - 5000 mW/cm <sup>2</sup>
Auslösung	0,01 mW
Sensorkopf-Abmessung, Typ 2	$\varnothing$ 40 mm, h 25 mm

Tabella 30 Geräteeigenschaften Spektralradiometer

### Spektral-Densitometer SpectroDens

Bezeichnung	Werte
Polarisationsfilter	aus
Lichtart	D50, 2° Normalbeobachter
Durchmesser der Messblende	3 mm

Tabella 31 Geräteeinstellungen Spektral-Densitometer

### Rasterelektronenmikroskop (REM)

Bezeichnung	Werte
Modellreihe	FEI Quanta 450
Detektorart	BSED (backscattered electron detector)
Auflösung	< 1,5 - 2,0 nm

Tabella 32 Geräteeinstellungen Spektral-Densitometer

### Konfokales 3D Laserscanning Mikroskop VK-X 100K + VK-X 150 K

Bezeichnung	Werte
x10 Messoptik, W/D (mm)	16.5
x20 Messoptik, W/D (mm)	3.1
x50 Messoptik, W/D (mm)	0.54
x100 Messoptik, W/D (mm)	0.3

*Tabelle 33 Geräteeigenschaften Konfokales 3D Laserscanning Mikroskop*

B) Abschätzung der Reynolds- und Ohnesorge-Zahl

### Hydrochrome sensorische Tinte

#### Gegeben

Geschwindigkeit Tropfen:  $v \approx 5 \frac{m}{s}$  (Okumura & Takahashi, 2007, S. 317)

Düsendurchmesser:  $d \approx 61 \mu m$  (Kapitel 4.4.3)

Dichte (Ethanol):  $\rho \approx 807 \frac{Kg}{m^3}$

Viskosität:  $\eta \approx 1,89 \text{ mPa}\cdot\text{s}$

Oberflächenspannung:  $\sigma \approx 37 \frac{mN}{m}$

#### Formeln

$$Re = \frac{\rho \cdot v \cdot d}{\eta} \quad Oh = \frac{\eta}{\sqrt{\sigma \cdot \rho \cdot d}}$$

#### Berechnung

$$Re = \frac{807 \frac{Kg}{m^3} \cdot 5,0 \frac{m}{s} \cdot 0,000061 m}{0,00189 Pa \cdot s} = 130,2$$

$$Oh = \frac{0,00189 Pa \cdot s}{\sqrt{0,037 \frac{N}{m} \cdot 807 \frac{Kg}{m^3} \cdot 0,000061 m}} = 0,044$$

### Photochrome sensorische Tinte

#### Gegeben

Geschwindigkeit Tropfen:  $v \approx 5 \frac{m}{s}$  (Okumura & Takahashi, 2007, S. 317)

Düsendurchmesser:  $d \approx 61 \mu m$  (Kapitel 4.4.3)

Dichte (Wasser):  $\rho \approx 1000 \frac{Kg}{m^3}$

Viskosität:  $\eta \approx 4,15 \text{ mPa}\cdot\text{s}$

Oberflächenspannung:  $\sigma \approx 46 \frac{mN}{m}$

#### Formeln

$$Re = \frac{\rho \cdot v \cdot d}{\eta} \quad Oh = \frac{\eta}{\sqrt{\sigma \cdot \rho \cdot d}}$$

#### Berechnung

$$Re = \frac{1000 \frac{Kg}{m^3} \cdot 5,0 \frac{m}{s} \cdot 0,000061 m}{0,00415 Pa \cdot s} = 73,5$$

$$Oh = \frac{0,00415 Pa \cdot s}{\sqrt{0,037 \frac{N}{m} \cdot 1000 \frac{Kg}{m^3} \cdot 0,000061 m}} = 0,087$$

## Halochrome sensorische Tinte

### Gegeben

Geschwindigkeit Tropfen:  $v \approx 5 \frac{m}{s}$  (Okumura & Takahashi, 2007, S. 317)

Düsendurchmesser:  $d \approx 61 \mu m$  (Kapitel 4.4.3)

Dichte (Ethanol):  $\rho \approx 807 \frac{Kg}{m^3}$

Viskosität:  $\eta \approx 1,38 \text{ mPa}\cdot\text{s}$

Oberflächenspannung:  $\sigma \approx 22 \frac{mN}{m}$

### Formeln

$$Re = \frac{\rho \cdot v \cdot d}{\eta} \quad Oh = \frac{\eta}{\sqrt{\sigma \cdot \rho \cdot d}}$$

### Berechnung

$$Re = \frac{807 \frac{Kg}{m^3} \cdot 5,0 \frac{m}{s} \cdot 0,000061 m}{0,00138 Pa \cdot s} = 178,3$$

$$Oh = \frac{0,00138 Pa \cdot s}{\sqrt{0,022 \frac{N}{m} \cdot 807 \frac{Kg}{m^3} \cdot 0,000061 m}} = 0,042$$

### C) Programmstruktur Erkennungsprozess

```
//Das Design für den Kamerazugriff ist nach dem Vorbild von Kasper Kamperman mit der
//Bezeichnung "Mobile (and Desktop) Camera App Template".
//Quelle: https://github.com/kasperkamperman/MobileCameraTemplate

var video;
var checkButton;
var toggleFullScreenButton;
var goToScans;
var amountOfCameras = 0;
var currentCameraIndex = 0;
var currentCameraId = "";
var currentFacingMode = 'environment';

function opencvIsReady() {
  initCameraUI();
  initCameraStream();
}
document.addEventListener('DOMContentLoaded', function(event) {
  // do some WebRTC checks before creating the interface
  DetectRTC.load(function() {
    // do some checks
    if (DetectRTC.isWebRTCSupported == false) {
      alert('Please use Chrome, Firefox, iOS 11, Android 5 or higher, Safari 11 or
      higher,');
    } else {
      if (DetectRTC.hasWebcam == false) {
        alert('Please install an external webcam device.');
      } else {
        amountOfCameras = DetectRTC.videoInputDevices.length;
      }
    }
    // debug information
    console.log(
      'RTC Debug info: ' +
      '\n OS: ' +
      DetectRTC.osName +
      ' ' +
      DetectRTC.osVersion +
      '\n browser: ' +
      DetectRTC.browser.fullVersion +
      ' ' +
      DetectRTC.browser.name +
      '\n is Mobile Device: ' +
      DetectRTC.isMobileDevice +
      '\n has webcam: ' +
      DetectRTC.hasWebcam +
      '\n has permission: ' +
      DetectRTC.isWebsiteHasWebcamPermission +
      '\n getUserMedia Support: ' +
      DetectRTC.isUserMediaSupported +
      '\n isWebRTC Supported: ' +
      DetectRTC.isWebRTCSupported +
      '\n WebAudio Supported: ' +
      DetectRTC.isAudioContextSupported
    );
  });
});

function initCameraUI() {
  video = document.getElementById('video');
  checkButton = document.getElementById('checkButton');
  toggleFullScreenButton = document.getElementById('toggleFullScreenButton');
  goToScans = document.getElementById('goToScans');

  checkButton.setAttribute('status', 'invisible');

  goToScans.addEventListener('click', function() {
    window.location.href = "/scans"
  });

  // -- fullscreen part
```



```

function fullScreenChange() {
    if (screenfull.isFullscreen) {
        toggleFullScreenButton.setAttribute('aria-pressed', true);
    } else {
        toggleFullScreenButton.setAttribute('aria-pressed', false);
    }
}

if (screenfull.isEnabled) {
    screenfull.on('change', fullScreenChange);

    toggleFullScreenButton.style.display = 'block';

    // set init values
    fullScreenChange();

    toggleFullScreenButton.addEventListener('click', function() {
        screenfull.toggle(document.getElementById('container')).then(function() {
            console.log('Fullscreen mode: ' + (screenfull.isFullscreen ? 'enabled' :
                'disabled'));
        });
    });
} else {
    console.log("iOS doesn't support fullscreen (yet)");
}

// -- switch camera part
/*if (amountOfCameras > 1) {
    switchCameraButton.style.display = 'block';

    switchCameraButton.addEventListener('click', function() {
        currentCameraIndex++;
        //if (currentFacingMode === 'environment') currentFacingMode = 'user';
        //else currentFacingMode = 'environment';
        //alert(currentCameraIndex % amountOfCameras);
        navigator.mediaDevices.enumerateDevices()
            .then(function(devices) {
                currentCameraId = devices[currentCameraIndex % amountOfCameras].deviceId;
            });
        initCameraStream();
    });
}*/

// Listen for orientation changes to make sure buttons stay at the side of the
// physical (and virtual) buttons (opposite of camera) most of the layout change is done
// by CSS media queries
window.addEventListener('orientationchange', function() {
    // iOS doesn't have screen.orientation, so fallback to window.orientation.
    // screen.orientation will
    if (screen.orientation) angle = screen.orientation.angle;
    else angle = window.orientation;

    var guiControls = document.getElementById('gui_controls').classList;
    var vidContainer = document.getElementById('vid_container').classList;

    if (angle == 270 || angle == -90) {
        guiControls.add('left');
        vidContainer.add('left');
    } else {
        if (guiControls.contains('left')) guiControls.remove('left');
        if (vidContainer.contains('left')) vidContainer.remove('left');
    }
    //0 portrait-primary
    //180 portrait-secondary device is down under
    //90 landscape-primary buttons at the right
    //270 landscape-secondary buttons at the left
    },
    false,
);
}

```

```

// opencv variables
let scanning = true;
let videoWidth = 720;
let videoHeight = 0;
let width = 320;
let height = 0;
let streaming = false;
let stream = null;
let vc = null;
let scancounter = 0;

let src = null;
let dstC1 = null;
let dstC4 = null;

function initCameraStream() {
  // stop any active streams in the window
  if (window.stream) {
    window.stream.getTracks().forEach(function(track) {
      track.stop();
    });
  }

  // we ask for a square resolution, it will be cropped on top (landscape)
  // or cropped at the sides (landscape)
  var size = 480;

  var constraints = {
    audio: false,
    video: {
      frameRate: { ideal: 15 },
      width: { min: 320, ideal: 480, max: size },
      height: { min: 180, ideal: 480, max: size },
      //width: { ideal: size },
      //height: { ideal: size },
      //width: { min: 1024, ideal: window.innerWidth, max: 1920 },
      //height: { min: 776, ideal: window.innerHeight, max: 1080 },
      facingMode: currentFacingMode,
      deviceId: currentCameraId
    },
  };
  navigator.mediaDevices
    .getUserMedia(constraints)
    .then(handleSuccess)
    .catch(handleError);
  function handleSuccess(stream) {
    window.stream = stream; // make stream available to browser console
    video.srcObject = stream;
    video.play();

    /*if (constraints.video.facingMode) {
      if (constraints.video.facingMode === 'environment') {
        switchCameraButton.setAttribute('aria-pressed', true);
      } else {
        switchCameraButton.setAttribute('aria-pressed', false);
      }
    }*/
    video.addEventListener("canplay", function (ev) {
      if (!streaming) {
        videoHeight = video.videoHeight / (video.videoWidth / videoWidth);
        height = video.videoHeight / (video.videoWidth / width);
        video.setAttribute("width", videoWidth);
        video.setAttribute("height", videoHeight);
        streaming = true;
        vc = new cv.VideoCapture(video);
      }
      startVideoProcessing();
    }, false);
    const track = window.stream.getVideoTracks()[0];
    const settings = track.getSettings();

```

```

    str = JSON.stringify(settings, null, 4);
    console.log('settings ' + str);
    //return navigator.mediaDevices.enumerateDevices();
}
function startVideoProcessing() {
    src = new cv.Mat(videoHeight, videoWidth, cv.CV_8UC4);
    dstC1 = new cv.Mat(height, width, cv.CV_8UC1);
    dstC4 = new cv.Mat(height, width, cv.CV_8UC4);
    requestAnimationFrame(processVideo);
}

function processVideo() {
    vc.read(src);
    cv.imshow("canvasOutput", readCode(src));
    requestAnimationFrame(processVideo);
}

function handleError(error) {
    console.log(error);
    if (error === 'PermissionDeniedError') {
        alert('Permission denied. Please refresh and give permission.');
```

```

    }
}
let lastEdgesCode;
let lastEdgesMarker0;
let lastEdgesMarker1;
let lastEdgesMarker2;
```

```

function readCode(src) {
    scancounter++;
    if (scancounter % 15 !== 0) {
        if (lastEdgesCode !== null && lastEdgesMarker0 !== null) {
            drawEdges(lastEdgesCode, new cv.Scalar(255, 0, 0, 255));
            drawEdges(lastEdgesMarker0, new cv.Scalar(0, 255, 0, 255));
            drawEdges(lastEdgesMarker1, new cv.Scalar(0, 255, 0, 255));
            drawEdges(lastEdgesMarker2, new cv.Scalar(0, 255, 0, 255));
        }
        return src;
    }
}
```

```
lastEdgesCode = lastEdgesMarker0 = lastEdgesMarker1 = lastEdgesMarker2 = null;
```

```
//checkbox.style.display = "none";
```

```
// blur check, stop checking if image is too blurry
```

```

let src_grey = new cv.Mat();
let dst_blur = new cv.Mat();
let men_blur = new cv.Mat();
let men0_blur = new cv.Mat();
cv.cvtColor(src, src_grey, cv.COLOR_RGB2GRAY, 0);
// You can try more different parameters
cv.Laplacian(src_grey, dst_blur, cv.CV_64F, 1, 1, 0, cv.BORDER_DEFAULT);
cv.meanStdDev(dst_blur, men0_blur, men_blur);
if(men_blur.data64F[0] > 10) {
    src_grey.delete();
    dst_blur.delete();
    men_blur.delete();
    men0_blur.delete();
    console.log("not blur");
} else {
    console.log("blur");
    src_grey.delete();
    dst_blur.delete();
    men_blur.delete();
    men0_blur.delete();
    return src;
}
let hierarchy = new cv.Mat();
let contours = new cv.MatVector();
let foundMarker = new cv.MatVector();
let resizedSrc = new cv.Mat();
let dstSize = new cv.Size(width, height);
```

```

cv.resize(src, resizedSrc, dstSize, 0, 0, cv.INTER_AREA);

cv.cvtColor(resizedSrc, dstC1, cv.COLOR_RGBA2GRAY);
cv.threshold(dstC1, dstC4, 120, 200, cv.THRESH_BINARY);
cv.findContours(dstC4, contours, hierarchy, 2, 2, {x: 0, y: 0});

let sortedContours = [];
// add all contours to new array
for (let i = 0; i < contours.size(); i++) {
    sortedContours.push(contours.get(i));
}
// sort all found contours by its contourArea from
// smallest to biggest by implementing own sorting function
sortedContours.sort(function(x, y) {
    if (cv.contourArea(x) < cv.contourArea(y)) return -1;
    if (cv.contourArea(x) > cv.contourArea(y)) return 1;
    return 0;
});
// get x biggest contours by slicing
let biggestContours = sortedContours.slice(sortedContours.length-8, sortedContours.length);

// iterate over all found biggest contours to find a possible code
// Note: starting at the end to go from biggest to smallest
for (let i = biggestContours.length-1; i >= 0; i--) {
    let approx = new cv.Mat();
    let cnt = biggestContours[i];
    cv.approxPolyDP(cnt, approx, 0.03 * cv.arcLength(cnt, true), true); //3
    let rect = cv.boundingRect(cnt);

    let ar = rect.width / rect.height;

    // if the contour has 4 rows and a width / height ratio
    // between 0.9 and 1.1 (~square) add it to the found markers
    if (approx.rows == 4 && ar >= 0.9 && ar <= 1.1 && rect.width <= 55) {
        foundMarker.push back(approx);
    }
    // if we found 3 marker continue
    if (foundMarker.size() == 3) {
        let cnt0 = foundMarker.get(0); // marker 0
        let cnt1 = foundMarker.get(1); // marker 1
        let cnt2 = foundMarker.get(2); // marker 2
        let allMarkerHull = new cv.Mat(); // smartcode

        // create a matrix manually from all the found coordinates
        let allMarkerMatrix = cv.matFromArray(12, 2, cv.CV_32S,
            [
                cnt0.data32S[0], cnt0.data32S[1], cnt0.data32S[2], cnt0.data32S[3],
                cnt0.data32S[4], cnt0.data32S[5], cnt0.data32S[6], cnt0.data32S[7],
                cnt1.data32S[0], cnt1.data32S[1], cnt1.data32S[2], cnt1.data32S[3],
                cnt1.data32S[4], cnt1.data32S[5], cnt1.data32S[6], cnt1.data32S[7],
                cnt2.data32S[0], cnt2.data32S[1], cnt2.data32S[2], cnt2.data32S[3],
                cnt2.data32S[4], cnt2.data32S[5], cnt2.data32S[6], cnt2.data32S[7]
            ]
        );

        // create a hull for all marker
        cv.convexHull(allMarkerMatrix, allMarkerHull, false, true);

        // get minarearectangle
        let minAreaHull = cv.minAreaRect(allMarkerHull);

        // get the ratio of the whole code
        let hullratio = minAreaHull.size.width / minAreaHull.size.height;

        // if the whole code is a rectangle continue
        if (hullratio >= 0.9 && hullratio <= 1.1) {

            // create a matrix manually for each marker to get the hulls
            let cnt0Matrix = cv.matFromArray(4, 2, cv.CV_32S, [cnt0.data32S[0],
                cnt0.data32S[1], cnt0.data32S[2], cnt0.data32S[3], cnt0.data32S[4],
                cnt0.data32S[5], cnt0.data32S[6], cnt0.data32S[7]]);
            let cnt1Matrix = cv.matFromArray(4, 2, cv.CV_32S, [cnt1.data32S[0],

```

```

cnt1.data32S[1], cnt1.data32S[2], cnt1.data32S[3], cnt1.data32S[4],
cnt1.data32S[5], cnt1.data32S[6], cnt1.data32S[7]);
let cnt2Matrix = cv.matFromArray(4, 2, cv.CV_32S, [cnt2.data32S[0],
cnt2.data32S[1], cnt2.data32S[2], cnt2.data32S[3], cnt2.data32S[4],
cnt2.data32S[5], cnt2.data32S[6], cnt2.data32S[7]]);

// the hulls
let cnt0Hull = new cv.Mat();
let cnt1Hull = new cv.Mat();
let cnt2Hull = new cv.Mat();

// get hulls from their matrices
cv.convexHull(cnt0Matrix, cnt0Hull, false, true);
cv.convexHull(cnt1Matrix, cnt1Hull, false, true);
cv.convexHull(cnt2Matrix, cnt2Hull, false, true);

// create the minarectangle
cnt0minAreaHull = cv.minAreaRect(cnt0Hull);
cnt1minAreaHull = cv.minAreaRect(cnt1Hull);
cnt2minAreaHull = cv.minAreaRect(cnt2Hull);

// use the point to get the edges
let edgesMarker0 = cv.RotatedRect.points(cnt0minAreaHull);
let edgesMarker1 = cv.RotatedRect.points(cnt1minAreaHull);
let edgesMarker2 = cv.RotatedRect.points(cnt2minAreaHull);

// get the coordinates of the corners of the smartcode
let edgesCode = cv.RotatedRect.points(minAreaHull);
//console.log("Code found!");

let finalCode = new cv.Mat();
let dsize = new cv.Size(300,300);

// since we were working with a downscaled image, we need to
// calculate the coordinates for the higher resolution
let srcMatrix = cv.matFromArray(4, 1, cv.CV_32FC2,
[
    edgesCode[1].x / width * videoWidth, edgesCode[1].y / height *
    videoHeight,
    edgesCode[2].x / width * videoWidth, edgesCode[2].y / height *
    videoHeight,
    edgesCode[0].x / width * videoWidth, edgesCode[0].y / height *
    videoHeight,
    edgesCode[3].x / width * videoWidth, edgesCode[3].y / height *
    videoHeight
]);

// get the smartcode and leave a small frame around the code to make it easier
// to detect the code later on
let dstMatrix = cv.matFromArray(4, 1, cv.CV_32FC2, [25, 25, 275, 25, 25, 275,
275, 275]);
let perspectiveTransform = cv.getPerspectiveTransform(srcMatrix, dstMatrix);

// turn the image so that the smartcode has the same orientation every time
cv.warpPerspective(src, finalCode, perspectiveTransform, dsize,
cv.INTER_LINEAR, cv.BORDER_CONSTANT, new cv.Scalar());

// show the found code to a hidden canvas to get it later to send it to the
// server
cv.imshow('foundCode', finalCode);

// to prevent the scanner from sending 100 of images per seconds, I check if
// scanning equals true
if (scanning == true) {
    scanning = false;
    console.log("hourglass");
    checkButton.setAttribute('status', 'loading');
    //checkButton.style.display = "block";
    sendCodeToServer();
}

