

Inhalt

Inhalt 1

Abbildungsverzeichnis.....	3
Abkürzungsverzeichnis.....	4
0 Übersicht	5
0.1 Motivation	5
0.2 Zielsetzung	5
0.3 Kapitelübersicht.....	6
1 Grundlagen.....	8
1.1 Licht / Optik.....	8
1.1.1 Naturgesetze	8
1.1.2 Lichtquellen.....	11
1.1.3 Schattenwurf.....	13
1.1.4 Mechanismus des Sehens	13
1.1.5 Parameter	14
1.1.6 (Video-) Kamera	15
1.2 Schrift- / Texterkennung	16
1.2.1 OCR (Optical Character Recognition).....	17
1.2.2 HCR (Handprint Character Recognition)	17
1.2.3 Übergreifendes	18
1.3 Programmiersprache.....	18
1.4 Der Hardwareschnittstellen (Stecker)	18
2 Analyse der Aufgabenbedingungen	20
3 Vorhandene Geräte und Annahmen.....	22
3.1 System 1: Keyence.....	22
3.1.1 Kameras.....	22
3.1.2 Beleuchtung.....	23
3.1.3 Filter	24
3.1.4 Steuereinheiten, Kabel, Stecker, Befestigung	24
3.1.5 Software.....	24
3.2 System 2: TheImageSource, Tesseract.....	26
3.2.1 Vorabtests.....	26

4	Erkundungsaufbau	29
4.1	<i>Ziel</i>	29
4.2	<i>Material</i>	29
4.3	<i>Aufbau.....</i>	29
4.4	<i>Durchführung.....</i>	30
4.5	<i>Ergebnisse</i>	30
5	Experiment 1 „Licht und Schatten“:	31
5.1	<i>Ziel</i>	31
5.2	<i>Material / Gegenstände / Software.....</i>	31
5.3	<i>Aufbau.....</i>	33
5.4	<i>Durchführung.....</i>	34
5.5	<i>Ergebnisse</i>	35
6	Experiment 2 „Textstile“:.....	36
6.1	<i>Ziel</i>	36
6.2	<i>Material</i>	36
6.3	<i>Aufbau / Durchführung.....</i>	37
6.4	<i>Ergebnisse</i>	37
7	Experiment 3 "Sonderlaute":	38
7.1	<i>Ziel</i>	38
7.2	<i>Material / Aufbau.....</i>	38
7.3	<i>Durchführung</i>	38
7.4	<i>Ergebnisse</i>	38
8	Programmierung	39
Literatur		41
Anlagen		45
Anlagen, Teil 1		47
Anlagen, Teil 2		57
Anlagen, Teil 3		58
Selbstständigkeitserklärung		61

Abbildungsverzeichnis

Abbildung 4-1	8
Abbildung 4-2	9
Abbildung 4-3	10
Abbildung 4-4	10
Abbildung 4-5	12
Abbildung 4-6	13
Abbildung 4-7	13
Abbildung 4-8	14

Abkürzungsverzeichnis

API	Application Programming Interface
bzw.	Beziehungsweise
CCD	Charge Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
CSV	Comma Separated Values
etc.	et cetera
HCR	Handprint Character Recognition
i.O.	in Ordnung
IP	Internet Protocol
IR	Infra Rot
ISO/IEC	International Organization for Standardization / International Electro-technical Commision
LAN	Local Area Network
LED	Light Emitting Diode
MP	MegaPixel
N.i.O.	Nicht in Ordnung
o.ä.	oder ähnliches
OCR	Optical Character Recognition
PDF	Portable Document Format
QR	Quick Response
SPS	SpeicherProgrammierbare Steuerung
u.a.	unter anderem
Usw.	und so weiter
UV	Ultra Violett
z.B.	zum Beispiel
zw.	Zwischen

0 Übersicht

Im einleitenden Kapitel werden die Motivation und die Aufgabenstellung dieser Diplomarbeit besprochen. Gleichzeitig erfolgt ein kurzer Überblick zu den einzelnen Kapiteln dieser Arbeit.

0.1 Motivation

Für eine Firma ist wirtschaftliche Produktivität das wichtigste um am hart umkämpften Kartenmarkt zu überleben. Der Kunde besteht auf ein 100-prozentig fehlerfreies Produkt von spitzen Qualität. Bei dem Unternehmen Plasticard ZFT werden Karten jeder Art hergestellt und individualisiert. Des Weiteren wird die Lettershop als Dienstleistung angeboten. Die aktuelle Mailing-Strecke bei Plasticard ZFT ist für eine 100% Prüfung der Briefe nicht ausgestattet. Es ist eine Erweiterung der Maschine geplant, die es ermöglicht, fertige Briefe zu 100% auf korrektes Gewicht und Anschreiben zu prüfen. Zusätzlich soll das System automatisiert, nicht korrekte Briefe aussteuern.

Zum Verringern der Kosten für das Kontrollsystem sollen hauptsächlich Geräte und Systeme verwendet werden, welche bereits vorhanden sind. Besondere Herausforderung ist es, die verschiedenen Hard- und Softwareschnittstellen zu entwickeln.

0.2 Zielsetzung

Hauptschwerpunkt der vorliegenden Bachelorarbeit ist die Schrift- / Texterkennung. Bevor die im Betrieb vorhandenen Geräte aufgezeigt werden, wird ein Blick auf die benötigten Hard- und Softwaregrundlagen ermöglicht.

Ein weiterer Schwerpunkt ist die fehlerlose, schnelle Kommunikation zwischen unterschiedlichen Systemen. Die Daten aus der Texterkennung sollen in einer Datenbank und in einem CSV-File gespeichert werden und anschließend über einen noch zu bestimmenden Port zu der übergeordneten Recheneinheit übertragen werden.

Zuvor müssen die Parameter des praktischen Texterkennungssystems erfasst werden. Dazu gehören z. B. der minimale und der maximale Kontrast des Briefes, die verwendbare Beleuchtung oder die Objektschärfe. Diese Kennwerte müssen aus den Datenblättern der bestehenden Kameras, Lichtquellen und Steuereinheiten evaluiert werden.

Die so erhaltenen Daten müssen analysiert werden, um zu erfahren, welche die beste Komponenten-Kombination ist. Die Weiterentwicklung des gefundenen Systems schließt sich nahtlos an.

Die Kosten für die noch zu erwerbenden Komponenten der Weiterentwicklung werden ebenso Thema sein, wie die Analyse wichtiger Parameter des Gesamtsystems nach der Entwicklung. Zu diesen Parametern gehören unter anderem der benötigte Platz, die Machbarkeit, die Schnittstellen (Soft- und Hardware), Arbeitsgeschwindigkeit.

Nachdem die Lösung der vorangegangenen Punkte erfolgt ist, wird das Gesamtsystem in die vorhandene Mailing-Maschine (Neopost) implementiert.

0.3 Kapitelübersicht

Insgesamt neun Kapitel umfassen diese Bachelorarbeit.

Den Start bildet das **Kapitel 0**, welches diese Zeile enthält. Dort werden die Aufgabenstellung und ihre Gründe erörtert. Die Kurzinhalte über die einzelnen Kapitel beenden den Gliederungspunkt 0.

Das folgende **Kapitel 1** enthält Grundwissen zu drei verschiedenen Themenbereichen. Der Aufbau von Licht und die Möglichkeiten Licht zu beeinflussen werden im ersten Unterkapitel erläutert. Zusätzlich enthält dieser Abschnitt Informationen über Kameras, deren groben Aufbau und wichtige Eigenschaften. Wie die Schrift- und Texterkennung unterteilt wird, wie sie funktioniert erörtert der Gliederungspunkt 1.2 Schrift- / Texterkennung. Zuletzt werden einige Programmiersprachen analysiert.

Die Umgebungs- und Systemparameter der Aufgabenstellung werden im **Kapitel 2** ermittelt und gesammelt. Hier werden die Voraussetzungen für das erfolgreiche Lösen der gesamten Aufgabe dargelegt.

Nach erfolgter Aufgabenstellung werden im **Kapitel 3** bestehende Geräte (Kameras, Lichtquellen, ...) analysiert. Dabei werden die Geräte zu Systemen zusammengefasst. Jedes System ist so konzipiert, dass eine Texterkennung und –auswertung möglich ist.

Kapitel 4 beschreibt den ersten Versuch. Dieser dient der Erkundung und Erforschung des zweiten, in Kapitel 3 beschriebenen Systems. Dabei werden die Möglichkeiten und Grenzen der Kamera DMK23GP031 von TheImageSource untersucht.

Das Experiment 1 „Licht und Schatten“ ist in **Kapitel 5** niedergeschrieben. Ziel ist es die beste Lichtquelle für das System „TheImageSource“ zu finden.

Welche Textgestaltungen für die Adressen verwendet werden können, damit das System „TheImageSource“ diese korrekt erkennt, wird in **Kapitel 6** erörtert.

Kapitel 7 beschäftigt sich mit dem Problem der Sonderlaute.

Wie das Programm für die Kamera DMK23GP031 von TheImageSource konzipiert wurde enthält **Kapitel 8**.

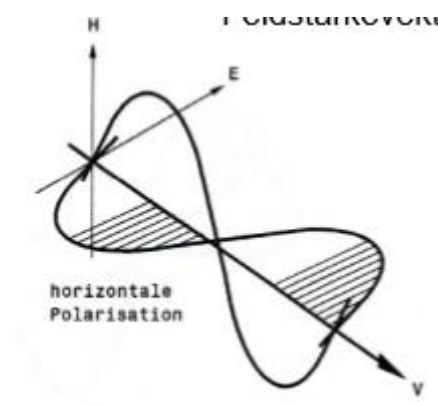
1 Grundlagen

1.1 Licht / Optik

Ohne Licht kein „Sehen“!

1.1.1 Naturgesetze

Das Licht ist ein Vertreter der elektromagnetischen Wellen. Wie bei elektromagnetischen Wellen üblich wird die Lichtenergie durch ein sich dynamisch änderndes gekoppeltes elektromagnetisches Feld übertragen.