```

```

// calculate new coordinates from old resolution to new resolution
for (let j = 0; j < 4; j++) {
    edgesCode[j].x = parseInt((edgesCode[j].x / width * videoWidth));
    edgesCode[j].y = parseInt((edgesCode[j].y / height * videoHeight));
    edgesMarker0[j].x = parseInt((edgesMarker0[j].x / width * videoWidth));
    edgesMarker0[j].y = parseInt((edgesMarker0[j].y / height * videoHeight));
    edgesMarker1[j].x = parseInt((edgesMarker1[j].x / width * videoWidth));
    edgesMarker1[j].y = parseInt((edgesMarker1[j].y / height * videoHeight));
    edgesMarker2[j].x = parseInt((edgesMarker2[j].x / width * videoWidth));
    edgesMarker2[j].y = parseInt((edgesMarker2[j].y / height * videoHeight));
}

// saving the coordinates to draw them in the frames that aren't calculated
lastEdgesCode = edgesCode;
lastEdgesMarker0 = edgesMarker0;
lastEdgesMarker1 = edgesMarker1;
lastEdgesMarker2 = edgesMarker2;

// draw rectangle around the hull / vertices
drawEdges(edgesCode, new cv.Scalar(255, 0, 0, 255));
drawEdges(edgesMarker0, new cv.Scalar(0, 255, 0, 255));
drawEdges(edgesMarker1, new cv.Scalar(0, 255, 0, 255));
drawEdges(edgesMarker2, new cv.Scalar(0, 255, 0, 255));

cnt0Matrix.delete();
cnt1Matrix.delete();
cnt2Matrix.delete();
cnt0Hull.delete();
cnt1Hull.delete();
cnt2Hull.delete();

srcMatrix.delete();
dstMatrix.delete();
perspectiveTransform.delete();
finalCode.delete();
}
allMarkerHull.delete();
break;
}
cnt.delete();
approx.delete();
}
hierarchy.delete();
contours.delete();
foundMarker.delete();
resizedSrc.delete();
return src;
}

// draws lines around a given set of edges
function drawEdges(edges, color) {
    for (let j = 0; j < 4; j++) {
        cv.line(src, edges[j], edges[(j + 1) % 4], color, 2, cv.LINE_AA, 0);
    }
}

function resetScanning() {
    //console.log("Reset");
    checkButton.setAttribute('status', 'invisible');
    scanning = true;
}

```

## D) Programmstruktur Separierung

```
import json
import cv2
import numpy as np
import argparse
import os
import sys
import subprocess
from pathlib import Path
from pylibdmtx.pylibdmtx import decode
from db_model import CodeModel, Color, Point
from db import insert into database, execute
from colordifference import calculate_latest_differece

parser = argparse.ArgumentParser()
parser.add_argument("-p", required=True)
args = parser.parse_args()
scandirpath = os.path.dirname(args.p)

if not os.path.exists(args.p):
    raise Exception("File not found!")

# Platzhalter
a = 0
b = 0
c = 0
d = 0
e = 0
f = 0
g = 0
h = 0
b_gr = 0
h_gr = 0
checksumme = 0
rotationsachse = 0 # gehoert zum Auswertemodul

# Adaptiver Threshold
schwellenwert = 5 # gehoert zum beiden Modulen

# Speicher Arrays
no_duplicates = []
marker = []
rotmarker = []

# Speicher Arrays Marker
OL = []
OR = []
UR = []

M1OL = []
M1OR = []
M1UL = []
M1UR = []

M2OL = []
M2OR = []
M2UL = []
M2UR = []

M3OL = []
M3OR = []
M3UL = []
M3UR = []

# Alle smarte Bereiche
sc_B = []
sc_G = []
sc_R = []

sm_B = []
sm_G = []
sm_R = []
```

```

sy_B = []
sy_G = []
sy_R = []

# smartcodeD
s1 = []
s1_B = []
s1_G = []
s1_R = []
s2 = []
s2_B = []
s2_G = []
s2_R = []
s3 = []
s3_B = []
s3_G = []
s3_R = []
s4 = []
s4_B = []
s4_G = []
s4_R = []

#####
# Auswertemodul
#####

# Ab hier beginnt die Vorauswertung
img = cv2.imread(args.p)
# cv2.imshow("Captured Original Image", img)
h, b = img.shape[:2]
org = img.copy()

# Parameter zum rotieren des Bildes
mitte = (h / 2, b / 2)
faktor = 1.0

# Start Farbkorrektur
img_corr = img.copy()
input_corr = img.copy()
d65 = np.load('/home/pi/django/codereader/reader/d65.npy') #/home/pi/django/codereader/reader/

# Masking parameter only background-white in upper and lower range
lower_white_corr = np.array([130, 130, 130], dtype=np.uint8)#100
upper_white_corr = np.array([255, 255, 255], dtype=np.uint8)#255
corrinp_corr = cv2.cvtColor(input_corr, cv2.COLOR_BGR2GRAY)
corrinp_corr = cv2.adaptiveThreshold(corrinp_corr, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY, 5, 9)

# Masking parameter only code black
img_lower_white_corr = np.array([0, 0, 0], dtype=np.uint8)#0
img_upper_white_corr = np.array([130, 130, 130], dtype=np.uint8)#130#180
graucorr_corr = cv2.cvtColor(img_corr, cv2.COLOR_BGR2GRAY)
graucorr_corr = cv2.adaptiveThreshold(graucorr_corr, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY, 5, 9)

#Target: Extract bg color
masked_scan_corr = cv2.bitwise_and(input_corr, input_corr, mask=graucorr_corr)

mittw_corr = np.mean(masked_scan_corr)#5.8 schwarz / 70.75 maske
mittw_corr = int(mittw_corr)
if mittw_corr <= 80:
    sys.exit('Kamera nicht geeignet: Farbfehler')

bg_code_corr = np.delete(masked_scan_corr, np.argwhere(masked_scan_corr == [0,0,0]), axis=0)
bg_code_corr = bg_code_corr[:3]

#convert the images from the RGB/BGR to Lab color space
scan_corr = cv2.cvtColor(bg_code_corr, cv2.COLOR_BGR2LAB).astype("float32")/255#bg_color
reference_corr = cv2.cvtColor(d65, cv2.COLOR_BGR2LAB).astype("float32")/255#d65
img_corr = cv2.cvtColor(img_corr, cv2.COLOR_BGR2LAB).astype("float32")/255#original_image

```



```

#Split Lab color space
(l_s_corr, a_s_corr, b_s_corr) = cv2.split(scan_corr)#masked_scan
(l_r_corr, a_r_corr, b_r_corr) = cv2.split(reference_corr)#reference
(l_o_corr, a_o_corr, b_o_corr) = cv2.split(img_corr)#original_scan

#standard deviation
std_ls_corr = np.std(l_s_corr)
std_lr_corr = np.std(l_r_corr)

#Color difference between both images
diff_l_corr = np.mean(l_s_corr) - np.mean(l_r_corr)
diff_a_corr = np.mean(a_s_corr) - np.mean(a_r_corr)
diff_b_corr = np.mean(b_s_corr) - np.mean(b_r_corr)

#subtract difference
l_o_corr -= diff_l_corr * np.std(l_r_corr)
a_o_corr -= diff_a_corr
b_o_corr -= diff_b_corr

# Zuordnen
l_corr = l_o_corr
a_corr = a_o_corr
b_corr = b_o_corr

# Merge the channels together and convert back to the RGB color
transfer_corr = cv2.merge([l_corr, a_corr, b_corr])*255
transfer_corr = cv2.cvtColor(transfer_corr.astype("uint8"), cv2.COLOR_LAB2BGR)
withblack_corr = cv2.bitwise_and(transfer_corr, transfer_corr, mask=graucorr_corr)

# Ende Farbkorrektur
grau = cv2.cvtColor(withblack_corr, cv2.COLOR_BGR2GRAY)
ret, grau = cv2.threshold(grau, 100, 200, cv2.THRESH_BINARY)
konturen, hierarchy = cv2.findContours(grau, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
konturen = sorted(konturen, key=cv2.contourArea, reverse=True)[1:5]

print(len(konturen))
for knt in konturen:
    approxk = cv2.approxPolyDP(knt, 0.03 * cv2.arcLength(knt, True), True)
    (xk, yk, wk, hk) = cv2.boundingRect(approxk)

    if len(approxk) == 4:
        rectk = cv2.minAreaRect(approxk)
        boxak = cv2.boxPoints(rectk)
        alle3marker = np.int0(boxak)
        marker.append(alle3marker)
        if len(marker) >= 4:
            marker = []
        if len(marker) == 3:
            # Marker entnehmen
            filt_marker1 = marker[0]
            filt_marker2 = marker[1]
            filt_marker3 = marker[2]

            p_mark1rahm_x = filt_marker1[0, 0] # M1UL
            p_mark1rahm_y = filt_marker1[0, 1] # M1UL
            p_mark2rahm_x = filt_marker2[0, 0] # M2UL
            p_mark2rahm_y = filt_marker2[0, 1] # M2UL
            p_mark3rahm_x = filt_marker3[0, 0] # M3UL
            p_mark3rahm_y = filt_marker3[0, 1] # M3UL

            #####
            # Position A
            if p_mark1rahm_x <= (b / 2) and p_mark1rahm_y <= (h / 2) and p_mark2rahm_x >= (b /
2) and p_mark2rahm_y <= (
                h / 2) and p_mark3rahm_x <= (b / 2) and p_mark3rahm_y >= (h / 2) or
                p_mark2rahm_x <= (
                b / 2) and p_mark2rahm_y <= (h / 2) and p_mark1rahm_x >= (b / 2) and
                p_mark1rahm_y <= (
                h / 2) and p_mark3rahm_x <= (b / 2) and p_mark3rahm_y >= (h / 2) or
                p_mark2rahm_x <= (

```

```

b / 2) and p_mark2rahm_y <= (h / 2) and p_mark3rahm_x >= (b / 2) and
p_mark3rahm_y <= (
h / 2) and p_mark1rahm_x <= (b / 2) and p_mark1rahm_y >= (h / 2) or
p_mark1rahm_x <= (
b / 2) and p_mark1rahm_y <= (h / 2) and p_mark3rahm_x >= (b / 2) and
p_mark3rahm_y <= (
h / 2) and p_mark2rahm_x <= (b / 2) and p_mark2rahm_y >= (h / 2) or
p_mark3rahm_x <= (
b / 2) and p_mark3rahm_y <= (h / 2) and p_mark1rahm_x >= (b / 2) and
p_mark1rahm_y <= (
h / 2) and p_mark2rahm_x <= (b / 2) and p_mark2rahm_y >= (h / 2) or
p_mark3rahm_x <= (
b / 2) and p_mark3rahm_y <= (h / 2) and p_mark2rahm_x >= (b / 2) and
p_mark2rahm_y <= (
h / 2) and p_mark1rahm_x <= (b / 2) and p_mark1rahm_y >= (h / 2):
# OL:Bl,OR:Gr,UL:Ro /OR/ OL:Gr,OR:Bl,UL:Ro /OR/ OL:Gr,OR:Ro,UL:Bl /OR/
OL:Bl,OR:Ro,UL:Gr
print("Position A")
Mom = cv2.getRotationMatrix2D(mitte, 270, faktor)
img_rot = cv2.warpAffine(withblack_corr.copy(), Mom, (h, b))
code = img_rot.copy()
rotationsachse = 1

# Position B
if p_mark1rahm_x <= (b / 2) and p_mark1rahm_y <= (h / 2) and p_mark3rahm_x >= (b /
2) and p_mark3rahm_y <= (
h / 2) and p_mark2rahm_x >= (b / 2) and p_mark2rahm_y >= (h / 2) or
p_mark3rahm_x <= (
b / 2) and p_mark3rahm_y <= (h / 2) and p_mark2rahm_x >= (b / 2) and
p_mark2rahm_y <= (
h / 2) and p_mark1rahm_x >= (b / 2) and p_mark1rahm_y >= (h / 2) or
p_mark3rahm_x <= (
b / 2) and p_mark3rahm_y <= (h / 2) and p_mark1rahm_x >= (b / 2) and
p_mark1rahm_y <= (
h / 2) and p_mark2rahm_x >= (b / 2) and p_mark2rahm_y >= (h / 2) or
p_mark2rahm_x <= (
b / 2) and p_mark2rahm_y <= (h / 2) and p_mark1rahm_x >= (b / 2) and
p_mark1rahm_y <= (
h / 2) and p_mark3rahm_x >= (b / 2) and p_mark3rahm_y >= (h / 2) or
p_mark1rahm_x <= (
b / 2) and p_mark1rahm_y <= (h / 2) and p_mark2rahm_x >= (b / 2) and
p_mark2rahm_y <= (
h / 2) and p_mark3rahm_x >= (b / 2) and p_mark3rahm_y >= (h / 2) or
p_mark2rahm_x <= (
b / 2) and p_mark2rahm_y <= (h / 2) and p_mark3rahm_x >= (b / 2) and
p_mark3rahm_y <= (
h / 2) and p_mark1rahm_x >= (b / 2) and p_mark1rahm_y >= (h / 2):
# OL:Bl,OR:Gr,UR:Ro /or/ OL:Bl,OR:Ro,UR:Gr /or/ OL:Ro,OR:Gr,UR:Bl /or/
OL:Ro,OR:Bl,UR:Gr /or/ OL:Gr,OR:Bl,UR:Rot
print("Position B")
code = withblack_corr.copy()
rotationsachse = 1

# Position C
if p_mark1rahm_x <= (b / 2) and p_mark1rahm_y <= (h / 2) and p_mark2rahm_x <= (b /
2) and p_mark2rahm_y >= (
h / 2) and p_mark3rahm_x >= (b / 2) and p_mark3rahm_y >= (h / 2) or
p_mark2rahm_x <= (
b / 2) and p_mark2rahm_y <= (h / 2) and p_mark1rahm_x <= (b / 2) and
p_mark1rahm_y >= (
h / 2) and p_mark3rahm_x >= (b / 2) and p_mark3rahm_y >= (h / 2) or
p_mark2rahm_x <= (
b / 2) and p_mark2rahm_y <= (h / 2) and p_mark3rahm_x <= (b / 2) and
p_mark3rahm_y >= (
h / 2) and p_mark1rahm_x >= (b / 2) and p_mark1rahm_y >= (h / 2) or
p_mark1rahm_x <= (
b / 2) and p_mark1rahm_y <= (h / 2) and p_mark3rahm_x <= (b / 2) and
p_mark3rahm_y >= (
h / 2) and p_mark2rahm_x >= (b / 2) and p_mark2rahm_y >= (h / 2) or
p_mark3rahm_x <= (
b / 2) and p_mark3rahm_y <= (h / 2) and p_mark1rahm_x <= (b / 2) and

```

```

p_mark1rahm_y >= (
h / 2) and p_mark2rahm_x >= (b / 2) and p_mark2rahm_y >= (h / 2) or
p_mark3rahm_x <= (
b / 2) and p_mark3rahm_y <= (h / 2) and p_mark2rahm_x <= (b / 2) and
p mark2rahm y >= (
h / 2) and p_mark1rahm_x >= (b / 2) and p_mark1rahm_y >= (h / 2):
# OL:Bl, UL:Gr, UR:Ro /or/ OL:Gr, UL:Bl, UR:Ro /or/ OL:Bl, UL:Ro, UR:Gr /or/
OL:Ro, UL:Bl, UR:Gr /or/ OL:Ro, UL:Gr, UR:Bl
print("Position C")
Mom = cv2.getRotationMatrix2D(mitte, 180, faktor)
img_rot = cv2.warpAffine(withblack_corr.copy(), Mom, (h, b))
code = img_rot.copy()
rotationsachse = 1

# Position D
if p_mark1rahm_x >= (b / 2) and p_mark1rahm_y <= (h / 2) and p_mark2rahm_x <= (b /
2) and p mark2rahm y >= (
h / 2) and p_mark3rahm_x > (b / 2) and p_mark3rahm_y >= (h / 2) or
p_mark2rahm_x >= (
b / 2) and p_mark2rahm_y <= (h / 2) and p_mark1rahm_x <= (b / 2) and
p_mark1rahm_y >= (
h / 2) and p_mark3rahm_x >= (b / 2) and p_mark3rahm_y >= (h / 2) or
p mark1rahm x >= (
b / 2) and p_mark1rahm_y <= (h / 2) and p_mark3rahm_x <= (b / 2) and
p_mark3rahm_y >= (
h / 2) and p_mark2rahm_x >= (b / 2) and p_mark2rahm_y >= (h / 2) or
p_mark3rahm_x >= (
b / 2) and p_mark3rahm_y <= (h / 2) and p_mark1rahm_x <= (b / 2) and
p mark1rahm y >= (
h / 2) and p_mark2rahm_x >= (b / 2) and p_mark2rahm_y >= (h / 2) or
p_mark2rahm_x >= (
b / 2) and p_mark2rahm_y <= (h / 2) and p_mark3rahm_x <= (b / 2) and
p_mark3rahm_y >= (
h / 2) and p_mark1rahm_x >= (b / 2) and p_mark1rahm_y >= (h / 2) or
p mark3rahm x >= (
b / 2) and p_mark3rahm_y <= (h / 2) and p_mark2rahm_x <= (b / 2) and
p_mark2rahm_y >= (
h / 2) and p_mark1rahm_x >= (b / 2) and p_mark1rahm_y >= (h / 2):
# UL:Gr, OR:Bl, UR:Ro /or / UL:Bl, OR:Gr, UR:Ro /or / OR:Bl,UL:Rot,UR:Gr /or /
OR:Rot,UL:Bl,OR:Gr /or / OR:Gr,UL:Rot,UR:Bl /or / OR:Rot,UL:Gr,UR:Bl
print("Position D")
Mom = cv2.getRotationMatrix2D(mitte, 90, faktor)
img_rot = cv2.warpAffine(withblack_corr.copy(), Mom, (h, b))
code = img_rot.copy()
rotationsachse = 1

# Sollange der Algorithmus berechnet erfolgt folgendes
elif rotationsachse == 0:
code = withblack_corr.copy()
print("Fehler: Bitte versuchen Sie es erneut")
#####
# Hier beginnt nach der Rotation die Auswertung
#####
if rotationsachse == 1:
# Marker im ergebnis Frame erkennen
grey = cv2.cvtColor(code, cv2.COLOR_BGR2GRAY)

ret2, grey = cv2.threshold(grey, 100, 200, cv2.THRESH_BINARY)
cv2.imwrite(os.path.join(scandirpath, "grau.jpg"), grey)
kont, hier = cv2.findContours(grey, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
kont = sorted(kont, key=cv2.contourArea, reverse=True)[2:5]

for ko in kont:
appri = cv2.approxPolyDP(ko, 0.03 * cv2.arcLength(ko, True), True)
(xko, yko, wko, hko) = cv2.boundingRect(appri)

# Quadrat Filter ar <= 1.05
ary = float(wko) / float(hko)
if ary >= 0.9 and ary <= 1.1 and wko <= 155:
if len(appri) == 4:
roti = cv2.minAreaRect(appri)

```

```

boks = cv2.boxPoints(roti)
dreimark = np.int0(boks)
rotmarker.append(dreimark)
if len(rotmarker) >= 4:
    marker = []
if len(rotmarker) == 3:
    # Marker sortieren und filtern und als neue variable speichern
    mark1 = rotmarker[0]
    m1_1 = mark1[0]
    mark2 = rotmarker[1]
    m2_1 = mark2[0]
    mark3 = rotmarker[2]
    m3_1 = mark3[0]

    # Sortier Filter zu Position B
    # Marker 1
    if m1_1[0] < (b / 2) and m1_1[1] < (h / 2):
        OL.append(rotmarker[0])
    if m2_1[0] < (b / 2) and m2_1[1] < (h / 2):
        OL.append(rotmarker[1])
    if m3_1[0] < (b / 2) and m3_1[1] < (h / 2):
        OL.append(rotmarker[2])
    # Marker 2
    if m1_1[0] > (b / 2) and m1_1[1] < (h / 2):
        OR.append(rotmarker[0])
    if m2_1[0] > (b / 2) and m2_1[1] < (h / 2):
        OR.append(rotmarker[1])
    if m3_1[0] > (b / 2) and m3_1[1] < (h / 2):
        OR.append(rotmarker[2])
    # Marker 3
    if m1_1[0] > (b / 2) and m1_1[1] > (h / 2):
        UR.append(rotmarker[0])
    if m2_1[0] > (b / 2) and m2_1[1] > (h / 2):
        UR.append(rotmarker[1])
    if m3_1[0] > (b / 2) and m3_1[1] > (h / 2):
        UR.append(rotmarker[2])

cv2.drawContours(withblack_corr, OL, 0, (0, 255, 0), 2) # gruen
cv2.drawContours(withblack_corr, OR, 0, (0, 255, 0), 2) # gruen
cv2.drawContours(withblack_corr, UR, 0, (0, 255, 0), 2) # gruen

# Punkte im Marker sortieren
marker1 = OL[0]
m1a1 = marker1[0]
m1a2 = marker1[1]
m1a3 = marker1[2]
m1a4 = marker1[3]
marker2 = OR[0]
m2a1 = marker2[0]
m2a2 = marker2[1]
m2a3 = marker2[2]
m2a4 = marker2[3]
marker3 = UR[0]
m3a1 = marker3[0]
m3a2 = marker3[1]
m3a3 = marker3[2]
m3a4 = marker3[3]

marka1 = [[m1a1[0], m1a1[1]], [m1a2[0], m1a2[1]], [m1a3[0], m1a3[1]],
[m1a4[0], m1a4[1]]]
marka2 = [[m2a1[0], m2a1[1]], [m2a2[0], m2a2[1]], [m2a3[0], m2a3[1]],
[m2a4[0], m2a4[1]]]
marka3 = [[m3a1[0], m3a1[1]], [m3a2[0], m3a2[1]], [m3a3[0], m3a3[1]],
[m3a4[0], m3a4[1]]]

marker1_sort = sorted(marka1, key=lambda b: [b[1], b[0]])
marker2_sort = sorted(marka2, key=lambda b: [b[1], b[0]])
marker3_sort = sorted(marka3, key=lambda b: [b[1], b[0]])

# Zugriff auf die Positionen von M1
m1_1 = marker1_sort[0:1]

```

```

ma1_1 = ma1_1[0]
ma1_2 = marker1_sort[1:2]
ma1_2 = ma1_2[0]
ma1_3 = marker1_sort[2:3]
ma1_3 = ma1_3[0]
ma1_4 = marker1_sort[3:4]
ma1_4 = ma1_4[0]

# Zugriff auf die Positionen von M2
ma2_1 = marker2_sort[0:1]
ma2_1 = ma2_1[0]
ma2_2 = marker2_sort[1:2]
ma2_2 = ma2_2[0]
ma2_3 = marker2_sort[2:3]
ma2_3 = ma2_3[0]
ma2_4 = marker2_sort[3:4]
ma2_4 = ma2_4[0]

# Zugriff auf die Positionen von M3
ma3_1 = marker3_sort[0:1]
ma3_1 = ma3_1[0]
ma3_2 = marker3_sort[1:2]
ma3_2 = ma3_2[0]
ma3_3 = marker3_sort[2:3]
ma3_3 = ma3_3[0]
ma3_4 = marker3_sort[3:4]
ma3_4 = ma3_4[0]

# Filter Start #####
# Breite und Hoehe Festlegen der einzelnen Marker
m1_bh = abs(ma1_4[0] + ma1_3[0])
m2_b = abs(ma2_4[0] + ma2_3[0])
m2_h = abs(ma2_2[1] + ma2_4[1])
m3_bh = abs(ma3_4[0] + ma3_3[0])

# Marker 1 OL
if ma1_1[0] < (m1_bh / 2) and ma1_1[1] < (m1_bh / 2):
    M1OL.append(ma1_1)
if ma1_2[0] < (m1_bh / 2) and ma1_2[1] < (m1_bh / 2):
    M1OL.append(ma1_2)
if ma1_3[0] < (m1_bh / 2) and ma1_3[1] < (m1_bh / 2):
    M1OL.append(ma1_3)
if ma1_4[0] < (m1_bh / 2) and ma1_4[1] < (m1_bh / 2):
    M1OL.append(ma1_4)

# Marker 1 OR
if ma1_1[0] > (m1_bh / 2) and ma1_1[1] < (m1_bh / 2):
    M1OR.append(ma1_1)
if ma1_2[0] > (m1_bh / 2) and ma1_2[1] < (m1_bh / 2):
    M1OR.append(ma1_2)
if ma1_3[0] > (m1_bh / 2) and ma1_3[1] < (m1_bh / 2):
    M1OR.append(ma1_3)
if ma1_4[0] > (m1_bh / 2) and ma1_4[1] < (m1_bh / 2):
    M1OR.append(ma1_4)

# Marker 1 UR
if ma1_1[0] > (m1_bh / 2) and ma1_1[1] > (m1_bh / 2):
    M1UR.append(ma1_1)
if ma1_2[0] > (m1_bh / 2) and ma1_2[1] > (m1_bh / 2):
    M1UR.append(ma1_2)
if ma1_3[0] > (m1_bh / 2) and ma1_3[1] > (m1_bh / 2):
    M1UR.append(ma1_3)
if ma1_4[0] > (m1_bh / 2) and ma1_4[1] > (m1_bh / 2):
    M1UR.append(ma1_4)

# Marker 1 UL
if ma1_1[0] < (m1_bh / 2) and ma1_1[1] > (m1_bh / 2):
    M1UL.append(ma1_1)
if ma1_2[0] < (m1_bh / 2) and ma1_2[1] > (m1_bh / 2):
    M1UL.append(ma1_2)
if ma1_3[0] < (m1_bh / 2) and ma1_3[1] > (m1_bh / 2):

```

```

M1UL.append(ma1_3)
if ma1_4[0] < (m1_bh / 2) and ma1_4[1] > (m1_bh / 2):
    M1UL.append(ma1_4)

M1OL = M1OL[0]
M1OR = M1OR[0]
M1UL = M1UL[0]
M1UR = M1UR[0]

# Marker 1
mark1rahm_x = M1UL[0] # M1UL
mark1rahm_y = M1UL[1] # M1UL
mark1rahm_x1 = M1OL[0] # M1OL
mark1rahm_y1 = M1OL[1] # M1OL
mark1rahm_x2 = M1OR[0] # M1OR
mark1rahm_y2 = M1OR[1] # M1OR
mark1rahm_x3 = M1UR[0] # M1UR
mark1rahm_y3 = M1UR[1] # M1UR

# Marker 2 OL
if ma2_1[0] < (m2_b / 2) and ma2_1[1] < (m2_h / 2):
    M2OL.append(ma2_1)
if ma2_2[0] < (m2_b / 2) and ma2_2[1] < (m2_h / 2):
    M2OL.append(ma2_2)
if ma2_3[0] < (m2_b / 2) and ma2_3[1] < (m2_h / 2):
    M2OL.append(ma2_3)
if ma2_4[0] < (m2_b / 2) and ma2_4[1] < (m2_h / 2):
    M2OL.append(ma2_4)

# Marker 2 OR
if ma2_1[0] > (m2_b / 2) and ma2_1[1] < (m2_h / 2):
    M2OR.append(ma2_1)
if ma2_2[0] > (m2_b / 2) and ma2_2[1] < (m2_h / 2):
    M2OR.append(ma2_2)
if ma2_3[0] > (m2_b / 2) and ma2_3[1] < (m2_h / 2):
    M2OR.append(ma2_3)
if ma2_4[0] > (m2_b / 2) and ma2_4[1] < (m2_h / 2):
    M2OR.append(ma2_4)

# Marker 2 UR
if ma2_1[0] > (m2_b / 2) and ma2_1[1] > (m2_h / 2):
    M2UR.append(ma2_1)
if ma2_2[0] > (m2_b / 2) and ma2_2[1] > (m2_h / 2):
    M2UR.append(ma2_2)
if ma2_3[0] > (m2_b / 2) and ma2_3[1] > (m2_h / 2):
    M2UR.append(ma2_3)
if ma2_4[0] > (m2_b / 2) and ma2_4[1] > (m2_h / 2):
    M2UR.append(ma2_4)

# Marker 2 UL
if ma2_1[0] < (m2_b / 2) and ma2_1[1] > (m2_h / 2):
    M2UL.append(ma2_1)
if ma2_2[0] < (m2_b / 2) and ma2_2[1] > (m2_h / 2):
    M2UL.append(ma2_2)
if ma2_3[0] < (m2_b / 2) and ma2_3[1] > (m2_h / 2):
    M2UL.append(ma2_3)
if ma2_4[0] < (m2_b / 2) and ma2_4[1] > (m2_h / 2):
    M2UL.append(ma2_4)

M2OL = M2OL[0]
M2OR = M2OR[0]
M2UL = M2UL[0]
M2UR = M2UR[0]

# Marker 2
rot2 = cv2.minAreaRect(OR[0])
rahm_marker2 = cv2.boxPoints(rot2)
mark2rahm = np.int0(rahm_marker2)
mark2rahm_x = M2UL[0] # M2UL
mark2rahm_y = M2UL[1] # M2UL
mark2rahm_x1 = M2OL[0] # M2O

```

```

mark2rahm_y1 = M2OL[1] # M2OL
mark2rahm_x2 = M2OR[0] # M2OR
mark2rahm_y2 = M2OR[1] # M2OR
mark2rahm_x3 = M2UR[0] # M2UR
mark2rahm_y3 = M2UR[1] # M2UR

# Marker 3 OL
if ma3_1[0] < (m3_bh / 2) and ma3_1[1] < (m3_bh / 2):
    M3OL.append(ma3_1)
if ma3_2[0] < (m3_bh / 2) and ma3_2[1] < (m3_bh / 2):
    M3OL.append(ma3_2)
if ma3_3[0] < (m3_bh / 2) and ma3_3[1] < (m3_bh / 2):
    M3OL.append(ma3_3)
if ma3_4[0] < (m3_bh / 2) and ma3_4[1] < (m3_bh / 2):
    M3OL.append(ma3_4)

# Marker 3 OR
if ma3_1[0] > (m3_bh / 2) and ma3_1[1] < (m3_bh / 2):
    M3OR.append(ma3_1)
if ma3_2[0] > (m3_bh / 2) and ma3_2[1] < (m3_bh / 2):
    M3OR.append(ma3_2)
if ma3_3[0] > (m3_bh / 2) and ma3_3[1] < (m3_bh / 2):
    M3OR.append(ma3_3)
if ma3_4[0] > (m3_bh / 2) and ma3_4[1] < (m3_bh / 2):
    M3OR.append(ma3_4)

# Marker 3 UR
if ma3_1[0] > (m3_bh / 2) and ma3_1[1] > (m3_bh / 2):
    M3UR.append(ma3_1)
if ma3_2[0] > (m3_bh / 2) and ma3_2[1] > (m3_bh / 2):
    M3UR.append(ma3_2)
if ma3_3[0] > (m3_bh / 2) and ma3_3[1] > (m3_bh / 2):
    M3UR.append(ma3_3)
if ma3_4[0] > (m3_bh / 2) and ma3_4[1] > (m3_bh / 2):
    M3UR.append(ma3_4)

# Marker 3 UL
if ma3_1[0] < (m3_bh / 2) and ma3_1[1] > (m3_bh / 2):
    M3UL.append(ma3_1)
if ma3_2[0] < (m3_bh / 2) and ma3_2[1] > (m3_bh / 2):
    M3UL.append(ma3_2)
if ma3_3[0] < (m3_bh / 2) and ma3_3[1] > (m3_bh / 2):
    M3UL.append(ma3_3)
if ma3_4[0] < (m3_bh / 2) and ma3_4[1] > (m3_bh / 2):
    M3UL.append(ma3_4)

M3OL = M3OL[0]
M3OR = M3OR[0]
M3UL = M3UL[0]
M3UR = M3UR[0]

# Marker 3
rot3 = cv2.minAreaRect(UR[0])
rahm_marker3 = cv2.boxPoints(rot3)
mark3rahm = np.int0(rahm_marker3)
mark3rahm_x = M3UL[0] # M3UL
mark3rahm_y = M3UL[1] # M3UL

mark3rahm_x1 = M3OL[0] # M3OL
mark3rahm_y1 = M3OL[1] # M3OL

mark3rahm_x2 = M3OR[0] # M3OR
mark3rahm_y2 = M3OR[1] # M3OR

mark3rahm_x3 = M3UR[0] # M3UR
mark3rahm_y3 = M3UR[1] # M3UR

# Marker Hoehen und Breiten
breite_mark1 = abs((mark1rahm_x - mark1rahm_y) - (mark1rahm_x1 -
mark1rahm_y1))
hoehe_mark1 = abs((mark1rahm_x2 - mark1rahm_y2) - (mark1rahm_x3 -

```

```

mark1rahm_y3))
breite_mark2 = abs((mark2rahm_x - mark2rahm_y) - (mark2rahm_x1 -
mark2rahm_y1))
hoehe_mark2 = abs((mark2rahm_x2 - mark2rahm_y2) - (mark2rahm_x3 -
mark2rahm_y3))
breite_mark3 = abs((mark3rahm_x - mark3rahm_y) - (mark3rahm_x1 -
mark3rahm_y1))
hoehe_mark3 = abs((mark3rahm_x2 - mark3rahm_y2) - (mark3rahm_x3 -
mark3rahm_y3))

# Marker RGB-Werte ausgeben
mark1 = code[mark1rahm_y1 + 7:mark1rahm_y1 + hoehe_mark1 - 7,
mark1rahm_x1 + 7:mark1rahm_x1 + breite_mark1 - 7]

# Daten sind in BGR
m1_B = []
m1_G = []
m1_R = []
marker1_colors = []
for i in mark1:
    for j in i:
        m1_B.append(int(j[0]))
        m1_G.append(int(j[1]))
        m1_R.append(int(j[2]))
        marker1_colors.append('#%02x%02x%02x' % (int(j[2]), int(j[1]),
int(j[0])))

m1_B = np.mean(m1_B)
print("B_Marker1", int(m1_B)) # Datenbank als B_Marker1_cyan
m1_G = np.mean(m1_G)
print("G_Marker1", int(m1_G)) # Datenbank als G_Marker1_cyan
m1_R = np.mean(m1_R)
print("R_Marker1", int(m1_R)) # Datenbank als R_Marker1_cyan

mark2 = code[mark2rahm_y1 + 7:mark2rahm_y1 + hoehe mark2 - 7,
mark2rahm_x1 + 7:mark2rahm_x1 + breite_mark2 - 7]

# Daten sind in BGR
m2_B = []
m2_G = []
m2_R = []
marker2_colors = []
for i in mark2:
    for j in i:
        m2_B.append(int(j[0]))
        m2_G.append(int(j[1]))
        m2_R.append(int(j[2]))
        marker2_colors.append('#%02x%02x%02x' % (int(j[2]), int(j[1]),
int(j[0])))
m2_B = np.mean(m2_B)
print("B_Marker2", int(m2_B)) # Datenbank als B_Marker2_gelb
m2_G = np.mean(m2_G)
print("G_Marker2", int(m2_G)) # Datenbank als G_Marker2_gelb
m2_R = np.mean(m2_R)
print("R_Marker2", int(m2_R)) # Datenbank als R_Marker2_gelb
mark3 = code[mark3rahm_y1 + 7:mark3rahm_y1 + hoehe_mark3 - 7,
mark3rahm_x1 + 7:mark3rahm_x1 + breite_mark3 - 7]

# Daten sind in BGR
m3_B = []
m3_G = []
m3_R = []
marker3_colors = []
for i in mark3:
    for j in i:
        m3_B.append(int(j[0]))
        m3_G.append(int(j[1]))
        m3_R.append(int(j[2]))
        marker3_colors.append('#%02x%02x%02x' % (int(j[2]), int(j[1]),
int(j[0])))
m3_B = np.mean(m3_B)

```



```

print("B Marker3", int(m3_B)) # Datenbank als B_Marker3_magenta
m3_G = np.mean(m3_G)
print("G Marker3", int(m3_G)) # Datenbank als G_Marker3_magenta
m3_R = np.mean(m3_R)
print("R Marker3", int(m3_R)) # Datenbank als R_Marker3_magenta

# Zwei smarte Bereiche Identifizieren
x1_sm_a = mark2rahm_x3 - mark2rahm_y3
y1_sm_a = mark3rahm_x2 - mark3rahm_y2
x2_sm_a = mark2rahm_x - mark2rahm_y
y2_sm_a = mark3rahm_x1 - mark3rahm_y1
breite_sm_a = abs(x1_sm_a - y1_sm_a)
hoehe_sm_a = abs(x2_sm_a - y2_sm_a)
sm_a = code[mark2rahm_y:mark2rahm_y + hoehe_sm_a + 3, mark2rahm_x -
5:mark2rahm_x + breite_sm_a]

##Smart Bereich Analyse: Smartbereich A
smartcodeA = sm_a.copy()
bild = np.rot90(smartcodeA)
bild = np.flip(bild, axis=1)
cv2.imwrite(os.path.join(scandirpath, "smartcodeA_analyse.jpg"), bild)
smartcodeA = cv2.imread(os.path.join(scandirpath, 'smartcodeA_analyse.jpg'))
smartcodeAa = smartcodeA.copy()
sm_a_gray = cv2.cvtColor(smartcodeA, cv2.COLOR_BGR2GRAY)
sm_a_gray = cv2.adaptiveThreshold(sm_a_gray, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY,
3, schwellenwert)
smartdots, hierarchs = cv2.findContours(sm_a_gray, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
smartdots = sorted(smartdots, key=cv2.contourArea, reverse=True)[1:13]
boundingBoxes_smartdots = [cv2.boundingRect(c) for c in smartdots]
(sort_smartdots, boundingBoxes_smartdots) = zip(
*sorted(zip(smartdots, boundingBoxes_smartdots), key=lambda b: [b[1],
b[0]]))
# Smartebereiche werden zuesortiert
sdt_count = 0
for sdt in sort_smartdots:
approxdot = cv2.approxPolyDP(sdt, 0.02 * cv2.arcLength(sdt, True), True)
(sm_a_x, sm_a_y, sm_a_w, sm_a_h) = cv2.boundingRect(approxdot)
cv2.rectangle(smartcodeA, (sm_a_x + 3, sm_a_y + 3), (sm_a_x + sm_a_w -
3, sm_a_y + sm_a_h - 3),
(0, 255, 0), 1)
smartA = smartcodeAa[sm_a_y + 2:sm_a_y + 2 + sm_a_h - 4, sm_a_x +
2:sm_a_x + 2 + sm_a_w - 4]
farbwerte_sdt_minvier = smartcodeA[sm_a_y + sm_a_h // 2:sm_a_y +
sm_a_h // 2 + 3,
sm_a_x + sm_a_w // 2:sm_a_x + sm_a_w // 2 + 3]
sdt_count = sdt_count + 1
# Zeichne die Konturnummer
M = cv2.moments(sdt)
cv2.putText(smartcodeA, "{}".format(str(sdt_count)), (sm_a_x, sm_a_y),
cv2.FONT_HERSHEY_SIMPLEX,
0.25, (0, 0, 0), 1)

for i in farbwerte_sdt_minvier:
for j in i:
if sdt_count <= 4:
sy_B.append(int(j[0]))
sy_G.append(int(j[1]))
sy_R.append(int(j[2]))
if sdt_count > 4 and sdt_count <= 8:
sm_B.append(int(j[0]))
sm_G.append(int(j[1]))
sm_R.append(int(j[2]))
if sdt_count > 8:
sc_B.append(int(j[0]))
sc_G.append(int(j[1]))
sc_R.append(int(j[2]))

x1_sm_b = mark1rahm_x2 - mark1rahm_y2
y1_sm_b = mark2rahm_x1 - mark2rahm_y1

```

```

x2_sm_b = marklrahm_x2 - marklrahm_y2
y2_sm_b = marklrahm_x3 - marklrahm_y3
breite_sm_b = abs(x1_sm_b - y1_sm_b)
hoehe_sm_b = abs(x2_sm_b - y2_sm_b)
sm_b = code[mark2rahm_y1 - 5:mark2rahm_y1 + hoehe_sm_b + 6,
            marklrahm_x2:marklrahm_x2 + breite_sm_b + 5]

##Smart Bereich Analyse: Smartbereich B
smartcodeB = sm_b.copy()
sm_b_height, sm_b_width, sm_b_depth = smartcodeB.shape
sm_b_gray = cv2.cvtColor(smartcodeB, cv2.COLOR_BGR2GRAY)
sm_b_gray = cv2.adaptiveThreshold(sm_b_gray, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY,
                                3, schwellenwert)
smartdotsB, hierarchsB = cv2.findContours(sm_b_gray, cv2.RETR_TREE,
                                          cv2.CHAIN_APPROX_SIMPLE)

smartdotsB = sorted(smartdotsB, key=cv2.contourArea, reverse=True)[1:13]

boundingBoxes_smartdotsB = [cv2.boundingRect(c) for c in smartdotsB]
(sort_smartdotsB, boundingBoxes_smartdotsB) = zip(
    *sorted(zip(smartdotsB, boundingBoxes_smartdotsB), key=lambda b:
             [b[1], b[0]]))
countsdtB = 0
for sdtB in sort_smartdotsB:
    approxdotB = cv2.approxPolyDP(sdtB, 0.02 * cv2.arcLength(sdtB, True),
    True)
    (sm_b_x, sm_b_y, sm_b_w, sm_b_h) = cv2.boundingRect(approxdotB)
    cv2.rectangle(smartcodeB, (sm_b_x, sm_b_y), (sm_b_x + sm_b_w, sm_b_y +
    sm_b_h), (0, 255, 0), 1)
    smartB = sm_b[sm_b_y + 2:sm_b_y + 2 + sm_b_h - 4, sm_b_x + 2:sm_b_x +
    2 + sm_b_w - 4]
    farbwerte_smartB_minvier = smartcodeB[sm_b_y + sm_b_h // 2:sm_b_y +
    sm_b_h // 2 + 3,
                                          sm_b_x + sm_b_w // 2:sm_b_x + sm_b_w // 2 +
    3]

    # Zeichnet die Konturnummer
    M = cv2.moments(sdtB)
    countsdtB = countsdtB + 1
    cv2.putText(smartcodeB, "{}".format(str(countsdtB)), (sm_b_x, sm_b_y),
    cv2.FONT_HERSHEY_SIMPLEX,
    0.25, (0, 0, 0), 1)

    for i in farbwerte_smartB_minvier:
        for j in i:
            if countsdtB <= 4:
                sy_B.append(int(j[0]))
                sy_G.append(int(j[1]))
                sy_R.append(int(j[2]))
            if countsdtB > 4 and countsdtB <= 8:
                sm_B.append(int(j[0]))
                sm_G.append(int(j[1]))
                sm_R.append(int(j[2]))
            if countsdtB > 8:
                sc_B.append(int(j[0]))
                sc_G.append(int(j[1]))
                sc_R.append(int(j[2]))

x1_sm_c = marklrahm_x3 - marklrahm_y3
y1_sm_c = marklrahm_x2 - mark3rahm_y1
x2_sm_c = marklrahm_x - marklrahm_y
y2_sm_c = marklrahm_x - mark3rahm_y1
breite_sm_c = abs(x1_sm_c - y1_sm_c)
hoehe_sm_c = abs(x2_sm_c - y2_sm_c)
sm_c = code[marklrahm_y:marklrahm_y + hoehe_sm_c, marklrahm_x -
5:marklrahm_x + (marklrahm_x3)]

##Smart Bereich Analyse: Smartbereich C
smartcodeC = sm_c.copy()