H ... Magnetische Feldstärke

E ... Elektrische Feldstärke

V ... Ausbreitungsrichtung

Abbildung 4-1

So breitet sich Licht durch Verändern des nächsten Atoms aus. Durch die geringe Wellenlänge müssen nur die Elektronen oder Quanten des Atoms um einige Energieniveaus verschoben werden. Fallen diese wieder zurück, wird eine sehr geringe Menge Licht emittiert und mit dieser Energie werden wiederum die Nachbaratome in Bewegungsrichtung angeregt. Erst durch die Addition der emittierten Lichtenergien vieler Atome wird die Leistung groß genug, um von einem menschlichen Auge wahrgenommen zu werden. Wird ein Strahlbild von einem Strahl (siehe nächster Absatz) erzeugt, so bilden die Atome, welche gleichzeitig Licht emittieren eine gebogene Fläche, welche einem Halbkreis ähnelt, vorausgesetzt es sind keine Hindernisse in der Nähe. (Huygenssches Prinzip [Born1985])

In dieser Arbeit wird zwischen Strahl und Lichtstrahl unterschieden. Ein Strahl bildet hierbei den Vektor für die Hauptbewegungsrichtung der elektromagnetischen Welle ab. Aus mehreren parallelen Strahlen, deren zugehörige Wellen sich teilweise überlagern, kann ein Lichtstrahl entstehen (Ausdehnung in Bewegungsrichtung >> Ausdehnung in andere Richtungen).

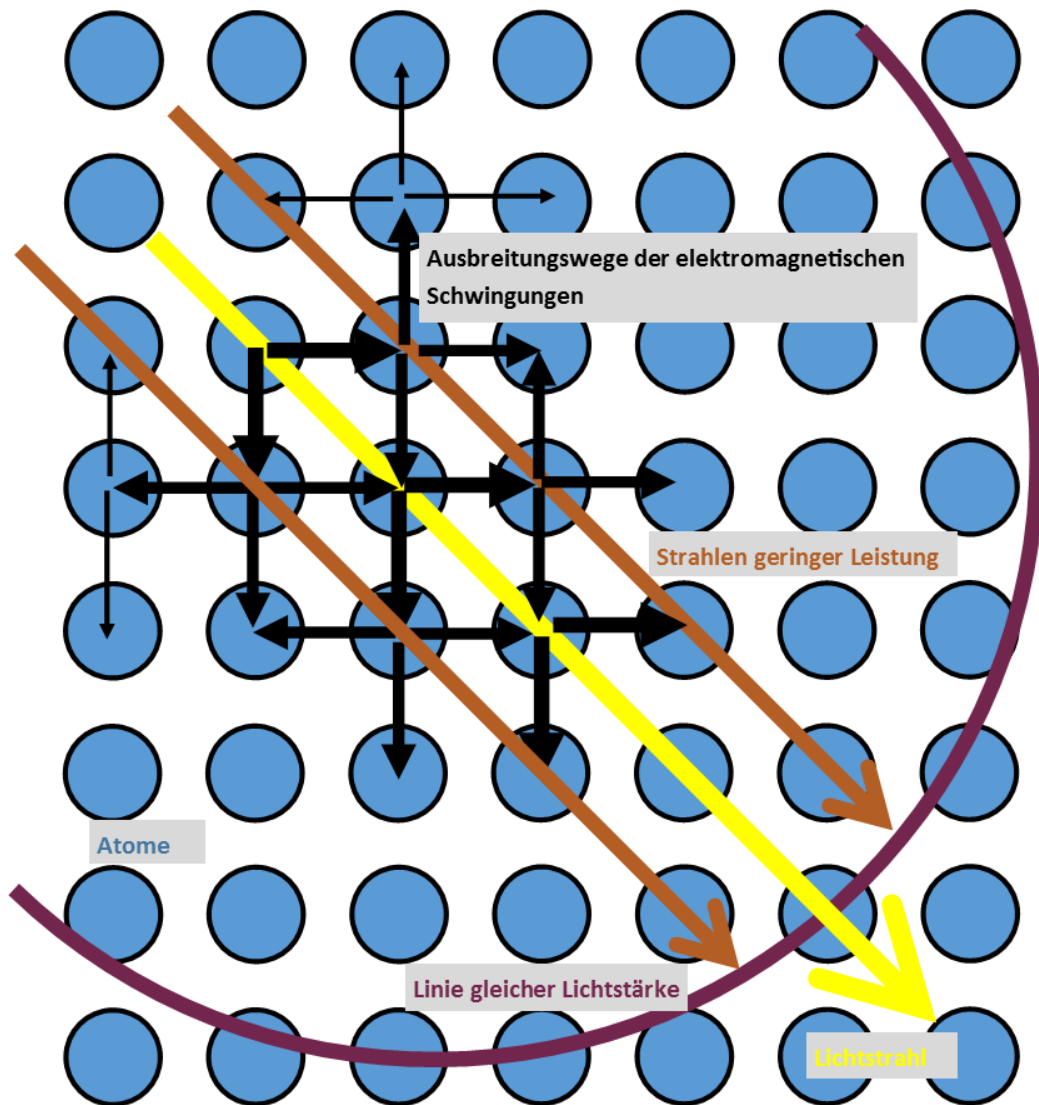


Abbildung 4-2

Wie in [LiTa1997] beschrieben, ist die Lichtgeschwindigkeit (c_0) als eine materialabhängige Naturkonstante anzusehen, deren Wert im Vakuum auf $2,997\,924\,58 \cdot 10^8 \text{ m s}^{-1}$ berechnet wurde. Die daraus resultierenden Übergangseffekte werden im Abschnitt Brechung erklärt.

„Das Licht der verschiedenen Quellen kann monochromatisch bzw. spektral kontinuierlich verteilt sein oder aus einzelnen Spektrallinien bestehen.“ [PeBa2005]

Monochromatische Kameras können zum Beispiel nur Schwarz-Weiß-Bilder aufnehmen. Diese unterscheiden die Wellenlänge nicht, sondern registrieren nur, ob Licht einfällt oder nicht.

Für die Betrachtung des Brechungsverhaltens wird das Teilchenmodell angenommen. Danach bewegt sich ein Lichtteilchen nur auf geraden Flugbahnen, sogenannten Lichtstrahlen. Die Bahnen können durch Hindernisse geknickt werden. [Youn1997] formulierte die Tatsache der Brechung in wenigen Worten wie folgt: „Wenn ein Lichtstrahl unter einem Winkel auf eine ebene Grenzfläche zwischen zwei transparenten Medien trifft, wird er gebrochen.“ Warum dies nur für $i > \arcsin(n'/n)$ gilt wird zwei Absätze später im Themengebiet Totalreflexion näher erläutert. Zur besseren Bestimmung der Stärke der Brechung wurde der Brechungsindex n eingeführt. Dieser ist materialbezogen und wurde für Vakuum willkürlich gleich eins gesetzt. Die Luft und das Vakuum besitzen hinsichtlich der für die Ausbreitung des Lichtes wichtigen Parameter die gleichen Werte. $n=1$. Außerdem sei die Abhängigkeit des Brechungsindex von der Wellenlänge des Lichtes erwähnt. Die Brechungskoeffizienten n und n' verhalten sich wie der Sinus der Brechungswinkel i' und i .

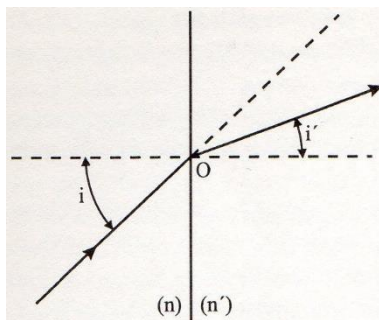


Abbildung 4-4

$$\frac{n}{n'} = \frac{\sin i'}{\sin i}$$

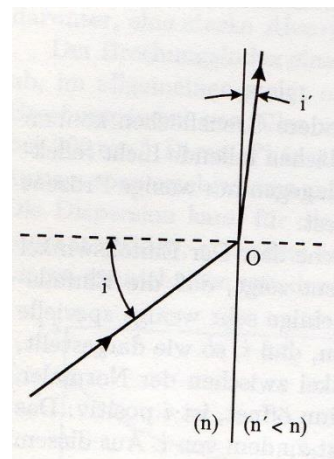


Abbildung 4-3

Das Teilchenmodell ist auch für die Betrachtung des Reflexionsverhaltens zu bevorzugen, da es deutliche Vereinfachungen bringt und das Ergebnis nicht verfälscht. Je glatter eine Oberfläche ist, desto größer wird der reflektierende Anteil eines farbigen (undurchsichtig) Gegenstandes. Eine polierte Metallplatte zum Beispiel reflektiert besser als eine oxidierte Metallplatte oder eine Glasplatte. Wird eine Senkrechte zur Oberfläche des reflektierenden Gegenstandes gebildet und die Winkel zwischen dieser Geraden und den einfallenden bzw. reflektierten Strahlen gemessen, so sind die Winkel dem Betrag nach identisch.

Von Totalreflexion wird nur gesprochen, wenn der Lichtstrahl reflektiert wird, obwohl das reflektierende Material eigentlich lichtdurchlässig ist. Damit solch ein Material reflektiert, muss der Strahl aus einem höherbrechenden Material (mit größerem n) kommen. Außerdem muss der Einfallswinkel so geartet sein, dass nach Anwendung des Brechungsgesetzes der Brechungswinkel größer 90° werden würde. Dies ist unmöglich, da der gebrochene Strahl sich dann wieder im Ausgangsmedium befände. Da die Lichtenergie irgendwo hin weiterübertragen werden muss, wird aus

der Brechung eine Reflexion. Das reflektierte Licht erfüllt [dabei] das Reflexionsgesetz. [Youn1997] Grafisch dargestellt wird dies in der Abbildung 4-3 (siehe oben).

$$i' > 90^\circ \text{ wenn } i > \sin^{-1} \frac{n'}{n}$$

Weitere Gesetzmäßigkeiten können in den im Literaturverzeichnis genannten Büchern zum Thema Optik nachgelesen werden.

„Obwohl Wellen sich geradlinig im Raum fortpflanzen, kann man beobachten, daß sie sich auch in den geometrischen Schatten hinter Hindernissen ausbreiten. Diese Erscheinung wird als Beugung bezeichnet und ist wie die Interferenz ein typisches Wellenphänomen.“ [Kühl2004]

1.1.2 Lichtquellen

Die Lichtquellen werden in zwei Bereiche unterteilt. Der erste Bereich sind die natürlichen Quellen. Beispiele aus diesem Segment sind die Sonne, Streuungserscheinungen im Himmel und brennendes Holz. Das Sonnenlicht und die Streuungserscheinungen sind kostenlos und benötigen keine von Menschen gebaute Energiequelle. Jedoch ist ihre Existenz wetterabhängig und kann daher nicht garantiert werden. Quellen, wie brennendes Holz hingegen erzeugen zusätzlich viel Wärme und Abgase. Des Weiteren müssen diese in feuerfesten Behältnissen kontrolliert werden. Die Beherrschung der natürlichen Lichtquellen ist sehr schwierig.

Vertreter des zweiten Bereichs werden künstliche Quellen genannt. Dazu gehören laut [PeBa2005] Temperaturstrahler und Lumineszenzstrahler. Die Temperaturstrahler untergliedern sich nach derzeitigem Stand in Glühlampen, Globalquellen, schwarze Strahler und Hochintensive Lichtbogenquellen. Die Gruppe der Lumineszenzstrahler umfasst die Entladungslampen, die Leuchtstoffröhren, die Halbleiterdioden (LEDs¹, UV-Dioden², IR-Dioden³) und die kohärenten Lichtquellen (Laser). Lumineszenzstrahler sind effizienter als Temperaturstrahler, denn diese wandeln nur einen geringen Anteil der zugeführten Energie in Wärmeenergie. Bis auf die Globalquellen werden alle künstlichen Leuchtquellen mit elektrischem Strom gespeist.

Der Grundvorgang der Lichtentstehung in allen anderen elektrisch betriebenen Leuchtquellen ist ähnlich. Licht entsteht, wenn einem Elektron Energie zugeführt wird, dieses dadurch einige Energieniveaus ansteigt und anschließend wieder zurück fällt. Auf dem Rückweg muss das Elektron jene Energie wieder abgeben. Je

¹ LED, Light Emitting Diode

² UV, Ultra Violett

³ IR, Infra Rot

nach zurückgelegtem Energieunterschied entsteht Licht mit definierten Wellenlängen.

Es unterscheiden sich nur die Materialien in denen die Lichtentstehung geschieht. In Glühlampen ist es ein Metallfaden, in Leuchtstoffröhren ein Gas (-gemisch), in LEDs ein Halbleiterkristall, usw.

Die Atomrümpfe des Metalls schwingen immer am gleichen Ort. Die fließenden Elektronen des Stroms kommen unterschiedlich weit, bevor sie mit Atomen zusammen stoßen. So entsteht gleichmäßig verteiltes, weißes Licht, welches sich aus unterschiedlichen Wellenlängen zusammensetzt.

In Gasen sind die Atome weiter voneinander entfernt. Dies reduziert die Wahrscheinlichkeit auf Zusammenstöße. Des Weiteren gibt es keine direkte elektrische Leitung. Diese muss erst durch Stoßionisation aufgebaut werden. Dabei werden den Gasatomen Elektronen weggenommen. So entstehen Ionen. Aus der Stromquelle kommen neue Elektronen. Die Ionen saugen diese an. Dadurch entsteht Licht in Gasen. Bevor die nächsten Elektronen Licht erzeugen können, müssen neue Ionen geschaffen werden. Das von Leuchtstoffröhren entsandte Licht ist zeitlich inkonsistent. Dies bedeutet hellere und dunklere Phasen wechseln sich ab.

In Halbleitern wird das Licht genauso erzeugt, wie in Wolfram. Daher ergeben sich die gleichen Eigenschaften. Durch die Bauweise (Halterung des Kristalls) der LEDs ist der Abstrahlwinkel deutlich geringer als bei Glühlampen. Des Weiteren gewährt die gleichmäßige Bauart des Kristalls ein fast paralleles Licht.

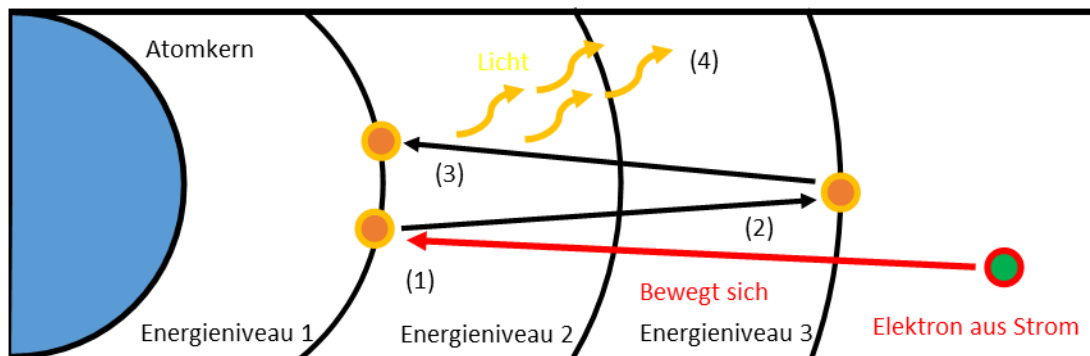


Abbildung 4-5

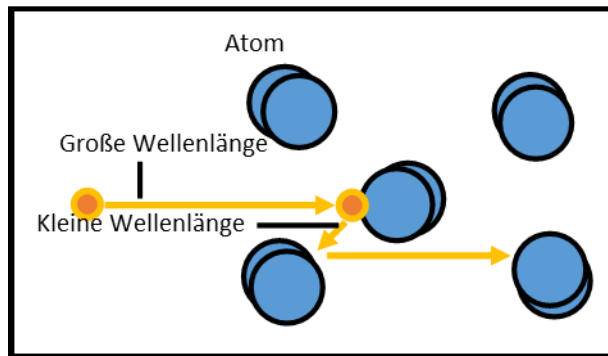


Abbildung 4-6

1.1.3 Schattenwurf

Als sich der Name Schatten entwickelt hat, ging man davon aus, dass Licht sich nur geradlinig ausbreiten kann. Somit konnte eine Lichtquelle nur Gebiete beleuchten, welche von einem geradlinigen Strahl erreicht werden konnten, welcher höchstens von einem durchsichtigen Gegenstand abgelenkt wird. Die so entstehenden, von der Leuchtquelle unerreichbaren Orte nannte man Schatten. Diese erscheinen wesentlich dunkler, als die beleuchteten Regionen.

Die Wellentheorie hat jedoch bewiesen, dass Licht auch gebeugt werden kann und damit auch in Schattenbereich vordringen kann. Daher kann ein lichtleerer Raum nur erzeugt werden, wenn er von allen Seiten hermetisch durch lichtundurchlässige Materialien geschirmt wird.

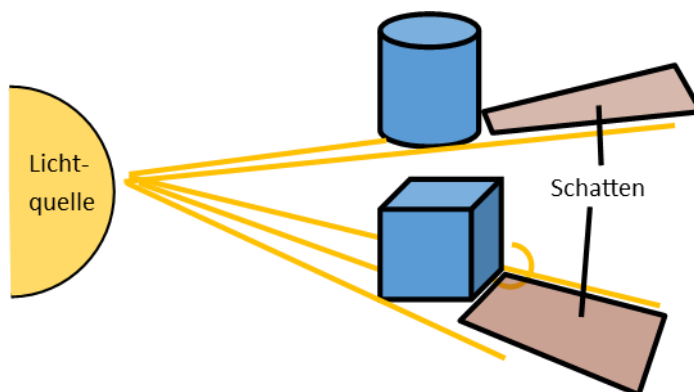


Abbildung 4-7

1.1.4 Mechanismus des Sehens

Das Licht wird von einer Quelle abgestrahlt und trifft auf ein Objekt. Dieses Objekt absorbiert einen Teil des Lichtes (z.B. den blauen und den roten Anteil), lässt einen Teil durch (siehe Brechung) und der übrig gebliebene Teil wird zum Betrachter reflektiert. Dabei ist zu beachten: „Die Lichtstrahlen sind farblos.“ Und „Das Sehorgan macht daraus erst eine Information, indem es Farbempfindung produziert.“

[Küpp1992]

Der Betrachter kann zum Beispiel ein menschliches Auge sein. Die Linse des Auges lenkt das aus unterschiedlichen Richtungen kommende Licht auf einen kleinen Punkt, an dem sich die Sinneszellen des Sehens befinden. Es gibt zwei Arten von Sinneszellen. Die Stäbchen benötigen nur sehr geringe Lichtmengen, können jedoch auch die Helligkeit bestimmen (viel oder wenig Licht). Die Zapfen dagegen werden in drei Farbrichtungen unterteilt, Orangerot, Grün, Violettblau. Jeder Farbton kann nur einen begrenzten Bereich des Lichtes abdecken, aber zusammen decken die Zapfen ein breites Lichtspektrum ab. Beim Durchgang der Strahlen durch die Linse dreht sich das Bild. Die Sinneszellen zum Sehen wandeln das Bild in ein elektrochemisches Signal und senden dieses über den Sehnerv zum Gehirn, welches die Daten verarbeitet, dabei das Bild falls notwendig dreht und anschließend eine Sehempfindung generiert. [Küpp1992]

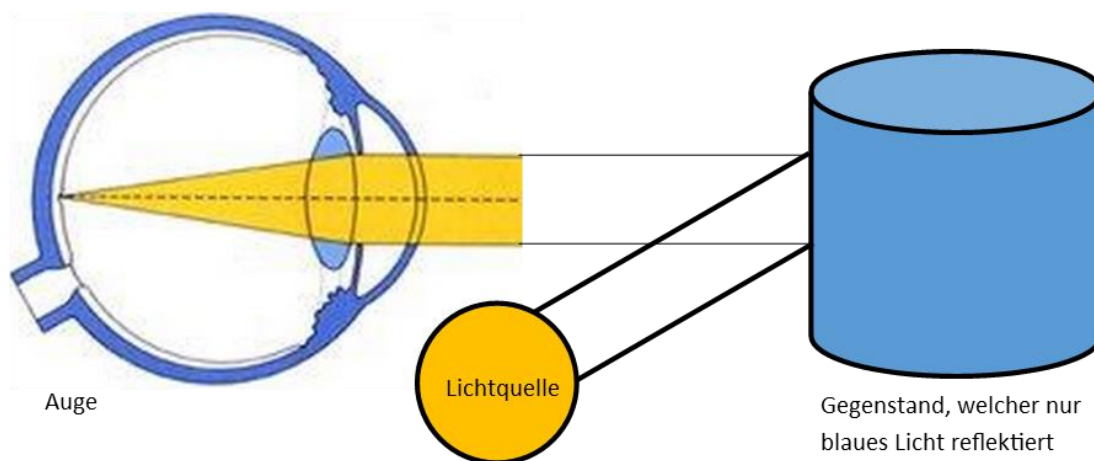


Abbildung 4-8

Die menschlichen Nachbauten, auch Kameras genannt, funktionieren nach eben diesem Prinzip. Das Gerät, welches die Aufgabe der Sinneszellen übernimmt wird Bildsensor genannt. Welche Vertreter es gibt wird im Kapitel 1.1.6 näher erläutert. Eine Vorverarbeitung kann sich anschließen, oder das Bildsignal wird direkt an eine Steuereinheit (SPS oder Computer) weitergeleitet. Dort wird sie verarbeitet und danach kann die Information gespeichert werden, oder an ein Ausgabegerät geschickt werden.