```

```

bild = np.rot90(smartcodeC)
bild = np.flip(bild, axis=1)
cv2.imwrite(os.path.join(scandirpath, "smartcodeC_analyse.jpg"), bild)
smartcodeC = cv2.imread(os.path.join(scandirpath, 'smartcodeC_analyse.jpg'))
smartcodeCc = smartcodeC.copy()
sm_c_gray = cv2.cvtColor(smartcodeC, cv2.COLOR_BGR2GRAY)
sm_c_gray = cv2.adaptiveThreshold(sm_c_gray, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY,
3, schwellenwert)
smardots, hierarchs = cv2.findContours(sm_c_gray, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
smardots = sorted(smardots, key=cv2.contourArea, reverse=True)[1:13]
boundingBoxes_smardots = [cv2.boundingRect(c) for c in smardots]
(sort_smardots, boundingBoxes_smardots) = zip(
*sorted(zip(smardots, boundingBoxes_smardots), key=lambda b: [b[1],
b[0]]))

sdt_count = 0
for sdt in sort_smardots:
approxdot = cv2.approxPolyDP(sdt, 0.02 * cv2.arcLength(sdt, True), True)
(sm_c_x, sm_c_y, sm_c_w, sm_c_h) = cv2.boundingRect(approxdot)
cv2.rectangle(smartcodeC, (sm_c_x + 3, sm_c_y + 3), (sm_c_x + sm_c_w -
3, sm_c_y + sm_c_h - 3),
(0, 255, 0), 1)
smartC = smartcodeCc[sm_c_y + 2:sm_c_y + 2 + sm_c_h - 4, sm_c_x +
2:sm_c_x + 2 + sm_c_w - 2]
farbwerte_sdt_minvier = smartcodeC[sm_c_y + sm_c_h // 2:sm_c_y +
sm_c_h // 2 + 3,
sm_c_x + sm_c_w // 2:sm_c_x + sm_c_w // 2 + 3]

# Zeichne die Konturnummer
M = cv2.moments(sdt)
sdt_count = sdt_count + 1
cv2.putText(smartcodeC, "{}".format(str(sdt_count)), (sm_c_x, sm_c_y),
cv2.FONT_HERSHEY_SIMPLEX,
0.25, (0, 0, 0), 1)

for i in farbwerte_sdt_minvier:
for j in i:
if sdt_count <= 4:
sy_B.append(int(j[0]))
sy_G.append(int(j[1]))
sy_R.append(int(j[2]))
if sdt_count > 4 and sdt_count <= 8:
sm_B.append(int(j[0]))
sm_G.append(int(j[1]))
sm_R.append(int(j[2]))
if sdt_count > 8:
sc_B.append(int(j[0]))
sc_G.append(int(j[1]))
sc_R.append(int(j[2]))

##### Datenbereich
x1_dat = mark1rahm_x3 - mark1rahm_y3
y1_dat = mark2rahm_x - mark2rahm_y
x2_dat = mark3rahm_x - mark3rahm_y
y2_dat = mark1rahm_x3 - mark3rahm_y
breite_dat = abs(x1_dat - y1_dat)
hoehe_dat = abs(x2_dat - y2_dat)

dat = code[mark1rahm_y - 6:mark1rahm_y + hoehe_dat + 16,
mark1rahm_x3 - 6:mark1rahm_x3 + breite_dat + 6]

# Datenausgabe
auswertung = decode(dat)
if len(auswertung) == 0:
print("READING ERROR")
if len(auswertung) > 0:
auswertung = auswertung[0][0].decode('utf8')
print("Auswertung: ", auswertung) # Datenbank als Auswertung

```

```

##### Smartbereich D
x1_dat = mark1rahm_x - mark3rahm_y1
y1_dat = mark3rahm_x1 - mark3rahm_y1
x2_dat = mark3rahm_x - mark3rahm_y
y2_dat = mark1rahm_x - mark3rahm_y
breite_dat = abs(x1_dat - y1_dat)
hoehe_dat = abs(x2_dat - y2_dat)

# roi = image[y:y+h, x:x+w]
kaldat = code[mark3rahm_y1 - 5:mark3rahm_y1 + mark3rahm_y + 5,
             mark1rahm_x - 5:mark1rahm_x + breite_dat + 5]

##Smart Bereich Analyse: Smartbereich D
smartcodeD = kaldat.copy()
smartcodeD_gray = cv2.cvtColor(smartcodeD, cv2.COLOR_BGR2GRAY)
smartcodeD_gray = cv2.adaptiveThreshold(smartcodeD_gray, 255,
                                       cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                       cv2.THRESH_BINARY, 3, schwellenwert)
kalcont, hierkal = cv2.findContours(smartcodeD_gray, cv2.RETR_TREE,
                                    cv2.CHAIN_APPROX_SIMPLE)
kalcont = sorted(kalcont, key=cv2.contourArea, reverse=True)[1:17]

boundingBoxes_kal = [cv2.boundingRect(c) for c in kalcont]
(sort_smartdotk, boundingBoxes_kal) = zip(
    *sorted(zip(kalcont, boundingBoxes_kal), key=lambda b: [b[1], b[0]]))
countsdk = 0
for sdtk in sort_smartdotk:
    approxdotk = cv2.approxPolyDP(sdtk, 0.02 * cv2.arcLength(sdtk, True),
                                  True)
    (sm_k_x, sm_k_y, sm_k_w, sm_k_h) = cv2.boundingRect(approxdotk)
    cv2.rectangle(smartcodeD, (sm_k_x + 2, sm_k_y + 2),
                  (sm_k_x + 2 + sm_k_w - 6, sm_k_y + sm_k_h - 4), (0, 255,
0), 1)
    smartk = kaldat[sm_k_y + 2:sm_k_y + 2 + sm_k_h - 4, sm_k_x + 2:sm_k_x
+ 2 + sm_k_w - 2]
    farbwerte_smartk_minvier = smartcodeD[sm_k_y + sm_k_h // 2:sm_k_y +
sm_k_h // 2 + 3,
                                           sm_k_x + sm_k_w // 2:sm_k_x + sm_k_w // 2 +
3]

# Zeichnet die Konturnummer
M = cv2.moments(sdtk)
countsdk = countsdk + 1
cv2.putText(smartcodeD, "{}".format(str(countsdk)), (sm_k_x, sm_k_y),
           cv2.FONT_HERSHEY_SIMPLEX,
           0.25, (0, 0, 0), 1)

for i in farbwerte_smartk_minvier:
    for j in i:
        if countsdk == 1:
            s1.append(j)
        if countsdk == 2:
            s2.append(j)
        if countsdk == 3:
            s3.append(j)
        if countsdk == 4:
            s4.append(j)
        if countsdk > 4 and countsdk < 8:
            sy_B.append(int(j[0]))
            sy_G.append(int(j[1]))
            sy_R.append(int(j[2]))
        if countsdk >= 9 and countsdk <= 12:
            sm_B.append(int(j[0]))
            sm_G.append(int(j[1]))
            sm_R.append(int(j[2]))
        if countsdk >= 13:
            sc_B.append(int(j[0]))
            sc_G.append(int(j[1]))
            sc_R.append(int(j[2]))

```

```

# Graustufen s1-s4 anpassen
for i in s1:
    s1_B.append(int(i[0]))
    s1_G.append(int(i[1]))
    s1_R.append(int(i[2]))
for i in s2:
    s2_B.append(int(i[0]))
    s2_G.append(int(i[1]))
    s2_R.append(int(i[2]))
for i in s3:
    s3_B.append(int(i[0]))
    s3_G.append(int(i[1]))
    s3_R.append(int(i[2]))
for i in s4:
    s4_B.append(int(i[0]))
    s4_G.append(int(i[1]))
    s4_R.append(int(i[2]))

print("#####")
print("sc_B", np.mean(sc_B)) # Datenbank als SmartC_B
print("sc_G", np.mean(sc_G)) # Datenbank als SmartC_G
print("sc_R", np.mean(sc_R)) # Datenbank als SmartC_R
print("#####")
print("sm_B", np.mean(sm_B)) # Datenbank als SmartM_B
print("sm_G", np.mean(sm_G)) # Datenbank als SmartM_G
print("sm_R", np.mean(sm_R)) # Datenbank als SmartM_R
print("#####")
print("sy_B", np.mean(sy_B)) # Datenbank als SmartY_B
print("sy_G", np.mean(sy_G)) # Datenbank als SmartY_G
print("sy_R", np.mean(sy_R)) # Datenbank als SmartY_R
print("#####")
print("s1_B", np.mean(s1_B)) # Datenbank als S1_B
print("s1_G", np.mean(s1_G)) # Datenbank als S1_G
print("s1_R", np.mean(s1_R)) # Datenbank als S1_R
print("#####")
print("s2_B", np.mean(s2_B)) # Datenbank als S2_B
print("s2_G", np.mean(s2_G)) # Datenbank als S2_G
print("s2_R", np.mean(s2_R)) # Datenbank als S2_R
print("#####")
print("s3_B", np.mean(s3_B)) # Datenbank als S3_B
print("s3_G", np.mean(s3_G)) # Datenbank als S3_G
print("s3_R", np.mean(s3_R)) # Datenbank als S3_R
print("#####")
print("s4_B", np.mean(s4_B)) # Datenbank als S4_B
print("s4_G", np.mean(s4_G)) # Datenbank als S4_G
print("s4_R", np.mean(s4_R)) # Datenbank als S4_R

# Marker und Code als Bild speichern
cv2.imwrite(os.path.join(scandirpath, "PositionB.jpg"), code)
cv2.imwrite(os.path.join(scandirpath, "mark1_PositionB_OL.jpg"), mark1)
cv2.imwrite(os.path.join(scandirpath, "mark2_PositionB_OR.jpg"), mark2)
if len(mark3) > 0: # Notwendig weil der Probleme macht und den Marker
nicht erkennt
    cv2.imwrite(os.path.join(scandirpath, "mark3_PositionB_UR.jpg"), mark3)
cv2.imwrite(os.path.join(scandirpath, "PositionB_SM_A.jpg"), smartcodeA)
cv2.imwrite(os.path.join(scandirpath, "PositionB_SM_B.jpg"), smartcodeB)
cv2.imwrite(os.path.join(scandirpath, "PositionB_SM_C.jpg"), smartcodeC)
cv2.imwrite(os.path.join(scandirpath, "smartcodeD.jpg"), smartcodeD)
cv2.imwrite(os.path.join(scandirpath, "Datenbereich.jpg"), dat)

f = open(os.path.join(scandirpath, "colorvalues.txt"), "w")
colorvalues_json = {"marker1": marker1_colors, "marker2": marker2_colors,
"marker3": marker3_colors}
f.write(json.dumps(colorvalues_json))
f.close()

# codemodel
dbmodel = CodeModel()
path = Path(args.p).parts
dbmodel.name = path[len(path) - 2] #filename

```

```

dbmodel.datamatrix = auswertung
dbmodel.alpha = 0.0 # alpha
dbmodel.beta = 0.0 # beta
dbmodel.paperwhite_color = Color(255, 255, 255) # Color(int(hint_R),
int(hint_G), int(hint_B))
dbmodel.marker1_position = Point(mark1rahm_x1, mark1rahm_y1)
dbmodel.marker2_position = Point(mark2rahm_x1, mark2rahm_y1)
dbmodel.marker3_position = Point(mark3rahm_x1, mark3rahm_y1)
dbmodel.marker1_color = Color(int(m1_R), int(m1_G), int(m1_B))
dbmodel.marker2_color = Color(int(m2_R), int(m2_G), int(m2_B))
dbmodel.marker3_color = Color(int(m3_R), int(m3_G), int(m3_B))
dbmodel.smartC_color = Color(int(np.mean(sc_R)), int(np.mean(sc_G)),
int(np.mean(sc_B)))
dbmodel.smartM_color = Color(int(np.mean(sm_R)), int(np.mean(sm_G)),
int(np.mean(sm_B)))
dbmodel.smartY_color = Color(int(np.mean(sy_R)), int(np.mean(sy_G)),
int(np.mean(sy_B)))
dbmodel.s1_color = Color(int(np.mean(s1_R)), int(np.mean(s1_G)),
int(np.mean(s1_B)))
dbmodel.s2_color = Color(int(np.mean(s2_R)), int(np.mean(s2_G)),
int(np.mean(s2_B)))
dbmodel.s3_color = Color(int(np.mean(s3_R)), int(np.mean(s3_G)),
int(np.mean(s3_B)))
dbmodel.s4_color = Color(int(np.mean(s4_R)), int(np.mean(s4_G)),
int(np.mean(s4_B)))
# only prints out sql string, not really inserting
insert_into_database(dbmodel)
print("#scs-42")
calculate latest differece()

cv2.waitKey()
cv2.destroyAllWindows()

```

## E) Programmstruktur Auswerteeinheit

```
import MySQLdb
from skimage import color
from create_plot import plot

def calculate_latest_differerece():
    # Stelle Verbindung zur Datenbank her
    db = MySQLdb.connect(host="██████████", user="██████████", passwd="██████████",
db="db_django")
    # Setze Autocommit auf True (Damit Änderungen an der Datenbank direkt übernommen werden
    und nicht nochmal bestätigt werden müssen)
    db.autocommit(True)
    cur = db.cursor(MySQLdb.cursors.DictCursor)

    # sql Abfrage, um die neueste Differenz zu bekommen, also den neuesten Vergleich
    sql_get_latest_difference = "select * from codereader_scandifference order by id desc
    limit 1;"
    # ausführen der Abfrage
    cur.execute(sql_get_latest_difference)
    latest_diff = cur.fetchone()
    latest_id = -1
    # falls es einen letzten Vergleich gibt, speichere die ID des 1. Scans des Vergleichs
    if latest_diff is not None:
        latest_id = latest_diff['id_scan1']

    # sql Abfrage, um die beiden aktuellsten Scans zu bekommen
    sql_get_scans = "select * from codereader_scan order by datetime desc limit 2;"
    cur.execute(sql_get_scans)
    # speichere die beiden Scans in Variablen
    scan1 = cur.fetchone()
    scan2 = cur.fetchone()
    # Wenn die ID des letzten Vergleichs nicht mit der ID des einen Scans übereinstimmt, dann
    erstelle einen neuen Vergleich
    if int(latest_id) != int(scan1['id']):
        # speichere alle relevanten Werte in Variablen
        scan1_marker1_lab = color.rgb2lab(hex2rgb(scan1['marker1_color']))
        scan1_marker2_lab = color.rgb2lab(hex2rgb(scan1['marker2_color']))
        scan1_marker3_lab = color.rgb2lab(hex2rgb(scan1['marker3_color']))
        scan1_smartC_lab = color.rgb2lab(hex2rgb(scan1['smartC_color']))
        scan1_smartM_lab = color.rgb2lab(hex2rgb(scan1['smartM_color']))
        scan1_smartY_lab = color.rgb2lab(hex2rgb(scan1['smartY_color']))
        scan1_s1_lab = color.rgb2lab(hex2rgb(scan1['s1_color']))
        scan1_s2_lab = color.rgb2lab(hex2rgb(scan1['s2_color']))
        scan1_s3_lab = color.rgb2lab(hex2rgb(scan1['s3_color']))
        scan1_s4_lab = color.rgb2lab(hex2rgb(scan1['s4_color']))

        scan2_marker1_lab = color.rgb2lab(hex2rgb(scan2['marker1_color']))
        scan2_marker2_lab = color.rgb2lab(hex2rgb(scan2['marker2_color']))
        scan2_marker3_lab = color.rgb2lab(hex2rgb(scan2['marker3_color']))
        scan2_smartC_lab = color.rgb2lab(hex2rgb(scan2['smartC_color']))
        scan2_smartM_lab = color.rgb2lab(hex2rgb(scan2['smartM_color']))
        scan2_smartY_lab = color.rgb2lab(hex2rgb(scan2['smartY_color']))
        scan2_s1_lab = color.rgb2lab(hex2rgb(scan2['s1_color']))
        scan2_s2_lab = color.rgb2lab(hex2rgb(scan2['s2_color']))
        scan2_s3_lab = color.rgb2lab(hex2rgb(scan2['s3_color']))
        scan2_s4_lab = color.rgb2lab(hex2rgb(scan2['s4_color']))

        # Berechne die benötigten Werte und Speichere sie in Tupeln: (delta l, delta a, delta
        b, delta cie2000, delta cie76)
        delta_marker1 = (scan2_marker1_lab[0] - scan1_marker1_lab[0],
            scan2_marker1_lab[1] - scan1_marker1_lab[1],
            scan2_marker1_lab[2] - scan1_marker1_lab[2],
            color.deltaE_ciede2000(scan2_marker1_lab, scan1_marker1_lab),
            color.deltaE_cie76(scan2_marker1_lab, scan1_marker1_lab))

        delta_marker2 = (scan2_marker2_lab[0] - scan1_marker2_lab[0],
            scan2_marker2_lab[1] - scan1_marker2_lab[1],
            scan2_marker2_lab[2] - scan1_marker2_lab[2],
            color.deltaE_ciede2000(scan2_marker2_lab, scan1_marker2_lab),
            color.deltaE_cie76(scan2_marker2_lab, scan1_marker2_lab))
```

```

delta_marker3 = (scan2_marker3_lab[0] - scan1_marker3_lab[0],
                scan2_marker3_lab[1] - scan1_marker3_lab[1],
                scan2_marker3_lab[2] - scan1_marker3_lab[2],
                color.deltaE_ciede2000(scan2_marker3_lab, scan1_marker3_lab),
                color.deltaE_cie76(scan2_marker3_lab, scan1_marker3_lab))

delta_smartC = (scan2_smartC_lab[0] - scan1_smartC_lab[0],
                scan2_smartC_lab[1] - scan1_smartC_lab[1],
                scan2_smartC_lab[2] - scan1_smartC_lab[2],
                color.deltaE_ciede2000(scan2_smartC_lab, scan1_smartC_lab),
                color.deltaE_cie76(scan2_smartC_lab, scan1_smartC_lab))

delta_smartM = (scan2_smartM_lab[0] - scan1_smartM_lab[0],
                scan2_smartM_lab[1] - scan1_smartM_lab[1],
                scan2_smartM_lab[2] - scan1_smartM_lab[2],
                color.deltaE_ciede2000(scan2_smartM_lab, scan1_smartM_lab),
                color.deltaE_cie76(scan2_smartM_lab, scan1_smartM_lab))

delta_smartY = (scan2_smartY_lab[0] - scan1_smartY_lab[0],
                scan2_smartY_lab[1] - scan1_smartY_lab[1],
                scan2_smartY_lab[2] - scan1_smartY_lab[2],
                color.deltaE_ciede2000(scan2_smartY_lab, scan1_smartY_lab),
                color.deltaE_cie76(scan2_smartY_lab, scan1_smartY_lab))

delta_s1 = (scan2_s1_lab[0] - scan1_s1_lab[0],
            scan2_s1_lab[1] - scan1_s1_lab[1],
            scan2_s1_lab[2] - scan1_s1_lab[2],
            color.deltaE_ciede2000(scan2_s1_lab, scan1_s1_lab),
            color.deltaE_cie76(scan2_s1_lab, scan1_s1_lab))

delta_s2 = (scan2_s2_lab[0] - scan1_s2_lab[0],
            scan2_s2_lab[1] - scan1_s2_lab[1],
            scan2_s2_lab[2] - scan1_s2_lab[2],
            color.deltaE_ciede2000(scan2_s2_lab, scan1_s2_lab),
            color.deltaE_cie76(scan2_s2_lab, scan1_s2_lab))

delta_s3 = (scan2_s3_lab[0] - scan1_s3_lab[0],
            scan2_s3_lab[1] - scan1_s3_lab[1],
            scan2_s3_lab[2] - scan1_s3_lab[2],
            color.deltaE_ciede2000(scan2_s3_lab, scan1_s3_lab),
            color.deltaE_cie76(scan2_s3_lab, scan1_s3_lab))

delta_s4 = (scan2_s4_lab[0] - scan1_s4_lab[0],
            scan2_s4_lab[1] - scan1_s4_lab[1],
            scan2_s4_lab[2] - scan1_s4_lab[2],
            color.deltaE_ciede2000(scan2_s4_lab, scan1_s4_lab),
            color.deltaE_cie76(scan2_s4_lab, scan1_s4_lab))

# sql Anfrage, um alle berechneten Werte in die Datenbank zu speichern, die
# Platzhalter werden ersetzt durch die Werte unten
sql_insert = "insert into codereader_scandifference values" \
            "(NULL, '{0}', '{1}', '{2}', '{3}', '{4}', '{5}', '{6}', '{7}', '{8}',
            '{9}', '{10}', " \
            "'{11}', '{12}', '{13}', '{14}', '{15}', '{16}', '{17}', '{18}', '{19}',
            '{20}', " \
            "'{21}', '{22}', '{23}', '{24}', '{25}', '{26}', '{27}', '{28}', '{29}',
            '{30}', " \
            "'{31}', '{32}', '{33}', '{34}', '{35}', '{36}', '{37}', '{38}', '{39}',
            '{40}', " \
            "'{41}', '{42}', '{43}', '{44}', '{45}', '{46}', '{47}', '{48}', '{49}',
            '{50}', " \
            "'{51}', '{52}', '{53}', '{54}', '{55}', '{56}', '{57}', '{58}', '{59}',
            '{60}', " \
            "'{61}', '{62}', '{63}', '{64}', '{65}', '{66}', '{67}', '{68}', '{69}',
            '{70}', " \
            "'{71}");".format(
            scan1['id'], scan2['id'],
            delta_marker1[3], delta_marker1[4], delta_marker1[0],
            delta_marker1[1], delta_marker1[2],
            delta_marker2[3], delta_marker2[4], delta_marker2[0],
            delta_marker2[1], delta_marker2[2],