1.1.5 Parameter

In der Technik sind Parameter wichtige Werte, mit denen Größen gekennzeichnet werden. Parameter dienen der Beschreibung von Objekten und Anlagen. Sie sollten Information zu allen Schnittstellen und Aufgaben des Objektes (der Anlage) enthalten.

Das Wort **Format** wird für zwei verschiedene Situationen verwendet. In der ersten wird damit das Seitenverhältnis bezeichnet. Zum anderen beschreibt es die Abmessungen des Bildfeldes. Ein Beispiel ist das Format 1/2.5“ für ein Seitenverhältnis

von 4 zu 3 Einheiten, welches zum Beispiel für das Format 800 x 600 mm zutreffend wäre. [Gran1989]

Allgemein ist der **Kontrast** der Helligkeitsunterschied zwischen hellstem und dunkelstem Bereich. Dabei wird der Bereich je nach Anwendung unterschiedlich definiert. Wird die Kamera so angebracht, dass nur das Adressfeld im Sichtfeld der Kamera verbleibt, dann sind Inn- und Umfeld verschwommen, weswegen der Michelson-Kontrast zur Berechnung herangezogen werden muss.

$$K = \frac{(L_{max} - L_{min})}{(L_{max} + L_{min})} \quad L \dots \text{Leuchtdichte} \quad [\text{Spek2016}]$$

Die **Auflösung** ist ein Kennwert für die Bildpunktanzahl. Diese setzt sich aus der Auflösung der Höhe und der Breite zusammen. Des Weiteren gibt es eine Farbauflösung, da Computer nur ausgewählte Lichtwellenlängen produzieren können.

Die **Schärfe** ist ein Maß für die Detailgenauigkeit. Objekte können unterschieden werden durch Farbton, Helligkeit und Sättigung. Dabei erhöht eine große Auflösung den Kontrast an den Übergängen. Aber auch eine Verstärkung der Sättigung kann eine Erhöhung des Schärfeeindrucks hervorbringen. Je schärfer ein Bild ist, desto fehlerfreier ist die Informationsübertragung des Bildes. Als technisches Maß für die Bewertung der Schärfe wird die Anzahl unterscheidbarer Linien gezählt.

Adaptivität ist das englische Wort für Anpassungsfähigkeit. Für eine Kamera bedeutet dies zum Beispiel, wie gut diese sich an Lichtveränderungen anpassen kann. Die Adaptivität ist in diesem Fall somit ein Maß, wie schnell, weit und präzise eine Kamera die oben genannten Parameter anpassen kann.

„Die **Brennweite** rechnet man vom Hauptpunkt (optisches Zentrum) bis zum Brennpunkt“ [Gran1989] Brennpunkt ist der Punkt, an dem alle durch die Linse gehenden Strahlen, welche senkrecht zur optischen Achse der Linse verlaufen, sich treffen. Hauptpunkt ist der Punkt, um den man ein Objekt kippen kann, ohne dass sich der auf einer Projektionswand bildende Punkt sich bewegt. Nähert sich das Objekt, so muss das Objektiv ihm entgegen kommen, damit ein scharfes Bild entstehen kann.

Der **Bildwinkel** berechnet sich aus der Entfernung der Linse zum Bild und der maximalen Höhe / Breite / Diagonale des Bildes. Daher gibt es drei entscheidende Bildwinkel, den horizontalen, den vertikalen und den diagonalen. Umgedreht kann aus dem Bildwinkel und der Entfernung der Linse zum Bild die maximale Bildgröße ermittelt werden.

1.1.6 (Video-) Kamera

Eine Kamera ist ein Bildaufnahmegerät. Dabei werden von Objekten zurückgeworfene Lichtstrahlen von einem Bildsensor aufgenommen und in elektrische Signale gewandelt. Doch zuvor müssen die Strahlen durch ein Linsensystem, Objektiv genannt, damit die Parameter Schärfe und Bildgröße eingestellt werden können. Die

Objektive werden in drei Kategorien unterteilt. Die Unterteilung erfolgt aufgrund der Art des Sichtbereiches und der damit verbundenen Bildeigenschaften. Mit einem Teleobjektiv können weit entfernte Objekte vergrößert werden. Jedoch verbleibt nur eine geringe Schärfentiefe. Dies bedeutet, es kann nur ein bestimmtes Objekt gezielt scharf gestellt werden. Alles andere wird unscharf und damit zum Hintergrund. Normalobjektive sind für mittlere Entfernungen gut geeignet. Kennzeichnend für diese Objektivart ist die einfache Handhabung. Es sind keine Verzerrungen wie geringe Schärfentiefe oder Fischaugeneffekt zu erwarten. Des Weiteren lassen sich die meisten Normalobjektive scharfstellen. Dies bedeutet, die Position einer Linse kann verändert werden. Daher wird nicht die Entfernung zwischen Objekt und Kamera vom Objektiv sondern die Position der Linse von der Entfernung zw. Bildsensor und Objekt abhängig. Weitwinkelobjektive eignen sich besonders gut für Nahaufnahmen, denn sie besitzen einen großen Bildwinkel. Zu beachten ist, dass weiter entfernte Objekte stärker vergrößert werden als naheliegende. Dies kann durch eine Fischaugenlinse kompensiert werden, jedoch wird aus einem rechteckigen Sichtbereich ein ovaler.

Für digitale Kameras gibt es zwei wesentliche Arten von Bildsensoren. CMOS-Sensoren haben einen geringeren Stromverbrauch und eine geringere Baugröße als CCD-Sensoren. Ein weiterer Vorteil ist die hohe Bildrate. CCD-Sensoren haben einen größeren Füllfaktor als CMOS-Sensoren, weshalb Kameras mit CCD-Sensoren bei geringerem Lichteinfall immer noch gut arbeiten können, während CMOS-Kameras nur ein schwarzes Bild zeigen können. Durch die physische Kopplung einzelner Bildpunkte bei CCD-Sensoren kommt es zu einem gleichmäßigeren Farbverlauf als bei den CMOS-Sensoren, wo jeder Bildpunkt einzeln ausgelesen wird.

1.2 Schrift- / Texterkennung

Für die Texterkennung muss der nicht elektrische Text (z.B. Text auf Papier) in ein digitales Bild umgewandelt werden. Dabei muss zwischen Handschrift und Druckschrift unterschieden werden. Bei einem in Druckschrift geschriebenen Text kann jedes einzelne Zeichen von einem Wort getrennt werden, da sich keine Linien des einen Zeichens über ein Nachbarzeichen erstrecken. Bei Handschriften hingegen ist es möglich, dass ein Wort aus nur einer einzigen Linie besteht oder eine Linie mehrere Zeichen vereint. Daher gibt es zwei Richtungen in der Texterkennung, die eine wird OCR genannt und die andere HCR.

Die Wandlung in ein digitales Bild wird mit Hilfe eines Scanners, einer Kamera (mit Lichtquelle), o.ä. bewerkstelligt.

Gängige Bilddateiformate sind im Jahr 2016 JPG, JPEG, GIF, TIFF, PNG. PDF zählt nicht direkt zu Bilddateien, jedoch werden die Informationen als bildähnliche Datei gespeichert, um ein nachträgliches Ändern zu erschweren. Daher kann dieses Dateiformat ebenfalls als Ausgangsdatei für eine Texterkennung verwendet werden.

Die eigentliche Texterkennung erfolgt durch ein Computerprogramm, welches aus den Bilddaten (Bildpunktinformationen) Schriftzeichen oder Wörter extrahiert und diese anschließend in einem Schreibdokument zu einem Text zusammenfügt.

1.2.1 OCR (Optical Character Recognition)

[Ubun2015] erklärt seine Kunden OCR wie folgt:

„OCR is a technology that allows you to convert scanned images of text into plain text. This enables you to save space, edit the text and search/index it.“

Die deutsche Übersetzung lautet folgendermaßen: OCR ist eine Verfahrenstechnik, die es erlaubt eine eingescanntes Bild von einem Text in einen digitalen Text umzuwandeln. Dies eröffnet Ihnen die Möglichkeit, Platz zu sparen, den Text zu bearbeiten, zu durchsuchen und in ein Verzeichnis zu ordnen.

Wird OCR wort-wörtlich übersetzt, kommt man der Bedeutung schon sehr nahe. Es heißt „Optische Zeichenerkennung“ und beschreibt Methoden, wie Bilder in Zeichen umgewandelt werden können. Dies ist nur eine reine Erkennung möglicher Zeichen, dabei werden die Zusammenhänge zwischen den einzelnen Zeichen vollkommen ausgeblendet.

Die Arbeitsweise des OCR-Bausteins für den FineReader von Abbyy wird in den nachfolgenden Zeilen beschrieben:

„First, the program analyzes the structure of document image. It divides the page into elements such as blocks of texts, tables, images, etc. The lines are divided into words and then - into characters. Once the characters have been singled out, the program compares them with a set of pattern images.“ [Abbe2016]

In anderen Worten heißt es, das Bild wird in einzelne Teile zerlegt, um die Bereiche ohne Text nicht näher betrachten zu müssen. Dies spart Rechenkraft. Anschließend wird der Text in Zeilen, Wörter und letztlich in einzelne Zeichen zerlegt. Diese Zeichen werden mit vorher aufgenommen Bildern von Buchstaben und Zahlen in unterschiedlichen Schriftarten verglichen.

Die Universität Bamberg verallgemeinert die Definition in dem Sie das Dokument [Bamb2016] veröffentlicht und dort schreibt: „Unter Texterkennung oder OCR (Optical Character Recognition) versteht man die Umwandlung von Bildinformationen in Text.“ Allerdings verknüpft die Überschrift diese Aussage mit gescannten Dokumenten und schränkt die Informationsquelle somit wieder stark ein.

1.2.2 HCR (Handprint Character Recognition)

„Unter Handschrifterkennung sind alle Verfahren zu verstehen, die handgeschriebene Buchstaben, Ziffern, Wörter oder Sätze automatisch erkennen und in eine für den Computer zu verarbeitende Datei umformen.“ [ITWi2016-3]

Jeder Mensch hat seine ganz eigene Handschrift. Sie unterscheidet

1.2.3 Übergreifendes

„OCR-Programme verfügen über unterschiedliche Ausstattungsmerkmale. Besonders nützlich – und teuer – ist es, wenn sie gewissermaßen selbstlernend sind. Wenn sie also einen eingescannten Text lesen, digital umsetzen und uns dann zur Endkontrolle vorlegen, die sie wiederum dazu heranziehen, denselben Fehler nicht zweimal zu machen. Das klingt allerdings tatsächlich besser, als es zumeist funktioniert. Denn so richtig klappen kann das nur, wenn man immer wieder Vorlagen aus derselben Quelle hat.“ [Krat2014]

1.3 Programmiersprache

Die Auswahl der Programmiersprache ist wichtig, um schnell und effektiv programmieren zu können. In den letzten Jahren haben sich zwei wesentliche Hochsprachen herausgebildet. (Turbo) PASCAL verlangt eine klare Typzuweisung. Dies sorgt zwar für eine hohe Hackersicherheit, ist jedoch sehr steif. Daher wird PASCAL fast ausschließlich als Lernprogramm und für sicherheitsrelevante Programme verwendet. C ist die andere wesentliche Sprache. Sie bildet den Sockel, auf dem weitere Spezialisierungen wie C++, C# oder Java aufbauen. [DaBr2005] sagt über C: „C ist eine relativ ‚maschinennahe‘ Sprache.“, denn C arbeitet nur mit Zahlen und Adressen. Somit erlaubt diese Sprache eine hardwarenahe Programmierung. Allerdings enthält C auch Elemente der strukturierten Programmierung. Die Benutzung von impliziten Typumwandlungen erleichtert die Verwendung dieser Sprache.

Die Weiterentwicklung C++ wurde entwickelt, um objektorientierte Programmierung zu ermöglichen und ein strenges Typkonzept für selbst definierte Datentypen und Klassen zu realisieren. Die Autovervollständigen-Funktion kann nicht auf alle Elemente angewandt werden, was das Programmieren verlangsamt. Jedoch können auch Funktionen außerhalb einer Klassenfunktion gebildet werden, was in C# nicht geht.

C# ist ebenfalls eine Weiterentwicklung von C und stark auf Typ- und Objektorientierung ausgelegt. In C# können runtime-spezifische Objekte, wie der Garbage-Collector verwendet werden. Hier können Events verwendet werden, was in C++ nicht möglich ist. Da für den Trigger der Kamera solch ein Event unablässig ist, muss C# als Programmiersprache verwendet werden.

1.4 Der Hardwareschnittstellen (Stecker)

Die Literatur ist sich nicht einig, wie das Wort Hardware zu beschreiben ist. Daher sind hier drei verschiedene Definitionen zitiert.

„Hardware ist der Oberbegriff für alle mechanischen, elektrischen und elektronischen Komponenten, Baugruppen und Ausrüstungen eines Computers oder eines

Systems. DIN 44.300 definiert Hardware als "die Gesamtheit oder als Teil der apparativen Ausstattung von Computersystemen". " [ITWi2016-2]

„Gesamtheit der technischen Bestandteile eines Computers (oder anderer technischer Systeme), die nicht ausschließlich aus Daten bestehen, sondern physisch vorhanden sind“ [Wort2016]

„Gesamtheit der technisch-physikalischen Teile einer Datenverarbeitungsanlage “ [Dude2016-4]

Teil der Hardware sind die Hardwareschnittstellen. Wie im vorangegangenen Kapitel erklärt, ist eine Schnittstelle der Verbindungsbereich zwischen zwei „Dingen“. In diesem Fall werden die „Dinge“ Hardware genannt. Umgangssprachlich werden die Hardwareschnittstellen als Stecker, bzw. als Steckdose bezeichnet. Darüber können Daten von einer Hardware zu einer anderen übertragen werden. Dabei gibt es zwei grundlegende Arten der Übertragung. Werden die Daten parallel weitergereicht, so werden mehrere Leitungen benötigt. Werden die Daten seriell übermittelt, so wird nur eine Leitung benötigt, jedoch wird dies durch mehr Zeitaufwand erkauft.

2 Analyse der Aufgabenbedingungen

Das Texterkennungssystem wird Bestandteil eines derzeit in der Entwicklungsphase befindlichen Systems zur Identifikation falsch bestückter Briefe. Dazu werden die aus der Neopost kommenden Briefe gewogen. Danach wird entschieden, ob das Gewicht der Briefe im Rahmen liegt, oder ob diese zu schwer / leicht sind. Die Aufgabe, welche in dieser Bachelorarbeit gelöst werden soll, ist ein System, bestehend aus Hard- und Software, welches zu jedem Brief den entsprechenden Datensatz ermittelt.

Es sind zwei Maschinen vorhanden, an welche das Gesamtsystem gekoppelt werden soll. Die Aufgabe der ICS-Maschine ist es, Briefe zu drucken und mit den richtigen Karten zu verbinden. Die bedruckten Briefblätter werden an die Neopost⁴ übergeben. Diese faltet die Blätter und steckt sie anschließend in Briefumschläge.

Die Briefumschläge sind durch die DIN680 genormt. Die Neopost arbeitet nur mit den Größen C5 (162 x 229 mm), C6 (114 x 162 mm) und C5/6 (110 x 220 mm) mit Sichtfenster. Die durchsichtigen, 90 mal 50 mm großen Fenster können durch Spannungen in den Briefen Wölbungen aufweisen oder gar zerknittern. Dadurch wird das Licht unterschiedlich gebrochen. Dies führt zu Problemen beim Lesen der Adressdaten des Briefes durch das Sichtfenster. Eine weitere Schwierigkeit ergibt sich aus der Tatsache, dass die Dicke der Briefe zwischen 2 und 5 mm variiert.

Das Sichtfenster der Briefe kann aus verschiedenen Materialien hergestellt werden. Zellophan ist dabei der am häufigsten verwendete Werkstoff. Es ist ein aus Zellulose, Natronlauge und Schwefelkohlenstoff ([Wiki2015-2]) bestehender Werkstoff, welcher zu den Kunststoffen gehört. Es können jedoch auch Briefe mit Sichtfenstern aus anderen durchsichtigen, wasserbeständigen Materialien verwendet werden.

Die Lage des Papierblatts im Briefumschlag variiert in der Länge 19 mm und in der Breite 15 mm. Dies bedeutet, dass Adressfeld liegt nicht immer an derselben Stelle und kann ebenfalls ein wenig gedreht liegen. Die Schriftarten, welche für die Adressfelder verwendet werden, werden durch die Deutsche Post AG limitiert, denn die Briefe müssen auch von dieser Firma maschinell eingelesen werden können. Auch die Schriftgröße ist variabel und ist zwischen **6 und 14** anzusiedeln. Derzeit wird nur Schwarze Schrift gedruckt, jedoch gibt es die Möglichkeit, dass der Kunde bunte Adressen möchte, daher muss eine Texterkennung farbiger Schrift gewährleistet werden.

⁴ Neopost, Mailing-Maschine, faltet Briefblätter und verpackt diese

Die Neopost wirft bei voller Last ungefähr alle 1,7 Sekunden (mit Handy gestoppt) einen neuen Brief mit Kraft mit dem Sichtfeld in der linken oberen Ecke aus. Diese Briefe müssen auf einem Förderband ausgerichtet werden. Erst danach können die Briefe gewogen und die Adressfelder gelesen werden. Das Wiegen und die Texterkennung sollen gleichzeitig geschehen um Zeit und Platz zu sparen. Da ein Stau verhindert werden soll, muss das Förderband schneller sein, daher wird es so eingestellt, dass jede Sekunde genau ein Brief an der Kamera vorbei wandert.

Um die Waage zu schützen, wird das Gesamtsystem ringsherum von Platten geschlossen. Somit können kein Deckenlicht, oder andere störende Lichtquellen einfließen. Damit die Kamera etwas erkennen kann, benötigt diese mindestens eine Lichtquelle. Außerdem erzeugen die vorangehenden Maschinen (ICS und Neopost) kleine Erschütterungen, die vom Gesamtsystem kompensiert werden müssen (wegen Waage). Da es ein geschlossenes System wird, muss auch auf die Abwärme der eingebauten Komponenten und ihre Wärmeverträglichkeit geachtet werden. Ansonsten ist das Gesamtsystem nur normalen Innenraumbedingungen ausgesetzt (Raumtemperatur, normale Luftfeuchte, ...).

Die Druckdaten und die Informationen zum Bestücken mit Karten erhält die ICS-Maschine aus einer CSV-Datei. Daher muss die zu entwickelnde Maschine eine CSV-Datei erstellen können. Darin sollen alle Adressdaten und eine i.O/N.i.O-Zuweisung⁵ enthalten sein. Die Datei soll via Ethernet zur SPS⁶ der ICS-Maschine übertragen werden.

Die neue Maschine muss an den Auswurf der Neopost gekoppelt werden. An dieser Stelle ist in Auswurfrichtung nur 0,5 m Platz. Die Ausdehnung in die andere Richtung hingegen ist kaum begrenzt (über 1 m möglich). Die Höhe sollte gering gehalten werden, um ohne viel Materialaufwand eine große Stabilität zu gewährleisten.

Des Weiteren soll untersucht werden, ob im Haus vorhandene Hardware benutzt werden kann. Zur gegebenen Hardware gehört die Kamera DMK 23GP031 von IMAGINGSOURCE, die Kamera von, die SPS der ICS-Maschine als Steuereinheit, Ethernet-Kabel, Computermouse und -tastatur.

Zur Hardware gehört auch die nötige Software. Kostenfreie oder preiswerte Alternativen sind zu suchen und zu überprüfen.

⁵ (N.)i.O, Abkürzung für (nicht) in Ordnung

⁶ SPS, SpeicherProgrammierbare Steuerung, maschineller Computer

3 Vorhandene Geräte und Annahmen

Ein System für die Texterkennung besteht aus verschiedenen Komponenten. Die Kamera wandelt optische Signale in elektrische um. Die optischen Signale können durch unterschiedlichste Lichtquellen erzeugt und dann durch Filter verändert werden. Die von der Kamera kommenden Signale werden in der Steuerung (meist SPS, aber auch Computer) verarbeitet. Dies geschieht durch die installierte Software. Die Datenübertragung kann über verschiedenste Kabel oder drahtlos über Funk geschehen.