```



```

delta_marker3[3], delta_marker3[4], delta_marker3[0],
delta_marker3[1], delta_marker3[2],
delta_smartC[3], delta_smartC[4], delta_smartC[0], delta_smartC[1],
delta_smartC[2],
delta_smartM[3], delta_smartM[4], delta_smartM[0], delta_smartM[1],
delta_smartM[2],
delta_smartY[3], delta_smartY[4], delta_smartY[0], delta_smartY[1],
delta_smartY[2],
delta_s1[3], delta_s1[4], delta_s1[0], delta_s1[1], delta_s1[2],
delta_s2[3], delta_s2[4], delta_s2[0], delta_s2[1], delta_s2[2],
delta_s3[3], delta_s3[4], delta_s3[0], delta_s3[1], delta_s3[2],
delta_s4[3], delta_s4[4], delta_s4[0], delta_s4[1], delta_s4[2],
scan1_marker1_lab, scan2_marker1_lab, scan1_marker2_lab,
scan2_marker2_lab,
scan1_marker3_lab, scan2_marker3_lab, scan1_s1_lab, scan2_s1_lab,
scan1_s2_lab, scan2_s2_lab,
scan1_s3_lab, scan2_s3_lab, scan1_s4_lab, scan2_s4_lab,
scan1_smartC_lab, scan2_smartC_lab,
scan1_smartM_lab, scan2_smartM_lab, scan1_smartY_lab, scan2_smartY_lab)
# Printe als Kontrolle alle Werte noch mal in die Konsole
print(delta_marker1[3], delta_marker1[4], delta_marker1[0], delta_marker1[1],
delta_marker1[2],
delta_marker2[3], delta_marker2[4], delta_marker2[0], delta_marker2[1],
delta_marker2[2],
delta_marker3[3], delta_marker3[4], delta_marker3[0], delta_marker3[1],
delta_marker3[2],
delta_smartC[3], delta_smartC[4], delta_smartC[0], delta_smartC[1],
delta_smartC[2],
delta_smartM[3], delta_smartM[4], delta_smartM[0], delta_smartM[1],
delta_smartM[2],
delta_smartY[3], delta_smartY[4], delta_smartY[0], delta_smartY[1],
delta_smartY[2],
delta_s1[3], delta_s1[4], delta_s1[0], delta_s1[1], delta_s1[2],
delta_s2[3], delta_s2[4], delta_s2[0], delta_s2[1], delta_s2[2],
delta_s3[3], delta_s3[4], delta_s3[0], delta_s3[1], delta_s3[2],
delta_s4[3], delta_s4[4], delta_s4[0], delta_s4[1], delta_s4[2],
scan1_marker1_lab, scan2_marker1_lab, scan1_marker2_lab, scan2_marker2_lab,
scan1_marker3_lab, scan2_marker3_lab, scan1_s1_lab, scan2_s1_lab, scan1_s2_lab,
scan2_s2_lab,
scan1_s3_lab, scan2_s3_lab, scan1_s4_lab, scan2_s4_lab, scan1_smartC_lab,
scan2_smartC_lab,
scan1_smartM_lab, scan2_smartM_lab, scan1_smartY_lab, scan2_smartY_lab)
# führe Datenbank insert durch
cur.execute(sql_insert)
cur.execute(sql_get_latest_difference)
latest_diff = cur.fetchone()
plot(scan1["imagename"], scan2["imagename"], latest_diff["id"])
# schließe die DB-Verbindung
db.close()

# Funktion um Hexwerte #aabbcc -> in einen rgb Wert zu konvertieren (von 0.0-1.0, nicht 0-255)
def hex2rgb(hexval):
    # lösche das '#'
    h = hexval.lstrip('#')
    # erstelle ein 3er Tupel, indem erst mit "int(h[i:i+2], 16)" 2er Paare von Hex-Zahlen in
    Dezimalwerte umgewandelt werden,
    # dann die jeweiligen Werte durch 255 geteilt werden (Wertebereich von 0-1), auf 5
    Nachkommastellen gerundet werden
    # und das ganze insgesamt 3 mal, für alle 6 Zahlen des Hexwerts
    # Beispiel: #12ABD3 -> 12 = 18, 18/255=0.07059; AB = 171, 171/255 = 0.67059; D3 = 211,
    211/255=0.82745
    return tuple(round(int(h[i:i+2], 16) / 255, 5) for i in (0, 2, 4))

# Falls diese Datei hier direkt ausgeführt wird, führe automatisch die Funktion
calculate_latest_differerece aus.
# Falls diese Datei von einer anderen eingebunden wird, mache nichts
if __name__ == "__main__":
    calculate_latest_differerece()

```

## F) Programmstruktur Surround-Light-Assistent

```

#From Mustafa Bilgin #Python version: 2.7.13 #Opencv version: 3.4.1
#img: http://mustafa-bilgin.de/40_2.jpg
#d65: http://mustafa-bilgin.de/d65.npy
import cv2
import numpy as np
import sys

#Import Image
img = cv2.imread('40_2.jpg')
input = img.copy() #Code copy
d65 = np.load('d65.npy')

# Masking parameter only background-white in upper and lower range
lower_white = np.array([130, 130, 130], dtype=np.uint8)#100
upper_white = np.array([255, 255, 255], dtype=np.uint8)#255

# Masking parameter only code black
img_lower_white = np.array([0, 0, 0], dtype=np.uint8)#0
img_upper_white = np.array([130, 130, 130], dtype=np.uint8)#130#180
mask_img = cv2.inRange(img, img_lower_white, img_upper_white)
masked_img = cv2.bitwise_and(img, img, mask=mask_img)
masked_img = np.where(masked_img != 0, 255, masked_img)
withblack = cv2.inRange(masked_img, img_lower_white, img_upper_white)

#Target: Extract bg color
mask = cv2.inRange(input, lower_white, upper_white)
masked_scan = cv2.bitwise_and(input, input, mask=mask)
mittw = np.mean(masked_scan)#5.8 schwarz / 70.75 maske
if mittw < 80:
    sys.exit('Kamera nicht geeignet: Farbfehler')
bg_code = np.delete(masked_scan, np.argwhere(masked_scan == [0,0,0]), axis=0)
bg_code = bg_code[:3]

#convert the images from the RGB/BGR to Lab color space
scan = cv2.cvtColor(bg_code, cv2.COLOR_BGR2LAB).astype("float32")/255#bg color
reference = cv2.cvtColor(d65, cv2.COLOR_BGR2LAB).astype("float32")/255#d65
img = cv2.cvtColor(img, cv2.COLOR_BGR2LAB).astype("float32")/255#original_image

#Split Lab color space
(l_s, a_s, b_s) = cv2.split(scan)#masked_scan
(l_r, a_r, b_r) = cv2.split(reference)#reference
(l_o, a_o, b_o) = cv2.split(img)#original_scan

#standard deviation
std_ls = np.std(l_s)
std_lr = np.std(l_r)

#Color difference between both images
diff_l = np.mean(l_s) - np.mean(l_r)
diff_a = np.mean(a_s) - np.mean(a_r)
diff_b = np.mean(b_s) - np.mean(b_r)

#subtract difference
l_o -= diff_l * np.std(l_r)
a_o -= diff_a
b_o -= diff_b

# Zuordnen
l = l_o
a = a_o
b = b_o

# Merge the channels together and convert back to the RGB color
transfer = cv2.merge([l, a, b])*255
transfer = cv2.cvtColor(transfer.astype("uint8"), cv2.COLOR_LAB2BGR)
withblack = cv2.bitwise_and(transfer, transfer, mask=withblack)

#Show image
cv2.imshow("Scanned-Code", input)
cv2.imshow("correct_black", withblack)
cv2.waitKey()

```

## Literaturverzeichnis

- Aday, M. S. & Yener, U. (2015). Assessing consumers' adoption of active and intelligent packaging. *British Food Journal*, 117 (1), 157-177.
- Ahvenainen, R. Active and intelligent packaging. In *Novel Food Packaging Techniques* (S. 5-21).
- Aida, K., Araya, K., Kawasaki, M. & Mori, S. (2020). *Temperature Detecting Material, Temperature Detecting Ink Using Same, Temperature Indicator, and Product Control System. U.S. patent* (Hitachi Industrial Equipment Systems Co., Ltd., Hrsg.) (US20200041359A1).
- Alpeyev, P. & Furukawa, Y. (Bloomberg, Hrsg.). (2019). *Sony Can't Make Image Sensors Fast Enough to Keep Up With Demand*. Zugriff am 27.11.2020. Verfügbar unter <https://www.bloomberg.com/news/articles/2019-12-23/sony-can-t-make-image-sensors-fast-enough-to-keep-up-with-demand>
- Anton Paar (Hrsg.). (2021). *The influence of particles on suspension rheology*. Zugriff am 11.06.2021. Verfügbar unter <https://wiki.anton-paar.com/at-de/auswirkungen-von-partikeln-auf-die-suspensionsrheologie/>
- Atlanta Chemical Engin. (Atlanta Chemical Engineering LLC, Hrsg.). (2020). *THERMOCHROMIC PAINTS. The irreversible thermochromic paints change color just ONCE at the indicated activation temperature*. Thermochrome. Zugriff am 29.11.2020. Verfügbar unter <https://www.atlantachemical.com/tpt/tpt.html>
- Balaban, R. (2018). *Intro to Python Image Processing in Computational Photography*. Zugriff am 06.05.2021. Verfügbar unter <https://www.toptal.com/opencv/python-image-processing-in-computational-photography>
- BAM (Bundesanstalt für Materialforschung und -prüfung, Hrsg.). (2018). *Analytical Sciences. Sensorik*. Zugriff am 26.11.2020. Verfügbar unter [www.bam.de/Navigation/DE/Themen/Analytical-Sciences/Sensorik/sensorik.html](http://www.bam.de/Navigation/DE/Themen/Analytical-Sciences/Sensorik/sensorik.html)
- Bamfield, P. & Hutchings, M. (2018). *Chromic Phenomena*. Cambridge: Royal Society of Chemistry.
- Bamfield, P. & Hutchings, M. G. (2001). *Chromic phenomena. Technological applications of colour chemistry* (2. ed.). Cambridge: RSC Publ. Verfügbar unter <https://pubs.rsc.org/en/content/ebook/978-1-84755-868-8>
- Bayer, B. E. (1976, 20. Juli). *Color imaging array. US Patent* (Eastman Kodak Co, Hrsg.) (US3971065A).
- Bear, M. F., Connors, B. W. & Paradiso, M. A. (2018). *Neurowissenschaften. Ein grundlegendes Lehrbuch für Biologie, Medizin und Psychologie* (4. Aufl. 2018). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Bechtold, T., Mahmud-Ali, A. & Mussak, R. (2007). Anthocyanin dyes extracted from grape pomace for the purpose of textile dyeing. *Journal of the science of food and agriculture*, 87 (14), 2589-2595.
- Bielek & Barker. *Laser Activated Thermochromic Compositions. US Patent* (US 2006/024.1225A1).

- Bilgin, M. Halochromic Composite Materials. In *Composite Materials* .
- Bilgin, M. (Hrsg.). (2021). *Halochromic Composite Materials*. In *Encyclopedia of Materials: Composites Reference Module in Materials Science and Materials Engineering*. Amsterdam, Niederlande: Elsevier.
- Bilgin, M. & Backhaus, J. (2017a). *Intelligent codes by controlled responsiveness to external stimuli*. Chemnitz, Deutschland: VWB – Verlag für Wissenschaft und Bildung (S. 85-90).
- Bilgin, M. & Backhaus, J. (2017b). *Smart Packages by Means of Intelligent Codes*. Fribourg, Schweiz. *Advances in Printing and Media Technology*.
- Bilgin, M. & Backhaus, J. (2018a). *Development of a unidirectional switchable Photochromic Ink for Smart Packaging*. Warschau, Polen. *Advances in Printing and Media Technology*.
- Bilgin, M. & Backhaus, J. (2018b). *First Steps to Develop Molecular Switches for Piezoelectric Inkjet* (TAGA, Hrsg.). Baltimore, Maryland, U.S.A. *Technical Association for the Graphic Arts*.
- Bilgin, M. & Backhaus, J. (2018c). *Initial Steps to Develop Piezoelectric Inkjet Ink Containing Stimuli-responsive Pigments* (IC, Hrsg.) (Nr. 11). Toronto, Kanada. *International Circular of Graphic Education and Research, IC* (S. 8-13). Zugriff am 27.11.2020. Verfügbar unter [www.internationalcircle.net/international\\_circle/circular/issues/18\\_02/ICJ\\_11\\_2018\\_02](http://www.internationalcircle.net/international_circle/circular/issues/18_02/ICJ_11_2018_02)
- Bilgin, M. & Backhaus, J. (2019a). *Development Process of a Smart Code Based On Smart Materials* (TAGA, Hrsg.). Minneapolis, Minnesota, U.S.A. *Technical Association for the Graphic Arts*.
- Bilgin, M. & Backhaus, J. (2019b). *Smart Materials Detection Using Computer Vision*. Stuttgart, Deutschland. *Advances in Printing and Media Technology*.
- Bilgin, M. & Backhaus, J. (2019c). Smart packaging by the application of sensitive dyes. *Journal of Print and Media Technology Research* (2), 101-110.
- Bilgin, M. & Backhaus, J. (2020a). Development of an irreversible Hydrochromic Ink for Smart Packaging. *Journal of Print and Media Technology Research* (3), 137-143. Verfügbar unter [www.jpmttr.org/jpmttr\\_9\(2020\)3\\_web\\_2004.pdf](http://www.jpmttr.org/jpmttr_9(2020)3_web_2004.pdf)
- Bilgin, M. & Backhaus, J. (2020b). *Development process of a hydrochromic ink* (TAGA, Hrsg.). Oklahoma City, Oklahoma, U.S.A. *Technical Association for the Graphic Arts*.
- Bilgin, M. & Backhaus, J. (2020c). *Hydrochromic Ink as a Security Feature for Piezoelectric Inkjet Printing* (IC, Hrsg.) (Nr. 12). *International Circular of Graphic Education and Research, IC* (S. 17-28). Zugriff am 27.11.2020. Verfügbar unter [www.internationalcircle.net/international\\_circle/circular/issues/20\\_01/ICJ\\_12\\_2020\\_03](http://www.internationalcircle.net/international_circle/circular/issues/20_01/ICJ_12_2020_03)
- Bilgin, M. & Backhaus, J. (2020d). *Printable smart materials used as sensors for continuous monitoring in a smart code*. Oklahoma, Ok, U.S.A: Curran Associates, Inc.
- Binniger, B. (2016). *Vorlesung 3.5 Thermodynamik der Grenzflächen*. Verfügbar unter [www.itv.rwth-aachen.de/fileadmin/LehreSeminar/Thermodynamik\\_II/WS16\\_Vorlesungen/Thermodynamik\\_II\\_Kap3\\_Teil5von5.pdf](http://www.itv.rwth-aachen.de/fileadmin/LehreSeminar/Thermodynamik_II/WS16_Vorlesungen/Thermodynamik_II_Kap3_Teil5von5.pdf)
- Blank, F. (1947). The Anthocyanin Pigments of Plants. *Botanical Review*. Springer; New York Botanical Garden Press (Vol. 13, No. 5), 241-317.

- Blume, R. (2003). *Prof. Blumes Tipp des Monats: Chemie des Tintenkillers*, Chemie und Didaktik der Chemie (DC II). Universität Bielefeld. 75. Verfügbar unter [www.chemieunterricht.de/dc2/tip/09\\_03.htm](http://www.chemieunterricht.de/dc2/tip/09_03.htm)
- Böhringer, J., Bühler, P., Schlaich, P. & Sinner, D. (2014). *Kompendium der Mediengestaltung*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Brand, Hülser, Grimm & Zweck. (2009). *Internet der Dinge - Perspektiven für die Logistik. Übersichtsstudie* (Zukünftige Technologien Consulting der VDI Technologiezentrum GmbH, Hrsg.) (Nr. 80), Düsseldorf. Zugriff am 28.11.2020. Verfügbar unter [www.vditz.de/publikation/internet-der-dinge-perspektiven-fuer-die-logistik-1/](http://www.vditz.de/publikation/internet-der-dinge-perspektiven-fuer-die-logistik-1/)
- Branda et al. (2006). *Photochromic and Electrochromic compounds and methods of synthesizing*. US Patent (US 2006/0240197 A1).
- Brettle & Trundle. (1989). *Irreversible Photochromic Markings*. US Patent (4,812,171).
- Bromberger, B. & Mausbach, K. (1987). *Frauen und Frankfurt. Spuren vergessener Geschichte*. Frankfurt/Main: Verl. für Akademische Schriften VAS.
- Bühler, P., Schlaich, P. & Sinner, D. (2018). *Digitale Farbe*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Burger, W. & Burge, M. J. (2005). *Digitale Bildverarbeitung. Eine Einführung mit Java und ImageJ* (eXamen.press). Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg. Verfügbar unter [www.gbv.ebib.com/patron/FullRecord.aspx?p=1156284](http://www.gbv.ebib.com/patron/FullRecord.aspx?p=1156284)
- Burklund, A., Saturley-Hall, H. K., Franchina, F. A., Hill, J. E. & Zhang, J. X. J. (2019). Printable QR code paper microfluidic colorimetric assay for screening volatile biomarkers. *Biosensors & bioelectronics*, 128, 97-103.
- Cameron. (2004, 03. Februar). *Thermochromic bar code*. US Patent (US 6,685,094 B2).
- Campbell, M. K. & Farrell, S. O. (2008). *Biochemistry* (6. ed., internat. student ed.). Belmont Calif.: Brooks/Cole.
- Carey, F. A. & Sundberg, R. J. (2008). *Structure and mechanisms* (Advanced organic chemistry, / Francis A. Carey and Richard J. Sundberg ; Pt. A, 5. ed., corr. 2. printing). New York, NY: Springer.
- Carl Roth (Hrsg.). (2016). *Sicherheitsdatenblatt: Natriumhydrogencarbonat*. Zugriff am 27.11.2020. Verfügbar unter [www.mustafa-bilgin.de/dissertation/SDB-6885-DE-DE.pdf](http://www.mustafa-bilgin.de/dissertation/SDB-6885-DE-DE.pdf)
- Carlos, E., Goldsmith, C., Goldsmith, M. & Woodaman J. (2000). *Food Sentinel System*. EU Patent (SIRA Technologies, Hrsg.) (EP1018013A1).
- Chen, Y. et al. (2017). Low cost smart phone diagnostics for food using paper-based colorimetric sensor arrays. *Food Control*, 82, 227-232.
- Crane, R. (1997). *A simplified approach to image processing. Classical and modern techniques in C* (Hewlett-Packard professional books). Upper Saddle River, NJ: Prentice Hall.