3.1 System 1: Keyence

Die Firma Keyence vertreibt neben Messsystemen u.a. Systeme für die Bild- und Texterkennung. Dazu bietet Keyence drei verschiedene Modellreihen an. Für hohe Leistungen und komplexe Systeme bieten sie die Modellreihe XG-8000/7000 an. Diese ist jedoch auch mit hohen Anschaffungskosten verbunden. Die preiswerteste Reihe ist die Modellreihe IV. Allerdings kann deren Komponenten nur zur Anwesenheitserkennung verwendet werden. Die Mitte bildet die Modellreihe CV-X200/X100 und wird als Bedienerfreundlich angesprochen. Daher fällt die Wahl auf eben jene Reihe. Zu dieser Reihe gehören 7 verschiedene Steuergerätetypen, aus denen man eine je nach den Bedürfnissen auswählen kann. Die Anzahl der unterschiedlichen Kameras, welche sich mit einer dieser Steuergeräte verbinden lassen ist atemberaubend, siehe 3.1.1.

3.1.1 Kameras

Keyence bietet 24 verschiedene Kameras. Die Auswahl reicht von Monochrom über Farbe bis zu Infrarot und Ultraviolett, von 0,31 MP⁷ bis 21 MP. Dabei sind die Megapixelangaben nur gerundete Werte und wurden aus der Auflösung der Breite und der Höhe berechnet. Hier ein Beispiel: Ein Kamera mit einer Auflösung von 2432 x 2040 Pixeln gilt als 5 Megapixelkamera. Aber auch eine Kamera mit 2432 x 2050 zählt in diese Kategorie. In der Übertragungszeit der Kameras zeichnen sich jedoch große Unterschiede ab. Die schnellsten Kameras brauchen nur weniger als 3 ms, um ein Bild zu übertragen, während die langsamste über 100 ms benötigt, da diese eine viel größere Datenmenge übertragen muss. Daher kann der schnellste Kamerateyp bis zu 333 Bilder pro Sekunde übertragen, während die langsame Kamera mit ihren weniger als 10 Bildern pro Sekunde nicht einmal an Filmqualität kommt (25fps). Bei einer Objektgröße von 90 mm und einer möglichen Strichbreite von nur 90 µm ist eine Auflösung von mehr als 1000 Pixeln in die Breite und 450 Pixeln in

⁷ MP, MegaPixel

die Länge von Nöten. Dies ergibt eine Größe von 1,45 MP. Deshalb muss eine Kamera aus dem Bereich der 2MP-Kameras oder höher ausgewählt werden. Da die Deutsche Post die Vorschrift hat, nur Briefe mit schwarz/weißem Adressfeld zu bearbeiten, ist eine Schwarz/Weiß-Kamera die richtige Wahl. Bei der Verwendung einer Farbkamera könnten die durch Lichtbrechung entstehenden Farbnuancen zu Problemen bei der Texterkennung führen. Daher ist Monochrom zu bevorzugen. Die Übertragungszeit muss so klein wie möglich gewählt werden, um Zeit zu sparen, die für die Textverarbeitung und den Stringvergleich benötigt wird. Die 21MP-Kamera hat eine Übertragungszeit von ca. 110 ms und die anschließende Textverarbeitung wird durch die große Datenmenge ebenfalls mehr Zeit beanspruchen, daher ist diese Kamera für eine Echtzeit-Texterkennung von Adressfeldern ungeeignet. Eine schnelle 2MP-Kamera ist daher die optimale Wahl. Um die Lichteffekte durch Wölbungen und Knicke in der Folie des Sichtfensters eines Briefes aus dem Bild zu entfernen, welches an die Texterkennung übergeben werden soll, soll die Software LumiTrax verwendet werden. Daher muss die Kamera LumiTrax-kompatibel sein. Dies trifft nur auf eine 2MP-Kamera zu, die CA-HX200M.

Für jede Kamera wird ein Objektiv benötigt. Auch hier ist die Auswahl groß. 9 Modellreihen werden angeboten. Bei der Auswahl muss auf die Befestigung geachtet werden, denn nicht jedes Objektiv kann auf jeder Kamera befestigt werden. Um Verzerrungseffekten vorzubeugen sollte ein Normalobjektiv verwendet werden.

3.1.2 Beleuchtung

Keyence bietet eine große Auswahl an Leuchtquellen und Filtern. Die Stableuchten beleuchten ein Objekt auf der ganzen Länge gleichmäßig. Polarisationsfiltern reduzieren Glanzlichter, können jedoch den Schattenwurf durch starke Falten und Knicke in einer Folie nicht verhindern. Die Bildung von Lichtstreifen durch Knicke kann ebenfalls nicht gänzlich unterbunden werden. Kommt ein Brief mit zerknittertem Sichtfenster aus der Neopost, so erzeugt diese Lichtquelle Schatten und Lichtstreifen, die eine korrekte Texterkennung unmöglich machen. Ringleuchten verbreitern die negativen Effekte der Stableuchte und ist somit ebenfalls ungeeignet. Diffus-Leuchten sind besonders für stark spiegelnde Prüfobjekte prädestiniert. Allerdings beheben diese das Problem der fehlerhaften Lichtbrechung durch den Knick nicht, können also die Abbildung des Knicks auf dem Kamerabild nicht verhindern. Hintergrundleuchten ermöglichen nur die Sicht von Objektaußenkanten. Die Buchstaben stehen aber mitten auf dem Blatt, welches sich im Briefumschlag befindet. Daher sind Hintergrundleuchten für diese Aufgabe ungeeignet. Die Kuppelleuchten, auch Domleuchten genannt, erzeugen ein gleichmäßiges Licht von vielen Seiten, wodurch die Lichtstreifeneffekte verringern. Aber die Schatteneffekte bleiben dennoch und können die Texterkennung erschweren. Keyence bietet zusätzlich noch die LumiTrax-Leuchten-Reihe an. Diese bestehen aus acht einzeln schaltbaren LED-Reihen. Mit der zugehörigen Software für die Steuereinheit können mehrere Bilder mit unterschiedlicher Beleuchtung erstellt werden und somit die Lichtstreifen- und Schatteneffekte herausgerechnet werden.

3.1.3 Filter

Für jede Leuchtquelle und jedes Objektiv kann ein passender Diffusionsfilter, bzw. ein passender Polarisationsfilter geordert werden.

Eine Diffusionsplatte verhindert die Spiegelung einzelner LEDs. Das durchfallende Licht wird von einer solchen Platte gestreut. Durch diese Verteilung kann ein großes Objekt mit einer kleinen Lichtquelle gleichmäßig ausgeleuchtet werden.

Die Polarisationsplatte verringert Blendlichterscheinungen, indem sie die Lichtwellen nach Ausbreitungsrichtung filtert. Es werden nur parallele Strahlen einer bestimmten Richtung durch den Filter gelassen. Dadurch wird das Licht gezielter auf das Objekt gelenkt und der Lichtanteil, welcher auf Grund von Wölbungen in eine andere Richtung reflektiert wird, wird herausgefiltert.

Der Mitarbeiter von Keyence empfahl einen Polarisationsfilter vor der Lichtquelle und einen vor der Kamera. Dies demonstrierte er an einem Lifebeispiel. Negative Effekte, welche durch Wölbungen hervorgerufen wurden konnten fast vollständig beseitigt werden. Schädliche Effekte durch Knicke hingegen waren kaum gemildert. Experimente, welche mit der Kamera DMK23GP031 von TheImageSource durchgeführt wurden, zeigten, dass diffuses, gleichmäßiges Licht keine schädlichen Effekte entstehen lässt. Jedoch sinkt die Lichtdichte im Zielbereich und somit wird eine stärkere Lichtquelle benötigt.

3.1.4 Steuereinheiten, Kabel, Stecker, Befestigung

Die Steuereinheit muss die Kamera CA-HX200M unterstützen. Des Weiteren muss sie das Funktionieren der Software für LumiTrax sicherstellen. Die Bildspeicherung und die Textverarbeitung müssen in weniger als zwei Sekunden erfolgen. Die Wartung und Bedienung sollte so einfach wie möglich sein, daher ist die Modellreihe CV-X200/X100 zu empfehlen. Die Verbindung zw. Kamera und Steuereinheit bildet das mitgelieferte Spezialkabel. Es kann mit geraden oder mit abgewinkelten Steckern geliefert werden. Alle Komponenten besitzen Löcher mit Gewinde. Ein Gerüst muss vom Kunden selbst nach seinen Wünschen selbst gebaut werden.

3.1.5 Software

Für Kamerasysteme von Keyence werden Softwarepakete angeboten. Die Softwarekomponente LumiTrax hilft bei der Bilderfassung. Dabei werden vier Bilder mit unterschiedlicher Beleuchtung erzeugt, wofür eine LumiTrax-kompatible Kamera und eine LumiTrax-kompatible Lichtquelle benötigt werden. Anschließend vergleicht dieses Programm die Bilder, um die negativen Effekte zu isolieren. Somit kann der Glanz durch das Sichtfenster und etwaige Wölbungs- und Knittereffekte eliminiert werden. Sich verändernde Beleuchtungsverhältnisse durch Umgebungslicht stellen keine Probleme mehr dar. Laut Keyence ist auch die Änderung der Messobjektausrichtung durch Transportbedingungen tolerabel. LumiTrax kann ebenfalls verwendet

werden, um Glanzeffekte durch Knitter und Wölbungen zu erkennen. Das Programm ist für eine einfache Bedienung programmiert worden. So kann die Software so eingestellt werden, dass Gravuren sichtbar werden, oder so, dass das gesamte Objekt hinter einer Folie qualitativ hochwertig zu erkennen ist. Der Baustein Kratzer zum Beispiel hilft bei der Erkennung solcher. Die Auto-Teach-Funktion ermöglicht das leichte Anlernen neuer Bildsätze. So können auch bisher nicht verwendete Schriftarten und Schriftgrößen dem Texterkennungssystem schnell gelernt werden. Es muss nur eine kleine Anzahl fehlerfreier Testobjekte von der Software eingelesen werden, dann kann die Prüfvorrichtung verwendet werden. Dies ermöglicht eine hohe Flexibilität. Der Baustein Bildaufbereitungsfilter enthält 24 verschiedene Typen. Um einige zu nennen, Extraktion Linienfehler, RT-Kontrastkorrektur, RauschpegelEinstellung und Kontrasterweiterung sind einige Vertreter. Diese können jederzeit dazu geordert und integriert werden. Mit ShapeTrax2 kann ein Referenzsymbol auf einem Messobjekt gesucht werden. Dies gewährleistet eine genaue Positions- und Dimensionsmessung. Des Weiteren gibt es eine große Auswahl an Geometrie-Werkzeugen. Dabei wird versprochen: „Hochpräzise Prüfungen von Größen und Abständen (Dimensionsprüfungen) lassen sich mit einfachen Mausoperationen intuitiv durchführen.“ Für eine Adressfeldererkennung auf einem Brief ist die Funktion Rechteckmittelpunkt für die Bildbereichszentrierung hilfreich. Der Softwarebaustein OCR befasst sich mit der Texterkennung. Dieses Programm gibt nicht nur die erkannten Schriftzeichen wieder, sondern liefert gleichzeitig Informationen zu Extraktionsbedingungen und Stabilität der Zeichenerkennung. OCR vergleicht ausgewählte Ausschnitte eines Bildes eines Messobjektes mit einer Zeichenbibliothek. Diese Bibliothek kann mit Auto-Teach gefüllt werden. Durch Verzicht auf eine Binärverarbeitung ist die Software sehr tolerant gegenüber Helligkeitsveränderungen. Des Weiteren können auch verschlüsselte Datumsaufdrucke unterstützt werden, indem die beurteilten Zeichen mit einer Verschlüsselungstabelle umgewandelt werden. Für die Identifizierung von Briefen mit Strich-, bzw. Pixel- oder sogar Composite-Code kann der 1D-/2D-Codeleser eingesetzt werden. Diese Software kann mit der gleichen Kamera arbeiten, welche für die Texterkennung verwendet wird. Somit können Kosten gespart werden. Noch mehr Ersparnis birgt die Tatsache, dass das Codelesen und das Prüfen simultan geschehen können. Es werden eindimensionale Codes des Standards ISO/IEC 15416, zweidimensionale Codes der Standards ISO/IEC 15415, AIM DPM-1-2006, SAE AS9132 und die Composite Codes CC-A, CC-B und CC-C unterstützt. Die Druckqualität der gelesenen Codes kann gleichzeitig von demselben Programm überprüft werden. Keyence bietet noch eine große Menge weiterer Bausteine an, welche für dieses Projekt irrelevant sind und daher hier keine Erwähnung finden werden. Diese können in Prospekten der Firma Keyence oder auf deren Website „www.keyence.de“ nachgelesen werden.

Zusammengeführt werden diese Bausteine auf einer gemeinsamen Bedienoberfläche. Dazu kann aus einem Katalog eine passende Bildschirmaufteilung ausgewählt werden.

3.2 System 2: TheImageSource, Tesseract

Dieses System besteht aus der Kamera DMK23GP031, hergestellt von der Firma TheImageSource. Die Kamera hat einen LAN-Anschluss. Des Weiteren gibt es noch einen sechspoligen Anschluss für die Energieversorgung und einen externen Trigger. Der Trigger löst dann die Kamera an, welche dann ein Bild erstellt, welches über Ethernet abgerufen werden kann. Das Bild wird mit einer OpenSource-Software bearbeitet. Der Datenstring aus der Texterkennung und die aus der CSV-Datei der ICS-Maschine werden auf eine gemeinsame Datenstruktur gebracht und verglichen. Gibt es eine Übereinstimmung wird die Entscheidung, ob der Brief in Ordnung ist, der Waage überlassen. Gab es keine Übereinstimmung, wird sofort ein Signal an die Waage geschickt, damit diese den Brief auf den n.i.O.-Stapel wirft.

3.2.1 Vorabtests

Diese Versuche waren notwendig, um zu ermitteln, ob die Kamera den Anforderungen genügt. Ein wichtiger Parameter ist die Auflösung. Die Schriftgröße der Adresse auf einem Brief kann stark variieren. Zwischen der Schriftgröße 1 und 20 ist alles denkbar. Damit die kleinste Schrift, deren Striche ca. 0,36 mm breit sind, ebenfalls gut erkannt wird, muss die Auflösung bei einer Gesamtbreite von 90 mm mindestens 250 Pixel betragen. Um mögliche Serifen und Linien von Tabellen und Barcodes, bzw. Punkte von QR-Codes korrekt auslesen zu können sollte die Auflösung jedoch über 1000 Pixel betragen (0,09 mm Punktbreite). Die Maximalauflösung der DMK23PG031 wurde vom Hersteller mit 2592 x 1944 Pixel angegeben.

Um Probleme der Nahfeldeinflüsse von Linsensystemen zu umgehen sollte im Fernfeld gearbeitet werden. [Gran1989] hat in seinem Buch eine Formel beschrieben, mit der die notwendige Entfernung eines Objektes berechnet werden kann. In diese Formel fließen die Objektgröße, die Bildgröße und die Entfernung des Brennpunktes der Linse zur Bildfläche ein. Die Objektgröße ist die Breite und Länge des Sichtfeldes, welche 90 x 40 mm beträgt. Die Bildfläche wird von den Dimensionen des Sensors der Kamera begrenzt, welcher rund 5,7 x 4,2 mm groß ist. Die Lage der Brief unter der Kamera ist Toleranzen von mehreren Millimetern unterworfen und daher muss die Objektgröße etwas erweitert werden. Nach mehreren Rechnungen ergab sich eine zu überwachende Objektgröße von 114 x 84 mm. Der Brennpunkt der Linse ist 15 mm vom Sensor entfernt. So ergeben sich folgende Formeln:

$$En = \frac{O * f}{b} = \frac{114mm * 15 mm}{5,7mm} = 300,0mm = 30cm$$

$$En = \frac{O * f}{b} = \frac{84mm * 15 mm}{4,2mm} = 300,0mm = 30cm$$

Aus diesen Rechnungen ergibt sich ein Abstand von 30 cm zwischen dem Sichtfenster eines Briefes und der Kamera.

Ein weiterer bedeutender Parameter ist die Farbe. Die Kamera DMK23GP031 besitzt nur einen Schwarz/Weiß-Sensor. Eine Unterscheidung zwischen zwei bunten

Farben ist mit dieser Kamera nicht möglich, da jedoch nur Text von einem weißen Hintergrund unterschieden werden soll, kann diese Kamera dennoch problemlos eingesetzt werden.

Nachdem das Bild scharfgestellt ist, muss der Kontrast optimiert werden. Denn je höher der Kontrast ist, desto besser kann das Texterkennungsprogramm arbeiten. Mit verschiedenen Lichtquellen und Veränderung verschiedener Parameter der Kamera kann dies realisiert werden. Als Lichtquelle wurden Tageslicht zu verschiedenen Tageszeiten, eine punktförmig abstrahlende LED-Lampe mit mehreren LEDs und eine Lampe mit acht in einer Geraden angeordneten LEDs verwendet.

Die mit Tageslicht aufgenommenen Bilder wiesen den höchsten Kontrast auf, ohne dass die Knicke und Wölbungen des Sichtfensters des Briefes Auswirkungen auf das Bild hatten. Mit der Stabbeleuchte konnten annähernd gute Resultate beobachtet werden, wenn man mit dieser die eine Seite der Verkleidung beleuchtet, welche die gesamte Anordnung umgibt. Wird das Sichtfenster hingegen direkt angeleuchtet, so kommt es zu mannigfaltigen Reflektionen. Die Knicke und Wölbungen des Sichtfensters werden dadurch stark sichtbar und stören die Texterkennung. Die punktförmig abstrahlende Lichtquelle hingegen konnte nur einen kleinen Bereich ausleuchten und eignete sich gar nicht für die indirekte Beleuchtung.

Die im Tageslicht aufgenommenen Bilder haben die beste Qualität, jedoch mussten die Parameter der Kamera in kurzen Abständen von einem Menschen nachjustiert werden. Ein Computerprogramm zu schreiben, welches diese Parameter selbstständig regelmäßig anpasst, würde den Rahmen einer Bachelorarbeit sprengen und entfällt deshalb. Außerdem würde solch ein Programm viel Computerleistung benötigen und die Gefahr bergen, dass andere Programme nicht schnell genug bearbeitet werden können (z.B. die Texterkennung). Um die zweitbeste Variante, die Stabbeleuchten, einzusetzen muss die Anordnung aus Kamera und Brief (und Förderband) optisch abgegrenzt werden. Die so entstehende Kiste sollte aus lichtundurchlässigen, stabilen Platten gebaut werden, sodass sie die Anordnung auch vor Staub und anderen Umwelteinflüssen schützt. Die Innenseiten der Kiste sollen das Licht der Leuchtquelle im gesamten Innenraum verteilen. Dazu ist es wichtig das Licht zu reflektieren und zu zerstreuen. Wie in Kapitel 1.1 ersichtlich reflektieren helle Farben mehr Licht als dunkle. Weiß ist die hellste Farbe und ist somit die Farbe der Wahl. Die Streuung kann durch eine feinkörnige Rauheit erzeugt werden. Mit dieser Anordnung kann der Brief von allen Seiten gleichmäßig beleuchtet werden und es können keine Schatten hinter Wölbungen und Knicken des Sichtfensters entstehen. Damit das Licht sich extrem diffus ausbreiten kann, wird eine Streuscheibe aus milchigem Acrylglas mit einer Lichtdurchlässigkeit von 78% vor der Lichtquelle, welche eine LED-Matrix ist, aufgebaut. Die Stärke der Leuchtquelle muss so angepasst werden, dass es keine Reflektionen mehr gibt. Gleichzeitig müssen die Parameter der Kamera immer wieder nachjustiert werden, bis ein Bild ohne Schatten, ohne Lichtreflektionen und mit maximal möglichem Kontrast entsteht.