- CTI Inks. (2020). *Sunlight "Burnout" irreversible inks. photochromic ink, this technology activates under sunlight or UV light*. Zugriff am 29.11.2020. Verfügbar unter <https://www.ctiinks.com/glossary>
- Cummins, G. & Desmulliez, M. P. (2012). Inkjet printing of conductive materials: a review. *Circuit World*, 38 (4), 193-213.
- DensoWave. (2020). *History of the QR Code*. Zugriff am 27.11.2020. Verfügbar unter [www.qrcode.com](http://www.qrcode.com)
- Developer.mozilla (Mozilla, Hrsg.). (2020). *Browser compatibility MediaDevices*. Zugriff am 30.11.2020. Verfügbar unter [www.developer.mozilla.org/en-US/docs/Web/API/MediaDevices](http://www.developer.mozilla.org/en-US/docs/Web/API/MediaDevices)
- Dittmann, J. (2000). *Digitale Wasserzeichen*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- DIN ISO, 12040. *Druck- und Reproduktionstechnik - Drucke und Druckfarben - Bestimmung der Leuchteichtheit mit gefiltertem Xenon-Bogenlicht*. Berlin, Deutschland: Beuth Verlag GmbH.
- Duden (Hrsg.). (2020). *Wörterbuch*. Zugriff am 28.11.2020. Verfügbar unter [www.duden.de/](http://www.duden.de/)
- Durđević, S., Novaković, D., Adamović, S., Boadu, F., Rodríguez Lezaca, A. & Zeljković, Ž. (112018). Developing augmented reality app for smart packaging. In *Proceedings of 9th International Symposium on Graphic Engineering and Design* (S. 435-443). Faculty of Technical Sciences.
- Durini, D. (2020). *High performance silicon imaging. Fundamentals and applications of CMOS and CCD sensors* (Second edition).
- Ecker, M. & Pretsch, T. (2014). Novel design approaches for multifunctional information carriers. *RSC Adv*, 4 (87), 46680-46688.
- Edinger, M., Bar-Shalom, D., Sandler, N., Rantanen, J. & Genina, N. (2018). QR encoded smart oral dosage forms by inkjet printing. *International journal of pharmaceutics*, 536 (1), 138-145.
- Eiblmeier, J. & Metz, R. (Papiertechnische Stiftung (PTS), Hrsg.). (2015). *Entwicklung eines innovativen Manipulationskennzeichens für Hot-Melt-Verklebungen in pharmazeutischen Faltschachteln*. IGF 1311-0004. Zugriff am 26.11.2020. Verfügbar unter [www.ptspaper.de/fileadmin/PTS/PTSPAPER/06\\_Forschung/Dokumente/Forschungsprojekte/BAY1311\\_0004.pdf](http://www.ptspaper.de/fileadmin/PTS/PTSPAPER/06_Forschung/Dokumente/Forschungsprojekte/BAY1311_0004.pdf)
- Elhard & Heggs. (2004). *Solvent-Activated Color-Forming Compositions*. *US Patent* (US 2004/0067590 A1).
- Engel, T. & Reid, P. J. (2006). *Physikalische Chemie* (bc Chemie). München: Pearson Studium.
- Epson (Hrsg.). (2020). *EcoTank ET-2650*. Zugriff am 30.11.2020. Verfügbar unter [www.epson.de/products/printers/inkjet-printers/for-home/ecotank-et-2650](http://www.epson.de/products/printers/inkjet-printers/for-home/ecotank-et-2650)
- Ernst, H. (2003). *Grundkurs Informatik. Grundlagen und Konzepte für die erfolgreiche IT-Praxis - Eine umfassende, praxisorientierte Einführung* (3., überarbeitete und verbesserte Auflage). Wiesbaden: Vieweg+Teubner Verlag.
- Europäische Kommission. (2012). Verordnung der Kommission mit Spezifikationen für die in den Anhängen II und III der Verordnung (EG) Nr. 1333/2008 des Europäischen Parlaments und des Rates aufgeführten Lebensmittelzusatzstoffe. EU Nr. 231/2012. *Amtsblatt der Europäischen Union*.

- Verfügbar unter [www.eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32012R0231&from=LV](http://www.eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32012R0231&from=LV)
- DIN, 5033-1 (01.10.2017). *Farbmessung - Teil 1: Grundbegriffe der Farbmatrik*. Berlin, Deutschland: Beuth Verlag GmbH.
- DIN, 5033-7 (01.10.2014). *Farbmessung – Teil 7: Messbedingungen für Körperfarben*. Berlin, Deutschland: Beuth Verlag GmbH.
- DIN EN ISO/CIE, 11664-1 (01.03.2020). *Farbmatrik - Teil 1: CIE farbmatische Normalbeobachter*. Berlin, Deutschland: Beuth Verlag GmbH.
- DIN EN ISO/CIE, 11664-3 (01.03.2020). *Farbmatrik - Teil 3: CIE-Farbwerte*. Berlin, Deutschland: Beuth Verlag GmbH.
- DIN EN ISO/CIE, 11664-4 (01.03.2020). *Farbmatrik – Teil 4: CIE 1976 L\*a\*b\* Farbraum // DIN EN ISO/CIE 11664-4:2020-03, Farbmatrik\_ Teil\_4: CIE\_1976 L\*a\*b\* Farbraum (ISO/CIE\_11664-4:2019); Deutsche Fassung EN\_ISO/CIE\_11664-4:2019*. Berlin, Deutschland: Beuth Verlag GmbH.
- DIN EN ISO, 11664-6 (01.12.2016). *Farbmatrik – Teil 6: CIEDE2000 Formel für den Farbabstand*. Berlin, Deutschland: Beuth Verlag GmbH.
- Fischer et al. (1993, 05. Januar). *Photochromic Benzothioxanthone oxides, Process for their preparation and The use thereof. US Patent (5,177,218)*. U.S.A.
- Fischer et al. (1994, 10. Mai). *Photochromic Compounds, a Process for their Preparation, and their usw. US Patent (5,310,909)*. U.S.A.
- Fogra (Hrsg.). (2001). *Lichtechtheit, Was ist das?* Zugriff am 27.11.2020. Verfügbar unter [www.wolke.com/sites/default/files/downloads/lichtechtheit\\_was\\_ist\\_das.pdf](http://www.wolke.com/sites/default/files/downloads/lichtechtheit_was_ist_das.pdf)
- Frank, H., Jailbird & Phrood (Wikipedia, Hrsg.). (2005). *Elektromagnetisches Spektrum*. Zugriff am 30.11.2020. Verfügbar unter [www.de.wikipedia.org/wiki/Elektromagnetisches\\_Spektrum#/media/Datei:Electromagnetic\\_spectrum\\_-de\\_c.svg](http://www.de.wikipedia.org/wiki/Elektromagnetisches_Spektrum#/media/Datei:Electromagnetic_spectrum_-de_c.svg)
- Freeman, H. S. & Peters, A. T. (2000). *Colorants for non-textile applications* (1. ed.). Amsterdam: Elsevier.
- Fritzsche. (1867). Ueber die festen Kohlenwasserstoffe des Steinkohlentheers. *Journal fr Praktische Chemie*, 101 (1), 333-343.
- Fukuda, Y. (Ed.). (2007). *Inorganic chromotropism. Basic concepts and applications of colored materials ; with 34 tables*. Berlin: Springer.
- FX Pigments. (2020). *FX Pigments Pvt. Ltd. Thermochromic Ink Irreversible. Piezochromic Irreversible Ink*. Zugriff am 29.11.2020. Verfügbar unter <https://www.fxpigments.in/sitemap.html>
- Fyvie et al. (2011). *Thermochromic Ink and Coating Compositions. US Patent (US 8,002,851 B2)*.
- Gabbai, A. (Smithsonian Magazine, Hrsg.). (2015). *Kevin Ashton Describes "the Internet of Things"*. Zugriff am 26.11.2020. Verfügbar unter [www.smithsonianmag.com/innovation/kevin-ashton-describes-the-internet-of-things-180953749/](http://www.smithsonianmag.com/innovation/kevin-ashton-describes-the-internet-of-things-180953749/)

- Gao, Z., Han, Y. & Wang, F. (2018). Cooperative supramolecular polymers with anthracene–endoperoxide photo-switching for fluorescent anti-counterfeiting. *Nature communications*, 9 (1), 3977.
- Gligoric, N. et al. (2019). SmartTags: IoT Product Passport for Circular Economy Based on Printed Sensors and Unique Item-Level Identifiers. *Sensors* (19(03)). Zugriff am 27.11.2020. Verfügbar unter [www.mdpi.com/1424-8220/19/3/586](http://www.mdpi.com/1424-8220/19/3/586)
- Goldsmith, Goldsmith, C., Woodaman, J. G., Park, D. L. & Ayala, C. E. (1999, 25. März). *Food Sentinel System*. WIPO (PCT) (WO1999014598A1).
- Gottwein. (2020). *Griechisches Online-Wörterbuch*. Zugriff am 05.12.2020. Verfügbar unter [www.gottwein.de/GrWk/Gr00.php](http://www.gottwein.de/GrWk/Gr00.php)
- DIN ISO, 13655 (01.07.2017). *Graphic technology — Spectral measurement and colorimetric computation for graphic arts images*. Berlin, Deutschland: Beuth Verlag GmbH.
- Green, P. (2010). *Color management. Understanding and using ICC profiles* (The Wiley-IS&T Series in Imaging Science and Technology, v.17). Chichester, West Sussex, U.K: Wiley.
- (GS1, Hrsg.). (2018). *GS1 DataMatrix Guideline. Overview and technical introduction to the use of GS1 DataMatrix*, GS1 The Global Language of Business. Zugriff am 27.11.2020. Verfügbar unter [www.gs1.org/docs/barcodes/GS1\\_DataMatrix\\_Guideline.pdf](http://www.gs1.org/docs/barcodes/GS1_DataMatrix_Guideline.pdf)
- H2020 (Europäische Kommission, Hrsg.). (2019, 11. September). *TagItSmart! - Smart Tags driven service platform for enabling ecosystems of connected objects* (H2020-EU.2.1.1.). 199390. Zugriff am 26.11.2020. Verfügbar unter [www.cordis.europa.eu/project/id/688061/results/de](http://www.cordis.europa.eu/project/id/688061/results/de)
- Haarer & Levy. (2011, 14. April). *Method and system for determining the condition of a time-temperature indicator*. US Patent (US 2011/0084.128A1).
- Hädener, A. & Kaufmann, H. (2006). *Grundlagen der organischen Chemie*: Birkhäuser Basel.
- Haenselmann, T. (2016). *Vorlesungsinhalt: Bildverarbeitung SS2016*, Hochschule Mittweida. Zugriff am 27.11.2020. Verfügbar unter [https://www.staff.hs-mittweida.de/~haenselm/dbv\\_2016ss\\_hough.mp4](https://www.staff.hs-mittweida.de/~haenselm/dbv_2016ss_hough.mp4)
- Haenselmann, T. (2017). *Vorlesungsinhalt: Bildverarbeitung WS2017/18*, Hochschule Mittweida. Zugriff am 27.11.2020. Verfügbar unter [www.staff.hs-mittweida.de/~haenselm/topics/dbv/dbv\\_ws1617\\_konvexe\\_huelle.mp4](http://www.staff.hs-mittweida.de/~haenselm/topics/dbv/dbv_ws1617_konvexe_huelle.mp4)
- Hajime & Kang. (2012). *Adjustable colorimetric moisture indicators*. International Publication (WO 2014/078577 A1).
- Hanke. (2004). *Ink used for printing data on deposit strip or label for marking e.g. drink can or plastics or glass bottle, contains irreversibly thermochromic latent pigment undergoing significant color change below specified temperature*. Patentamt DE (DE 000010303914 A1).
- Hashizume, K., Taketani & Masatoshi. (2016). *Code, informationsverarbeitungsverfahren, Informationsverarbeitungsrichtung, Programm, Speichermedium, Tinte, Pigment, und Speichervorrichtung*. Patentamt (DE 112016003968 T5).



- Hauptert, V. (2020). *PhotoTAN-Banking ist nicht sicher. Informatiker der FAU manipulieren Transaktionen auf dem Smartphone*, Friedrich-Alexander-Universität Erlangen-Nürnberg. Zugriff am 28.11.2020.
- Havens, M. R. & Barmore, C. R. (2010, 06. Juli). *Non-invasive method of determining oxygen concentration in a sealed package*. US Patent (Cryovac LLC, Hrsg.) (US7749768B2).
- Heilmann, J., Beletski, N., Hurme, E., Smolander, M., Hakola, L., Linna, H. et al. (Hrsg.). (2008, 26. Juni). *Monitoring system based on the combination of a code and an indicator*. WIPO (PCT) (WO2008074917A1).
- Heitler, W. & London, F. (1927). Wechselwirkung neutraler Atome und homopolare Bindung nach der Quantenmechanik. *Zeitschrift für Physik*, 44 (6-7), 455-472.
- Herbst, W. & Hunger, K. (1995). *Industrielle Organische Pigmente*. Hoboken: Wiley-VCH. Verfügbar unter [www.gbv.ebib.com/patron/FullRecord.aspx?p=481659](http://www.gbv.ebib.com/patron/FullRecord.aspx?p=481659)
- Hering, E., Bressler, K. & Gutekunst, J. (2017). *Elektronik für Ingenieure und Naturwissenschaftler*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Herschel, J. F. W. (1842). XII. On the action of the rays of the solar spectrum on vegetable colours, and on some new photographic processes. *Philosophical Transactions of the Royal Society of London*, 132, 181-214.
- Hessler, B., Schiffhauer, N. & Neitzel, D. (Chip, Hrsg.). (2003). *Digitalfotografie: Sensor-Technologie. CMOS oder CCD: Duell der Sensoren*. Zugriff am 30.11.2020. Verfügbar unter [www.chip.de/artikel/Digitalfotografie-Sensor-Technologie-2\\_140028125.html](http://www.chip.de/artikel/Digitalfotografie-Sensor-Technologie-2_140028125.html)
- Hirokawa et al. (2011). *Perimidine-substituted squarylium dye, dispersion medium, detection medium and image forming material*. US Patent (US 7,985,291 B2).
- Hirshberg, Y. (Hrsg.). (1950). *Photochromie dans la serie de la bianthrone*. (Band 231. Nr. 903). Verfügbar unter [www.gallica.bnf.fr/ark:/12148/bpt6k3183z/f903.item.r=Yehuda%20Hirshberg](http://www.gallica.bnf.fr/ark:/12148/bpt6k3183z/f903.item.r=Yehuda%20Hirshberg)
- Hirshberg, Y. (1956). Reversible Formation and Eradication of Colors by Irradiation at Low Temperatures. A Photochemical Memory Model. *Journal of the American Chemical Society*, 78 (10), 2304-2312.
- Hirshberg, Y. & Fischer, E. (1954). Photochromism and reversible multiple internal transitions in some spiropyrans at low temperatures. Part I. *Journal of the Chemical Society (Resumed)*, 297.
- Hoath, S. D. (Ed.). (2016). *Fundamentals of inkjet printing. The science of inkjet and droplets*. Weinheim: Wiley-VCH.
- Hough, P. V. (1962). General purpose visual input for a computer. *Annals of the New York Academy of Sciences*, 99, 323-334.
- Hutchings & Martin. (2013). *Inkjet technology for digital fabrication*. Chichester: Wiley.
- Hyde & Pan. (2015, 29. Dezember). *Time-dependent identification systems, methods, and uses thereof*. US Patent (US 9,221.273 B1).

- IEEE (Bahn, C., Hrsg.). (1990). *IEEE Standard Glossary of Software Engineering Terminology*, Institute of Electrical and Electronics Engineers, IEEE. IEEE 610.12-1990. Zugriff am 27.11.2020. Verfügbar unter [www.standards.ieee.org/standard/610\\_12-1990.html](http://www.standards.ieee.org/standard/610_12-1990.html)
- ISO/IEC, 18004 (01.02.2015). *Informationstechnik - Automatische Identifikation und Datenerfassungsverfahren - Spezifikation der Barcode - Symbologie "QR Code"*. Berlin, Deutschland: Beuth Verlag GmbH.
- ISO/IEC, 16022 (01.09.2006). *Informationstechnik - Internationale Symbologie-Spezifizierung - Daten-Matrix*. Berlin, Deutschland: Beuth Verlag GmbH.
- Insilico. (2020). *Photochromic Microcapsule. Bichrom P Series is a reversible photochromic microcapsule that its color in lightless state changes to other color once exposed to light*. Zugriff am 29.11.2020. Verfügbar unter <http://www.insilico.co.kr/eng/project/photochromic/>
- Irie, M., Fukaminato, T., Matsuda, K. & Kobatake, S. (2014). Photochromism of diarylethene molecules and crystals: memories, switches, and actuators. *Chemical reviews*, 114 (24), 12174-12277.
- Jennings, C., Aboba, B., Bruaroey, J.-I. & Boström, H. (2020). *Media Capture and Streams. W3C Candidate Recommendation Draft*, W3C. Zugriff am 28.11.2020. Verfügbar unter [www.w3.org/TR/mediacapture-streams/](http://www.w3.org/TR/mediacapture-streams/)
- Kanakkanatt et al. (2012). *Liquid activated color change ink and methods of use. International Publication* (WO 2014/052549 A1).
- Kaplan et al. (2017). *Use of indigo-derivatives as irreversible color changeable dyes for textile materials, Novel compounds and a process for dyeing textile materials. International Publication* (WO 2017/215795 A1).
- Kawasaki, M. & Kohhei, A. (Hitachi, Ltd., Hrsg.). (2020). *Ink as a Sensor—Thermochromic Ink Innovates Quality Control in Logistics*. Zugriff am 30.11.2020. Verfügbar unter [www.hitachi.com/rd/sc/story/ink/index.html](http://www.hitachi.com/rd/sc/story/ink/index.html)
- Khatri, Z., Ali, S., Khatri, I., Mayakrishnan, G., Kim, S. H. & Kim, I.-S. (2015). UV-responsive polyvinyl alcohol nanofibers prepared by electrospinning. *Applied Surface Science*, 342, 64-68.
- Khorana, H. G. (1992). Rhodopsin, photoreceptor of the rod cell. An emerging pattern for structure and function. *The Journal of biological chemistry*, 267 (1), 1-4.
- Kim, Moon, J. H., Lee, S.-Y. & Park, J. (2010). Biologically inspired humidity sensor based on three-dimensional photonic crystals. *Applied Physics Letters*, 97 (10), 103701.
- Kim et al. (2017). *Hydrochromic polydacetylene Composite composition, Hydrochromic thin film using same, and use thereof. US Patent* (US 2017/0190814 A1).
- Kirby, J. (1993). Fading and Colour Change of Prussian Blue: Occurrences and Early Reports In: National Gallery Technical Bulletin. *National Gallery Publications* (Volume 14), 62-71.
- Klemm, F. (Wikimedia, Hrsg.). (2010, 09. Juli). *CFA Pattern fuer quadratische und rechteckige Pixel*. Zugriff am 30.11.2020. Verfügbar unter [www.de.wikipedia.org/wiki/Datei:CFA\\_Pattern\\_fuer\\_quadratische\\_und\\_rechteckige\\_Pixel.png](http://www.de.wikipedia.org/wiki/Datei:CFA_Pattern_fuer_quadratische_und_rechteckige_Pixel.png)

- Klöckl, I. (2015). *Chemie der Farbstoffe. In der Malerei*. Berlin/Boston: De Gruyter.
- Konen & Zielke. (2008). *Bildverarbeitung und Algorithmen*. Zugriff am 12.12.2020. Verfügbar unter <http://www.gm.fh-koeln.de/~konen/WPF-BV-WS08/BV06a.PDF>
- Konica Minolta (Hrsg.). (2020). *1931 x,y chromaticity diagram*. Zugriff am 30.11.2020. Verfügbar unter [www5.konicaminolta.eu/en/measuring-instruments/media-centre/poster-download.html](http://www5.konicaminolta.eu/en/measuring-instruments/media-centre/poster-download.html)
- Kraft, A. (2008). On the discovery and history of Prussian blue. *Bulletin for the History of Chemistry* (33), 61-67. Zugriff am 27.11.2020. Verfügbar unter [www.acshist.scs.illinois.edu/bulletin\\_open\\_access/v33-2/v33-2%20p61-67.pdf](http://www.acshist.scs.illinois.edu/bulletin_open_access/v33-2/v33-2%20p61-67.pdf)
- Kraft, A. (2012). *Wege des Wissens: Berliner Blau, 1706-1726* (GDC, Hrsg.) (Bd 22). Frankfurt / Main: Gesellschaft Deutscher Chemiker (GDC). Verfügbar unter [www.gdch.de/fileadmin/downloads/Netzwerk\\_und\\_Strukturen/Fachgruppen/Geschichte\\_der\\_Chemie/Mitteilungen\\_Band\\_22/2012-22-02.pdf](http://www.gdch.de/fileadmin/downloads/Netzwerk_und_Strukturen/Fachgruppen/Geschichte_der_Chemie/Mitteilungen_Band_22/2012-22-02.pdf)
- Krüss (Hrsg.). (2020). *Blasendruck-Tensiometer*. Zugriff am 30.11.2020. Verfügbar unter [www.kruss-scientific.com/de/service/schulung-theorie/glossar/blasendruck-tensiometer/](http://www.kruss-scientific.com/de/service/schulung-theorie/glossar/blasendruck-tensiometer/)
- Kurihara et al. (2012). *Discolorable ink composition and writing utensils*. US Patent (US 8,182,596 B2).
- Kurose, J. F. & Ross, K. W. (2013). *Computer networking. A top-down approach* (Always learning, 6. ed.). Boston, Mass.: Pearson.
- Kwan. (2014). *Irreversible colorchanging ink compositions*. US Patent (US 8,865,621 B2).
- Labs, L. (2007). *Barcodes in vier Dimensionen. Das Fotografieren von Barcodes ersetzt die mühsame Eingabe von Informationen auf Handys und Smartphones. Forscher der Uni Weimar erhöhen die übertragbare Datenmenge, indem sie die Barcodes in regelmäßigen Abständen austauschen*. Zugriff am 28.11.2020. Verfügbar unter [www.heise.de/newsticker/meldung/Barcodes-in-vier-Dimensionen-174706.html](http://www.heise.de/newsticker/meldung/Barcodes-in-vier-Dimensionen-174706.html)
- Lawrence Industries. (2020). *Chromazone (reversible) and Kromagen (irreversible) thermochromics are available in a variety of temperature and colour combinations to suit customers' individual needs*. Zugriff am 29.11.2020. Verfügbar unter <https://www.l-i.co.uk/principals/tmc>
- LCR Hallcrest. (2020a). *Irreversible Inks – Thermochromic permanent change inks are a high temperature activated, permanent change pigment used in manufacturing processes such as metalworking, annealing, welding and riveting for quality purposes*. Zugriff am 29.11.2020. Verfügbar unter <https://www.lcrhallcrest.com/thermochromic-technology/>
- LCR Hallcrest. (2020b). *The irreversible hydrochromics are available in various colours*. Zugriff am 29.11.2020. Verfügbar unter <https://www.lcrhallcrest.com/hydrochromics/>
- Lechner, M. D. (2017). *Einführung in die Quantenchemie. Aufbau der Atome und Moleküle, Spektroskopie*. Berlin: Springer Spektrum.
- Lee et al. (2016, 25. Oktober). *Time-temperature indicator, method for manufacturing the time-temperature indicator, quality guarantee system using the time-temperature indicator, and quality guarantee method using the quality guarantee system*. US Patent (US 9,476,083 B2).

- Lenk, B. (2002). *Handbuch der automatischen Identifikation* (1. Aufl., 1. Tsd., Stand: 1. Juli 2002). Kirchheim unter Teck: Lenk.
- Lenk, B. (2007). *Handbuch der automatischen Identifikation* (3. Aufl., Stand: 01. Dez. 2007). Kirchheim unter Teck: Lenk.
- Lent & Elue. (1988). *Thermochromic jet ink. EU Patent* (0 289 141 A2).
- Lewis. (2010). *Human and scanner readable radiation exposure indicator with reactive barcode. US Patent* (US 7,798.414 B2).
- Li, B. et al. (2020). A rapid and reversible thermochromic supramolecular polymer hydrogel and its application in protected quick response codes. *Materials Chemistry Frontiers*, 4 (3), 869-874.
- Liu, S., Ma, Y., Liu, S., Chen, K., Huang, W. & Zhao, Q. (2020). Achieving multiple emission states and controllable response behaviour in thermochromic luminescent materials for security applications. *Journal of Materials Chemistry C*, 8 (31), 10798-10804.
- Loreto, S. & Romano, S. P. (2014). *Real-time communication with WebRTC*. Sebastopol, CA: O'Reilly Media.
- Lübbe, E. (2013). *Farbempfindung, Farbbeschreibung und Farbmessung*. Wiesbaden: Springer Fachmedien Wiesbaden.
- Lucht & Euler. (2009). *Low temperature irreversible thermochromic compositions. International Publication* (WO 2009/137709 A).
- MacNair, R. N. (1967). Photochromism\* in triphenylmethanes. *Photochemistry and Photobiology*, 6 (11), 779-797.
- Magdassi, S. (2010). *The chemistry of inkjet inks*. Singapore: World Scientific Pub. Co.
- Magen, E. & Isbitsky, R. (2019, 31. Juli). *Elapsed time temperature indicator and method of activation and use. EU Patent* (EP 3 517 910 A1).
- Malic et al. (2009). *Stabilized photochromic. International Publication* (WO 2009/121149 A1).
- Masterton, W. L., Hurley, C. N. & Neth, E. J. (2012). *Chemistry. Principles and reactions* (7e). Belmont, CA: Brooks/Cole, Cengage Learning.
- Matsui. *Thermolock. Matsui Color: Thermolock Specialty Temperature Sensitive Product*. Zugriff am 29.11.2020. Verfügbar unter <http://matsui-color.com/thermolock-3>
- McHugh, S. (2005). *Color Space Conversion*, Cambridge in Colour. Zugriff am 26.11.2020. Verfügbar unter [www.cambridgeincolour.com/tutorials/color-space-conversion.htm](http://www.cambridgeincolour.com/tutorials/color-space-conversion.htm)
- Meng, Y., Qiu, J., Wu, S., Ju, B., Zhang, S. & Tang, B. (2018). Biomimetic Structural Color Films with a Bilayer Inverse Heterostructure for Anticounterfeiting Applications. *ACS applied materials & interfaces*, 10 (44), 38459-38465.
- Mezger, T. G. (2016). *Das Rheologie Handbuch. Für Anwender von Rotations- und Oszillations-Rheometern* (Farbe und Lack, 5. Auflage, revidierte Ausgabe). Hannover: Vincentz Network.
- Miyata et al. (2003). *Irreversible heat-sensitive composition. US Patent* (US 6,602,594 B2).

- Nanomatrixsecure. (2020). *Thermochromic Ink. Irreversible inks*. Zugriff am 29.11.2020. Verfügbar unter <https://www.nanomatrixsecure.com/thermochromic-ink/>
- Nassau, K. (1983). *The physics and chemistry of color. The fifteen causes of color* ((Wiley series in pure and applied optics)). New York: Wiley.
- Nassau, K. (2001). *The physics and chemistry of color. The fifteen causes of color* (Wiley series in pure and applied optics, 2nd ed.). New York: Wiley.
- NCC New Prismatic Ent. (2020). *Neben den Standard-Typen bietet NCC mittlerweile auch einige Sondertypen an, so. z.B. Irreversible Thermochromic Ink mit einem einmaligen irreversiblen Farbumschlag*. Zugriff am 29.11.2020. Verfügbar unter <https://www.raimund-mueller.de/de/partner/ncc>
- Nemet, Y. & Brand, E. (2013, 19. Dezember). *System and method for quality management utilizing barcode indicators. US Patent* (Varcode Ltd, Hrsg.) (US20130334301A1).
- Newcolorchem. (2020). *We offer irreversible water-based thermochormic ink now*. Zugriff am 29.11.2020. Verfügbar unter [https://newcolorchem.wordpress.com/2012/10/10/we-offer-irreversible-water-based-thermochromic-ink-now/amp/](https://newcolorchem.wordpress.com/2012/10/10/we-offer-irreversible-water-based-thermochromic-ink-now/)
- Nguyen, D. K., Bach, Q.-V., Lee, J.-H. & Kim, I.-T. (2018). Synthesis and Irreversible Thermochromic Sensor Applications of Manganese Violet. *Materials (Basel, Switzerland)*, 11 (9).
- Nicholas et al. (1993, 10. Februar). *Temperature-sensitive devices. EU Patent* (0 250 217 B1).
- NiGK Corporation. (2020). *Irreversible Temperature Indicator. radiation detectors*. Zugriff am 29.11.2020. Verfügbar unter <https://www.nichigi.co.jp/en.html>
- Norrby & Nygård. (2006, 29. März). *A label having a temperature-monitoring function, a package for goods Provided with a label, as well as a method for the application of labels to Packages for goods. EU Patent* (EP 1 901 925 B1).
- Okumura, M. & Takahashi, T. (2007). *Technical program and proceedings / NIP 23, 23rd International Conference on Digital Printing Technologies. September 16 - 21, 2007, Anchorage, Alaska*. Springfield, Va.: IS&T.
- OliKrom. (2020). *de la température (propriété thermochrome), de la lumière (propriété photochrome), de la pression (propriété piézochrome), d'un solvant (propriété solvatochrome), d'un gaz (propriété chimiochrome)*. Zugriff am 29.11.2020.
- Onoda, T. & Miwa, K. (2013, 20. August). *Layered two-dimensional code, creation method thereof, and read method. US Patent* (US8511562B2).
- Onslow, M. W. (1925). *The anthocyanin pigments of plants* (Second edition). Cambridge: At the University Press.
- OpenCV (Hrsg.). (2020a). *boxPoints*. Zugriff am 27.11.2020. Verfügbar unter [docs.opencv.org/master/d3/dc0/group\\_\\_imgproc\\_\\_shape.html#gaf78d467e024b4d7936cf9397185d2f5c](https://docs.opencv.org/master/d3/dc0/group__imgproc__shape.html#gaf78d467e024b4d7936cf9397185d2f5c)
- OpenCV (Hrsg.). (2020b). *Graustufenkonvertierung*. Zugriff am 27.11.2020. Verfügbar unter [https://docs.opencv.org/3.4/de/d25/imgproc\\_color\\_conversions.html](https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html)

- OpenCV (Hrsg.). (2020c). *Hierarchische Konturen*. Zugriff am 27.11.2020. Verfügbar unter [https://docs.opencv.org/master/d9/d8b/tutorial\\_py\\_contours\\_hierarchy.html](https://docs.opencv.org/master/d9/d8b/tutorial_py_contours_hierarchy.html)
- OpenCV (Hrsg.). (2020d). *Hough Line Transform*. Zugriff am 27.11.2020. Verfügbar unter [https://docs.opencv.org/3.4/d9/db0/tutorial\\_hough\\_lines.html](https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html)
- OpenCV (Hrsg.). (2020e). *Perspektivische Transformation*. Zugriff am 27.11.2020. Verfügbar unter [https://docs.opencv.org/master/da/d54/group\\_\\_imgproc\\_\\_transform.html](https://docs.opencv.org/master/da/d54/group__imgproc__transform.html)
- Otterstätter, G. (1999). *Coloring of Food, Drugs, and Cosmetics* (Food Science and Technology Ser.). Boca Raton: Chapman and Hall/CRC.
- Otterstätter, G. (2007). *Färbung von Lebensmitteln, Arzneimitteln, Kosmetika* (3., überarb. Aufl.). Hamburg: Behr. Verfügbar unter [www.behrs.de/](http://www.behrs.de/)
- DIN, 6730 (01.09.2017). *Papier, Pappe und Faserstoff - Begriffe*. Berlin, Deutschland: Beuth Verlag GmbH.
- Patel, S. K., Cao, J. & Lippert, A. R. (2017). A volumetric three-dimensional digital light photoactivatable dye display. *Nature communications*, 8, 15239.
- Pech-Pacheco, J. L., Cristobal, G., Chamorro-Martinez, J. & Fernandez-Valdivia, J. (2000, September). Diatom autofocusing in brightfield microscopy: a comparative study. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000* (S. 314-317). IEEE Comput. Soc.
- Pelikan. (2020). *Von der Erfindung des Tintenkillers über Funktionsweise und Inhaltsstoffe bis zum Aufbau*. Zugriff am 27.11.2020. Verfügbar unter [www.pelikan.com/pulse/Pulsar/de\\_DE.CMS.displayCMS.209818./tintenkiler-inhaltsstoffe-und-funktionsweise](http://www.pelikan.com/pulse/Pulsar/de_DE.CMS.displayCMS.209818./tintenkiler-inhaltsstoffe-und-funktionsweise)
- Pennanen, K. et al. (2015). European Consumers' Perceptions of Time-Temperature Indicators in Food Packaging. *Packaging Technology and Science*, 28 (4), 303-323.
- Petriashvili, G. et al. (2018). Spiropyran doped rewritable cholesteric liquid crystal polymer film for the generation of quick response codes. *Optical Materials Express*, 8 (12), 3708.
- Pietras. (2017). *Tamper-evident label containing an irreversible photochromic substance*. *International Publication* (WO 2017/112507 A2).
- Pretsch, T. & Fritzsche, N. (2015, 13. Mai). *Temperatur-Indikator zum Nachweis des Überschreitens eines oberen Temperaturschwellwertes innerhalb einer Kühl-Kette*. *Patentamt* (DE 10 2014 104 901 B3).
- Priese, L. (2015). *Computer Vision*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Prusik, T., Abdo, M. & Hohberger, C. (2020, 23. Juli). *Two dimensional barcode with dynamic environmental, data system, method, and apparatus*. *US Patent* (Temptime Corporation, Hrsg.) (US 2016/0292486A1).
- Pudewills, J. (2009). *Entwicklung und Charakterisierung photochromer Sicherheitsinkjettintinten auf Basis von Bakteriorhodopsin*. Zugriff am 29.11.2020. Verfügbar unter <https://archiv.ub.uni-marburg.de/diss/z2010/0131/>

- Ramalho, J. F. C. B. et al. (2019). Luminescence Thermometry on the Route of the Mobile-Based Internet of Things (IoT): How Smart QR Codes Make It Real. *Advanced science (Weinheim, Baden-Wuerttemberg, Germany)*, 6 (19), 1900950.
- Rassart, M., Colomer, J.-F., Tabarrant, T. & Vigneron, J. P. (2008). Diffractive hydrochromic effect in the cuticle of the hercules beetle *Dynastes hercules*. *New Journal of Physics*, 10 (3), 33014.
- Ribi, H. O. (2012, 29. Mai). *Co-topo-polymeric compositions, devices and systems for controlling threshold and delay activation sensitivities*. US Patent (US8187892B2).
- Richter, M. (2011). *Einführung in die Farbmeterik* (Sammlung Göschen). Berlin: De Gruyter.
- Riethmüller et al. (2016). *Thermochrome Pigmente, thermochrome Beschichtung, verfahren zu deren Herstellung sowie deren Verwendung*. Patentamt (DE 10 2014 018 464 A1).
- Roesky, H. W., Möckel, K. & Hoffmann, R. (1996). *Chemische Kabinettstücke. Spektakuläre Experimente und geistreiche Zitate* (1., korrigierter Nachdr. der 1. Aufl.). Weinheim: VCH.
- (RÖMPP, Hrsg.). (2020). *Römpp Lexikon Chemie*, Thieme Gruppe. Zugriff am 27.11.2020. Verfügbar unter [www.roempp.thieme.de/](http://www.roempp.thieme.de/)
- Ruprecht, J. (2001). *Radioaktivität: Berliner Blau als Arzneimittel* (108(26): A-1485 / B-1253 / C-1249). Berlin, Deutschland: Deutsches Ärzteblatt. Verfügbar unter [www.aerzteblatt.de/archiv/95515/Radioaktivitaet-Berliner-Blau-als-Arzneimittel](http://www.aerzteblatt.de/archiv/95515/Radioaktivitaet-Berliner-Blau-als-Arzneimittel)
- Scholz, P. (2005). *Softwareentwicklung eingebetteter Systeme. Grundlagen, Modellierung, Qualitätssicherung* (Xpert.press). Berlin: Springer. Verfügbar unter [www.site.ebrary.com/lib/alltitles/docDetail.action?docID=10182931](http://www.site.ebrary.com/lib/alltitles/docDetail.action?docID=10182931)
- Schwartz, M. M. (Ed.). (2002). *Encyclopedia of smart materials*. New York: Wiley.
- Scotter, M. J. (2015). *Colour additives for foods and beverages* (Woodhead Publishing in food science, technology, and nutrition, no. 279). Cambridge: Woodhead Publishing.
- SFXC. (2020). *SFXC® Irreversible Thermochromic Black Pigment 60°C*. Zugriff am 29.11.2020. Verfügbar unter <https://www.sfxc.co.uk/products/sfxc-irreversible-thermochromic-pigment-60-c-black>
- Sintal Chemie. (2020). *Irreversible Thermochrome Pigmente*. Zugriff am 29.11.2020. Verfügbar unter <https://www.sintal.de/irreversible.html>
- SITA (SITA Messtechnik GmbH, Hrsg.). (2021). *Blasendruckmethode zum Messen der Oberflächenspannung*. Zugriff am 11.06.2021. Verfügbar unter <https://www.sita-process.com/wissen-service/prozessmessgroesse-oberflaechenspannung/blasendruckmethode-zum-messen-der-oberflaechenspannung/>
- Skyrad. (2020). *Skyrad sells irreversible and reversible photochromic and thermochromic pigments*. Zugriff am 29.11.2020. Verfügbar unter [www.skyrad.com](http://www.skyrad.com)
- Smarol. (2020). *Thermochromic and Piezochromic irreversible*. Zugriff am 29.11.2020. Verfügbar unter <http://www.smarol.com/>
- Soga, T., Jimbo, Y., Suzuki, K. & Citterio, D. (2013). Inkjet-printed paper-based colorimetric sensor array for the discrimination of volatile primary amines. *Analytical chemistry*, 85 (19), 8973-8978.

- Steffens, S. (2017). *Intelligent Packaging. Prototyp-Entwicklung eines Matrix-Codes mit Datenvariabilität*. Wuppertal: Druck- und Medientechnologie, Bergische Universität Wuppertal.
- Süße, H. & Rodner, E. (Hrsg.). (2014). *Bildverarbeitung und Objekterkennung*. Wiesbaden: Springer Fachmedien Wiesbaden.
- Suzuki, S. & Abe, K. (1985). Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30 (1), 32-46.
- Szeliski, R. (2011). *Computer Vision*. London: Springer London.
- Tanimoto. (1992). *Water-sensitive coloring sheet*. *US Patent* (5,130,290).
- Tateoka et al. (1990). *Photochromic material containing aspirooxazine and a nitroxy free radical triplet state quencher*. *US Patent* (4,962,013).
- Tenzer, F. (Statista, Hrsg.). (2020, 03. September). *Statistiken zur Smartphone-Nutzung in Deutschland*. Zugriff am 27.11.2020. Verfügbar unter [www.de.statista.com/themen/6137/smartphone-nutzung-in-deutschland/](http://www.de.statista.com/themen/6137/smartphone-nutzung-in-deutschland/)
- Vaillant, R. (2010, 06. April). *Method, system and component for controlling the preservation of a product*. *US Patent* (CRYOLOG SA, Hrsg.) (US7691634B2).
- Vámos, É. & Gábor, E. S. (1973). Quantitative Vitamin-C-Bestimmung mit Hilfe der Berlinerblau-Reaktion und photometrischer Auswertung. 1. Mitt.L-Ascorbinsäurebestimmung in Modell-Lösungen. *Food / Nahrung*, 17 (4), 409-414.
- W3C. (2003). *Portable Network Graphics (PNG) Specification (Second Edition)*. *Information technology — Computer graphics and image processing — Portable Network Graphics (PNG): Functional specification*. *ISO/IEC 15948:2003 (E)*, The World Wide Web Consortium. Zugriff am 27.11.2020. Verfügbar unter [www.w3.org/TR/2003/REC-PNG-20031110/](http://www.w3.org/TR/2003/REC-PNG-20031110/)
- Wagner & Oetken. (2016). *Das elektrochrome Material Berliner Blau in Theorie und Praxis für die Schule und Hochschule*. In: *Nachrichten aus der Chemie* (64. 12). : Gesellschaft Deutscher Chemiker (GDCh). Zugriff am 27.11.2020. Verfügbar unter [www.gdch.de/fileadmin/downloads/Publikationen/Nachrichten\\_aus\\_der\\_Chemie/PDFs/Downloads/berliner\\_blau.pdf](http://www.gdch.de/fileadmin/downloads/Publikationen/Nachrichten_aus_der_Chemie/PDFs/Downloads/berliner_blau.pdf)
- Wang, H., Hussain, M. & Zhou, W. (2017). A novel method for natural dyeing of cotton fabrics with anthocyanin pigments from *Morus rubra* fruits. *Textile Research Journal*, 87 (15), 1896-1911.
- Wang et al. (2015). *Red thermochromic dyes and their nk compositions*. *US Patent* (US 2015/0090159 A1).
- Weston, M., Tjandra, A. D. & Chandrawati, R. (2020). Tuning chromatic response, sensitivity, and specificity of polydiacetylene-based sensors. *Polymer Chemistry*, 11 (2), 166-183.
- Wheeler, R. & Kurpjuweit, T. (Wikimedia, Hrsg.). (2009, 25. März). *QR Code Struktur Beispiel*. Zugriff am 30.11.2020. Verfügbar unter [www.de.m.wikipedia.org/wiki/Datei:QR\\_Code\\_Struktur\\_Beispiel.svg](http://www.de.m.wikipedia.org/wiki/Datei:QR_Code_Struktur_Beispiel.svg)



- Willenbacher, N. & Georgieva, K. (2013). Rheology of Disperse Systems. In U. Bröckel, W. Meier & G. Wagner (Hrsg.), *Product Design and Engineering* (S. 7-49). Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA.
- Winskowicz. (1999). *Golf ball immersion indicator. US Patent* (5,938,544).
- Wu et al. (2016, 28. Juni). *TCDA/znc oxdenanocomposites and film sensors. US Patent* (US 9,376,612 B2).
- Xie, K. (2020). *Programmiertechnische Entwicklung eines Color Management Systems für Smartphone Kameras auf Basis von ICC*. Wuppertal: Druck- und Medientechnologie, Bergische Universität Wuppertal.
- Yam, K. L., Takhistov, P. T. & Miltz, J. (2005). Intelligent Packaging: Concepts and Applications. *Journal of Food Science*, 70 (1), R1-R10.
- Yang, W. et al. (2019). Efficient cycling utilization of solar-thermal energy for thermochromic displays with controllable heat output. *Journal of Materials Chemistry A*, 7 (1), 97-106.
- Yang, Y., Zhou, X.-J., Zhao, N., He, Y.-C., Zhang, G. & Yang, J.-J. (2020). The photochromism of a coordination polymer constructed by Cd(II) and the mono-substituted viologen ligand 1-(3,5-dicarboxylbenzyl)-4,4'-bipyridinium bromide. *Inorganica Chimica Acta*, 502, 119292.
- Yong, J. et al. (2015). Bioinspired transparent underwater superoleophobic and anti-oil surfaces. *Journal of Materials Chemistry A*, 3 (18), 9379-9384.
- Yoon, B. et al. (2013). Inkjet-compatible single-component polydiacetylene precursors for thermochromic paper sensors. *ACS applied materials & interfaces*, 5 (11), 4527-4535.
- Yoon, B., Ham, D.-Y., Yarimaga, O., An, H., Lee, C. W. & Kim, J.-M. (2011). Inkjet printing of conjugated polymer precursors on paper substrates for colorimetric sensing and flexible electrothermochromic display. *Advanced materials (Deerfield Beach, Fla.)*, 23 (46), 5492-5497.
- Yuan, M., Jiang, Q., Liu, K.-K., Singamaneni, S. & Chakrabarty, S. (2018). Towards an Integrated QR Code Biosensor: Light-Driven Sample Acquisition and Bacterial Cellulose Paper Substrate. *IEEE transactions on biomedical circuits and systems*, 12 (3), 452-460.
- Zapka, W. (Ed.). (2018). *Handbook of industrial inkjet printing. A full system approach: Volume 1 & 2*. Weinheim: Wiley-VCH. Verfügbar unter [www.onlinelibrary.wiley.com/book/10.1002/9783527687169](http://www.onlinelibrary.wiley.com/book/10.1002/9783527687169)
- Zhou, Y. & Tang, R.-C. (2016). Modification of curcumin with a reactive UV absorber and its dyeing and functional properties for silk. *Dyes and Pigments*, 134, 203-211.

# Abbildungsverzeichnis

Abbildung 1 Konzeptioneller Architektorentwurf (Bilgin & Backhaus, 2017a) .....	8
Abbildung 2 QCS (Quality Control System) (Aida et al., 2020) .....	16
Abbildung 3 Funktionsdarstellung eines „Upcode“ Mockups (Gligoric et al., 2019) .....	17
Abbildung 4 Monitoring system (Heilmann et al., 2008).....	17
Abbildung 5 Erweiterter 2-D Code (Prusik et al., 2020) .....	18
Abbildung 6 Co-topo-polymeric indicator (Ribi, 2012).....	18
Abbildung 7 Food Sentinel System; Aufbau und Funktion (Carlos, E. et al., 2000) .....	22
Abbildung 8 Freshcode (Nemet & Brand, 2013) .....	23
Abbildung 9 O2xyDot (Havens & Barmore, 2010) .....	24
Abbildung 10 Timestrip Plus (Magen & Isbitsky, 2019) .....	25
Abbildung 11 Traceo (Vaillant, 2010).....	25
Abbildung 12 Phasen der Filtration .....	31
Abbildung 13 Drucksystem Epson EcoTank ET-2650 mit Tintentanksystem (Epson, 2020) .....	32
Abbildung 14 Testform CMYK Farben dargestellt in RGB Volltonfarben .....	33
Abbildung 15 Lichtarten + UV-Lichtquelle .....	34
Abbildung 16 Spektrale Verteilung UV-Lichtquelle.....	35
Abbildung 17 Farbfelder des Macbeth Colorchecker classic und die CIE-Normlichtart D65 .....	38
Abbildung 18 Intelligenter Code unter verschiedenen Lichtquellen .....	40
Abbildung 19 Client-Server-Architektur, angelehnt an Kurose und Ross (2013, S. 85) .....	41
Abbildung 20 Client-Server-Model Peer-to-Peer, angelehnt an Kurose und Ross (2013, S. 85).....	43
Abbildung 21 Elektromagnetisches Spektrum (Frank et al., 2005).....	47
Abbildung 22 Wechselwirkung zwischen Substrat (F) und Photochromen Sensor.....	49
Abbildung 23 Inkjetdruckverfahren (Varianten) .....	50
Abbildung 24 Zwei-Platten-Model ( Mezger, 2016, S. 23) .....	51
Abbildung 25 Micro-Piezo-Druckkopf und Tintenflusskanäle.....	53
Abbildung 26 Piezodüsenplatte samt Düsenausschnitt.....	53
Abbildung 27 Zielparameterfenster für den stabilen Betrieb des DOD-Inkjetdruck (Hoath, 2016, S. 9) ...	54
Abbildung 28 Orbitalmodell (Hädener und Kaufmann, 2006, S. 5) .....	56
Abbildung 29 Aufbauregel.....	57
Abbildung 30 Molekülorbitale und Elektronenübergänge.....	57
Abbildung 31 Strukturformel Patentblau V und Patentblau VF, ohne Hydroxygruppe (-OH, grau) .....	61
Abbildung 32 Konzept eines Tintenlöschers .....	61
Abbildung 33 Konzept einer hydrochromen sensorischen Farbe .....	62
Abbildung 34 Farbveränderung von blau zu transparent .....	62
Abbildung 35 L*-Wert (Patentblau V, hydrochromer Sensor) bei andauernder UV-Licht-Exposition .....	65
Abbildung 36 Lichtechtheit vor und nach der Exposition mit UV-Licht.....	66
Abbildung 37 Remissionskurven Patentblau .....	66
Abbildung 38 Strukturformel Preußisch Blau .....	68
Abbildung 39 Strukturformel Preußisch Blau .....	68
Abbildung 40 Strukturformel Ammoniumeisen(III)-citrat .....	69
Abbildung 41 Strukturformel Kaliumhexacyanidoferrat(III) .....	69
Abbildung 42 Halochrome Farbveränderung von Phenolphthalein.....	72

Abbildung 43 Gruppe von Anthocyanen mit typischen Strukturvariationen .....	72
Abbildung 44 Halochrome Formen von Cyanidin / R = glycosid.....	73
Abbildung 45 Strukturform von Diferuloylmethan / Gepunktet: (Di-)Ketogruppe.....	73
Abbildung 46 Viskositätsanalyse hydrochromer Sensoren; Konzentrationen nach Tabelle 19 .....	75
Abbildung 47 Oberflächenspannung bei verschiedenen Konzentrationen hydrochromer Sensoren .....	76
Abbildung 48 Hydrochromer Sensor und dessen Zielparameterfenster .....	77
Abbildung 49 Partikelgrößenverteilung der Konzentrationen hydrochromer Sensoren .....	77
Abbildung 50 Hydrochrome Partikel auf einem 220 nm Filtermembrane .....	78
Abbildung 51 Spektralphotometrische Untersuchung des hydrochromen Sensors .....	79
Abbildung 52 Viskositätsanalyse aller Konzentrationen photochromer Sensoren .....	79
Abbildung 53 Oberflächenspannung der verschiedenen Konzentrationen photochromer Sensoren .....	80
Abbildung 54 Photochromer Sensor und dessen Zielparameterfenster .....	81
Abbildung 55 Mikroskopische Aufsichtsaufnahme des photochromen Sensors .....	82
Abbildung 56 REM Aufnahme von der Filtermembran aufgefangenen photochrome Partikel.....	82
Abbildung 57 Farbveränderung des photochromen Sensors unter verschiedenen Lichtquellen .....	83
Abbildung 58 Lichtquellenbedingte L*-Wertveränderung (Tabelle 22) des photochromen Sensors .....	83
Abbildung 59 TL84, D65 und UV im Vergleich und ihre Exposition auf den photochromen Sensor .....	84
Abbildung 60 Viskositätsanalyse aller halochromen Konzentrationen im Vergleich.....	85
Abbildung 61 Oberflächenspannung verschiedener halochromer Konzentrationen .....	85
Abbildung 62 Halochromer Sensor und dessen Zielparameterfenster .....	86
Abbildung 63 Ungelöste Diferuloylmethan Partikel.....	87
Abbildung 64 pH-Wert Veränderungen halochromer Sensoren .....	87
Abbildung 65 Einflusspotential der Substrate auf das Farbwechselverhalten des hydrochromen Sensors ....	88
Abbildung 66 Einflusspotential der Substrate auf das Farbwechselverhalten des photochromen Sensors ....	89
Abbildung 67 Einflusspotential der Substrate auf das Farbwechselverhalten des halochromen Sensors.....	90
Abbildung 68 DataMatrix Code.....	93
Abbildung 69 DataMatrix Code Anordnungsschema und Resultat (ISO/IEC 16022, 2006, S. 23) .....	94
Abbildung 70 Funktionsweise von CCD- und CMOS-Sensoren (Hessler et al., 2003) .....	96
Abbildung 71 Für CCDs verwendete Farbmuster (Klemm, 2010) .....	97
Abbildung 72 Aufbau eines beispielhaften QR-Codes nach (Wheeler & Kurpjuweit, 2009).....	97
Abbildung 73 2D-Codes und ihre Suchmuster .....	98
Abbildung 74 Codeerkennungsprozess .....	99
Abbildung 75 Erkennbarkeit von Suchmustern .....	100
Abbildung 76 Varianten quadratförmiger Suchmuster .....	103
Abbildung 77 Gestaltungsraster auf Basis von Modulen.....	104
Abbildung 78 Aufbau eines intelligenten Codes.....	104
Abbildung 79 Computer Vision (Balaban, 2018) .....	106
Abbildung 80 Bild als Array .....	107
Abbildung 81 Prinzip der Erosion (Burger & Burge, 2005) .....	109
Abbildung 82 Prinzip der Dilatation (Burger & Burge, 2005) .....	109
Abbildung 83 Dualraum: Ortsraum und Parameterraum, mit Schnittpunkt (H) (OpenCV, 2020d).....	111
Abbildung 84 (a) äußerer Rand, (b) Lochrand (Suzuki & Abe, 1985, S. 35) .....	112
Abbildung 85 Softwarearchitektur (Legende in Tabelle 13) .....	115
Abbildung 86 Erkennungsprozess des intelligenten Codes .....	117
Abbildung 87 Browserkompatibilität der eingesetzten Funktionen (developer.mozilla, 2020) .....	117
Abbildung 88 Bestandteile des intelligenten Codes .....	122

Abbildung 89 Ampelsystem fürs Eingrenzen des Datensatzes .....	126
Abbildung 90 Funktionsweise des Umgebungslicht-Assistenten.....	127
Abbildung 91 sRGB, Adobe RGB (1998) und DCI-P3 im Vergleich (Konica Minolta, 2020).....	134
Abbildung 92 Flussdiagramm Colormanagement .....	135
Abbildung 93 Zugriff auf Mediengeräte .....	138
Abbildung 94 Hardwarearchitektur angelehnt an Kurose und Ross (2013, S. 85) .....	140
Abbildung 95 Wartet auf Zugriffsberechtigung .....	141
Abbildung 96 Zeigt aktive Kamera und Interface im Browser Firefox an.....	141
Abbildung 97 Scanübersicht mit Historie des intelligenten Codes .....	141
Abbildung 98 Detailansicht eines Vergleichs zwischen zwei Messzeitpunkten .....	142
Abbildung 99 Responsive Webinterface und Codeerfassung samt Statussymbole .....	142
Abbildung 100 Vollständig gedruckter intelligenter Code, links: nicht reagiert und rechts: reagiert.....	144

# Tabellenverzeichnis

Tabelle 1 Notation der verwendeten Zeichen und Symbole .....	V
Tabelle 2 Abkürzungsverzeichnis .....	VI
Tabelle 3 Irreversible Sensorische Farben und die Hervorhebung ihrer Inkjetfähigkeit (grau) .....	12
Tabelle 4 Veröffentlichungen und Patente Intelligenter Code-Technologien .....	15
Tabelle 5 Kommerzielle Smart Labels .....	21
Tabelle 6 Materialien für die Untersuchung einer photochromen Tinte.....	28
Tabelle 7 Materialien für die Untersuchung einer halochromen Tinte .....	28
Tabelle 8 Materialien für die Untersuchung einer hydrochromen Tinte.....	28
Tabelle 9 Additive für den möglichen Einsatz in den entwickelten sensorischen Tinten .....	29
Tabelle 10 Untersuchte Substrate .....	29
Tabelle 11 Materialien für den Filtrationsprozess PVDF (Hydrophilic Polyvinylidene Fluoride).....	32
Tabelle 12 Vergleich verschiedener Smartphone-Marken.....	38
Tabelle 13 Komponenten der Softwarearchitektur.....	42
Tabelle 14 Wahrnehmungsschwellen im L*a*b*-Farbraum (Bühler et al., 2018, S.16).....	49
Tabelle 15 Quantenzahlen in der Übersicht.....	56
Tabelle 16 Qualitative Bewertung der Ausprägungen.....	63
Tabelle 17 Identifizierung eines geeigneten Lösungsmittels für Patentblau V .....	64
Tabelle 18 Reduktionsmittel für die Entfärbung von Patentblau V .....	64
Tabelle 19 Hydrochrome Stoffkonzentrationen .....	64
Tabelle 20 Photochrome Stoffkonzentrationen.....	70
Tabelle 21 Halochrome Stoffkonzentrationen .....	74
Tabelle 22 Farbveränderung des photochromen Sensors unter verschiedenen Lichtquellen .....	83
Tabelle 23 L*a*b*-Werte der Einflusspotentiale auf den hydrochromen Sensor in Abbildung 65 .....	88
Tabelle 24 L*a*b*-Werte der Einflusspotentiale auf den photochromen Sensor in Abbildung 66 .....	89
Tabelle 25 L*a*b*-Werte der Einflusspotentiale auf den halochromen Sensor in Abbildung 67.....	90
Tabelle 26 Farbunterschiede der Testchart (Kapitel 3.7.3) verschiedener Smartphones .....	135
Tabelle 27 TCP/IP und ISO/OSI Referenzmodell im Vergleich .....	136
Tabelle 28 Geräteeigenschaften Rheometer .....	151
Tabelle 29 Geräteeigenschaften Blasendrucktensiometer.....	151
Tabelle 30 Geräteeigenschaften Spektralradiometer .....	151
Tabelle 31 Geräteeinstellungen Spektral-Densitometer.....	151
Tabelle 32 Geräteeinstellungen Spektral-Densitometer.....	151
Tabelle 33 Geräteeigenschaften Konfokales 3D Laserscanning Mikroskop .....	152

# Veröffentlichungen

## *Buchkapitel*

Bilgin, M. (2021). *Halochromic Composite Materials*. In Encyclopedia of Materials: Composites. Amsterdam, Niederlande: Elsevier B.V. DOI: 10.1016/B978-0-12-803581-8.11873-9

## *Journalbeitrag*

Bilgin, M., Backhaus, J. (2020). *Development of an irreversible Hydrochromic Ink for Smart Packaging*. Stuttgart, Deutschland: Journal of Print and Media Technology Research. DOI: 10.14622/JPMTR-2004

## *Journalbeitrag*

Bilgin, M., Backhaus, J. (2020). *Hydrochromic Ink as a Security Feature for Piezoelectric Inkjet Printing*. Issue No 12 2020. Toronto, Kanada: International Circular of Graphic Education and Research, IC. ISSN: 1868-0879 (Online), 1868-0712 (Print)

## *Vortrag, Paper*

Bilgin, M., Backhaus, J. (2020). *Printable smart materials used as sensors for continuous monitoring in a smart code*. Internationale Konferenz: 01.04. – 04.04.2020. Oklahoma City, Ok, USA: IEEE Green Technologies (Online Konferenz). DOI: 10.1109/GreenTech46478.2020

## *Vortrag, Paper*

Bilgin, M., Backhaus, J. (2020). *Development process of a hydrochromic ink*. Internationale Konferenz: 15.03. – 18.03.2020. Oklahoma City, Ok, USA: TAGA – Technical Association for the Graphic Arts (Online Konferenz)

## *Vortrag, Paper*

Bilgin, M., Backhaus, J. (2019). *Smart Materials Detection Using Computer Vision*. Internationale Konferenz: 15.09. – 18.09.2019. Stuttgart, Deutschland: Advances in Printing and Media Technology. Advances Vol.46 (2019), iarigai. ISBN: 978-3-948039-01-1, ISSN: 2409-4021 (Online). DOI: 10.14622/Advances\_46\_2019

## *Journalbeitrag*

Bilgin, M., Backhaus, J. (2019). *Smart packaging by the application of sensitive dyes*. Stuttgart, Deutschland: Journal of Print and Media Technology Research. JPMTR 123 | 1901 Research paper. DOI 10.14622/JPMTR-1901, UDC 621.798-027-035.67

## *Vortrag, Paper*

Bilgin, M., Backhaus, J. (2019). *Development Process of a Smart Code Based On Smart Materials*. Internationale Konferenz: 17.03. – 20.03.2019. Minneapolis, MN, USA: TAGA – Technical Association for the Graphic Arts industries

## *Journalbeitrag*

Bilgin, M., Backhaus, J. (2018). *Initial Steps to Develop Piezoelectric Inkjet Ink Containing Stimuli-responsive Pigments*. Toronto, Kanada: International Circular of Graphic Education and Research, IC Journal. Issue No.11, 2018. ISSN: 1868-0879 (Online), 1868-0712 (Print)

## *Vortrag, Paper*

Bilgin, M., Backhaus, J. (2018). *Development of a unidirectional switchable Photochromic Ink for Smart Packaging*. Internationale Konferenz: 03.10. – 07.10.2018. Warschau, Polen: Advances in Printing and Media Technology. Advances Vol.45 (2018), iarigai. ISBN: 978-3-948039-00-4, ISSN: 2409-4021 (Online)

*Vortrag, Paper*

Bilgin, M., Backhaus, J. (2018). *First Steps to Develop Molecular Switches for Piezoelectric Inkjet*. Internationale Konferenz: 18.03. – 21.03.2018. Baltimore, Maryland, USA: TAGA – Technical Association for the Graphic Arts industries

*Vortrag, Paper*

Bilgin, M., Backhaus, J. (2017). *Smart Packages by Means of Intelligent Codes*. Internationale Konferenz: 10.09. – 13.09.2017. Fribourg, Schweiz: Advances in Printing and Media Technology. Advances Vol.44 (2017), iarigai. ISBN 978-3-9870704-1-9, ISSN 2409-4021 (Online)

*Vortrag, Paper*

Bilgin, M., Backhaus, J. (2017). *Intelligent codes by controlled responsiveness to external stimuli*. Internationale Konferenz: 04.10. – 06.10.2017. Printing Future Days (PFD). Chemnitz, Deutschland: VWB – Verlag für Wissenschaft und Bildung. ISBN: 978-3-86135-627-1, p.85-90

Der Lebenslauf ist in der Online-Version aus Gründen des Datenschutzes nicht enthalten.





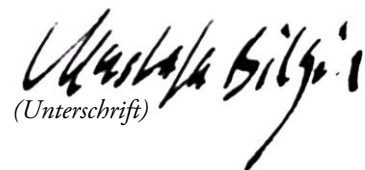
BILGIN, MUSTAFA  
(Name, Vorname)

## Erklärung

gem. § 7 Abs. 2 Promotionsordnung vom 29. Januar 2001

Hiermit erkläre ich, dass ich die von mir eingereichte Dissertation selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Stellen der Dissertation, die anderen Werken dem Wortlaut oder Sinn nach entnommen wurden, in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe. Entsprechendes gilt für beigegebene Zeichnungen und Darstellungen. Zudem erkläre ich, dass es sich um meine erste Promotion handelt und keine früheren Promotionsanträge angestrebt wurden.

31. Dezember 2020  
(Datum)

  
(Unterschrift)