Diese Bilder müssen in einen Text verwandelt werden, damit der Inhalt mit der CSV-Datei verglichen werden kann. Im Internet findet man eine Vielzahl an OCR-Software. Die meisten sind kostenpflichtig und auf einige Anwendungsfälle spezialisiert (z.B. Erstellung veränderbare PDF-Dateien). Das kostenlose OpenCV hingegen war in der Installation zu aufwendig, da noch viele weitere Programme zusätzlich installiert werden musste. Daher fiel die Entscheidung auf Tesseract. Dies ist eine leicht zugängliche, kostenfreie Software, welche sich mit wenigen Klicks downloaden lässt und ständig aktualisiert wird. Ein weiterer Pluspunkt ist, dass Tesseract 2016 als bestes OpenSource-OCR-Programm gilt. Die neueste Version kann Zeilen und Spalten trennen, dies bedeutet, es können auch Tabellen eingelesen werden. Bei der Installation des Programmes können verschiedene Sprachpakete ausgewählt werden. Des Weiteren ist zu nennen, dass Tesseract auch als API erhältlich ist und sich damit leichter in ein selbst geschriebenes Programm einarbeiten lässt. Als Nachteil sei zu nennen, dass Tesseract nicht selbstständig eine Sprache anhand von Beispielbildern lernen kann, sondern auf ein fertiges Sprachpaket angewiesen ist. Dennoch ist Tesseract der Wahlsieger.

Wenn ein Sichtfenster eines Briefes im Sichtbereich der Kamera liegt wird durch eine Lichtschranke ein Triggerereignis ausgelöst. Erkennt die Kamera solch ein Triggerereignis, macht diese ein Bild. Ein Programm muss dieses Bild abholen, in einen vereinbarten Ordner abspeichern und Tesseract starten, damit Tesseract sich dieses Bild holen kann und einen Text daraus erstellt. Diesen Text speichert Tesseract ebenfalls in einen vereinbarten Ordner, auf den ein weiteres Programm zugreift, um den Text in die richtige Reihenfolge zu bringen und mit den Daten aus der CSV-Datei der ICS-Maschine zu vergleichen. Nach dem Vergleich trifft dieses Programm eine Entscheidung und kommuniziert danach entsprechend mit der Waage.

4 Erkundungsaufbau

4.1 Ziel

Es ist wichtig sich mit den Eigenheiten eines neuen Systems vertraut zu machen. Die Features der Kamera müssen evaluiert werden, um die Eignung der Kamera für die Texterkennung zu überprüfen und zu bestätigen. Außerdem dient dieser Versuch der Analyse des mitgelieferten Computerprogramms. Diese Analyse wird benötigt, um später ein eigenes Programm schreiben zu können.

4.2 Material

Für das Gestell der Kamera werden Profile verwendet. Diese werden mit den dazugehörigen Schrauben fest verbunden, sodass sie sich nicht bewegen können. Des Weiteren wird eine Kamerabefestigung benötigt, welche an ein Profil anschraubbar ist. Diese wird eine Metallplatte mit Löcher passender Größe an den benötigten Stellen. Die Kabel werden mit Klebeband an den Profilen befestigt, damit diese nicht in das Bild rutschen. Für die Vereinfachung der Positionierung eines Briefes werden noch einige weiße Blätter Papier benötigt. Verwendung fanden A3-Blätter aus Restbeständen. Da diese jedoch zu klein sind, bedarf es zusätzlich durchsichtigen Klebebandes.

Nachdem die Umgebung geschaffen wurde, bedarf es des Testobjektes, ein C5/6-Briefumschlag. Als Kamera wird die DMK23GP031 von TheImageSource verwendet. Über ein Ethernet(LAN)-Kabel wird sie mit einem Laptop verbunden, auf dem die Software „IC Capture 2.4“ und „Tesseract“ aufgespielt sind.

4.3 Aufbau

Für gute Ergebnisse ist es unabdingbar, den Abstand zw. Kamera und Objekt gleich groß zu halten. Dafür wird ein Gestell benötigt. Die Entfernung der Kamera zum Boden berechnet sich aus der Verstärkung der Linse, dem Abstand zw. Objekt und Linse und der Entfernung des Bildsensors von der Linse und beträgt 30 cm. Nachdem die Kamera sicher befestigt ist, muss diese mit einem Computer verbunden werden. Dies geschieht mit einem Ethernet(LAN)-Kabel und einer IP-Konfiguration. Anschließend wird die Kamera mit dem Programm „IC Capture 2.4“ in Betrieb genommen. Der erste Parameter, welcher angepasst wird, ist die Auflösung. Damit das Versuchsobjekt, eine C5/6-Briefumschlag, an der richtigen Stelle liegt, wurde das Gestell der Kamera auf ein weißes Papier gestellt, auf dem die Umrisse eines Briefes an der ausgemessenen Stelle aufgezeichnet sind. Ist die perfekte Auflösung gefunden, wird das Programm nach weiteren Einstellungsmöglichkeiten durchsucht. Die ersten Parametereinstellungen finden bei Tageslicht statt, da um das Gestell keine Abdeckungen vorhanden sind. Bei Bedarf sollen die Versuche in der Dämmerung und in der Nacht wiederholt werden.

4.4 Durchführung

Nach der Berechnung des Abstandes wurden Profile von Item aus Restbeständen organisiert und so zusammen geschraubt, dass eine stabile Befestigung entsteht, welche in Längsrichtung zum Brief Öffnungen in Bodenhöhe aufweist. Anschließend wurde die Kamera daran befestigt. Aus zwei A3-Blättern entstand durch zusammenkleben mit durchsichtigem Klebeband ein Blatt Papier, auf welchem das gesamte Gestell Platz fand. Deswegen Position wurde umgehend festgehalten und die Lage des Briefes berechnet und ebenfalls aufgezeichnet. Vorausschauend wurde das Papier auf minimale Größe geschnitten. Nun wurde die Kamera angeschlossen, deren IP-Konfiguration programmiert, ein Brief auf die vorgezeichnete Position gelegt und das Programm „IC Capture 24“ gestartet. Zuerst sollte laut Aufbau die Auflösung angepasst werden. Laut Datenblatt der Kamera und anschließender Berechnung sollte es die Auflösung 1920 x 1080 Pixel sein. Im Praxistest bestätigte sich dies. Schnell fanden sich weitere einstellbare Parameter. Das Programm ermöglicht die Beeinflussung der Helligkeit durch Veränderung der Beleuchtungsdauer. Schärfe, Kontrast, Rauschen und weitere Parameter sind ebenfalls anpassbar. Anschließend wurde mit den besten Einstellungen ein Bild gemacht und mit der Software Tesseract in eine Textdatei umgewandelt.

Damit die korrekte Funktionsweise der Kamera auch bei schwächerem Licht sichergestellt werden kann, wurde der Test während der Dämmerung und in der Nacht wiederholt.

4.5 Ergebnisse

Die Einstellung der Parameter ist aufwendig, da diese sich gegenseitig beeinflussen und kaum Spielraum bieten. Jedoch konnte bei hellem Tageslicht ein Bild erzeugt werden, welches keine Grautöne, nur Schwarz und Weiß enthielt. Selbst die 6-Pixel große Schrift wurde fehlerfrei angezeigt und in Text verwandelt. Durch anpassen der Parameter kann auch bei schwachem Licht gearbeitet werden.

5 Experiment 1 „Licht und Schatten“:

5.1 Ziel

Durch Transportprozesse werden die Folien der Sichtfenster von Briefumschlägen zerknittert und verbogen. Fällt Licht durch so veränderte Folien, wird das Licht unkontrolliert gestreut. Dadurch entstehen Schatten und Überbeleuchtungen. Für eine sichere Texterkennung ist ein Bild ohne Lichtirritationen von Nöten. Die Licht- und Schattenstreifen müssen reduziert werden. Dieses Experiment analysiert die Ergebnisse durch unterschiedliche Lichtquellen. Damit soll eine Lichtquelle gefunden werden, welche die beste Bildqualität für einen lang andauernden Prozess ermöglichen kann.

5.2 Material / Gegenstände / Software

Als Testobjekt für die Ermittlung der besten Beleuchtung wurden sechs verschiedene Briefumschläge mit Sichtfenster verwendet. Dadurch wird sichergestellt, dass Briefe der Größe C4 genauso gelesen werden können, wie C5- oder C5/6-Briefumschläge. Die eine Hälfte der Briefumschläge ist glatt, während die andere Hälfte zerknittert wird. Das einzuschiebende A4-große Briefpapier kann einmal zu A5 gefaltet und zweimal zu 1/3 A4 gefaltet werden. Auf das Briefpapier wurden Adressfelder in verschiedenen Schriftgrößen aufgedruckt. Somit kann überprüft werden bis zu welcher Schriftgröße das Adressfeld mit dieser Beleuchtung korrekt ausgelesen werden kann. Mit dem ersten Adressfeld (Schriftgröße 12) wird überprüft, ob auch große schwarze Flächen zusammenhängend von der Kamera erkannt werden kann. Gleichzeitig muss sichergestellt werden, dass auch Adressfelder mit kleiner, dünner oder kursiver Schrift nicht durch Licht- bzw. Schatteneffekte unleserlich werden. Außerdem können Briefumschläge und Briefpapier für ein weiteres Experiment wiederverwendet werden.

Ohne Kamera können keine Testbilder festgehalten werden, welche zum Schluss hinsichtlich der Qualität verglichen werden können. Da die Kamera den derzeitigen Zustand der beleuchteten Objekte vergleichbar abbilden soll, ist es von oberster Wichtigkeit, dass immer ein- und dieselbe Kamera verwendet wird. Die Kamera DMK23GP031 von TheImaSource ist kompakt, durch das mitgelieferte Computerprogramm, bzw. den mitgelieferten Quellcode, mit dem ein eigenes, anwendungsspezifisches Programm erstellt werden kann, leicht einstellbar und über Ethernet (LAN) von fast jedem handelsüblichen Computer ansprechbar.

Damit die Kamera gute Bilder aufnehmen kann, muss der Abstand zw. Objekt, Linse und Bildsensor stimmen. Damit der Abstand bei allen Aufnahmen gleich bleibt, wurde die Kamera an einem Gestell gefestigt, dessen Aufbau im Kapitel 4 mit beschrieben wurde.

Die Auswahl der richtigen Vertreter für die Beleuchtung hingegen ist keine triviale Angelegenheit. Da Tageslicht eine natürliche Lichtquelle ist, welche keine Energieversorgung benötigt und kostenlos ist, bildet dies die erste Quelle für diesen Test. Diese Quelle für diffuses Licht ist jedoch tageszeit- und wetterabhängig. Um von sich ständig ändernden Umwelteinflüssen unabhängig zu sein, werden ebenfalls verschiedene künstliche Lichtquellen getestet. Damit einfallendes Tageslicht die Ergebnisse nicht verfälschen kann, muss die Testapparatur mit undurchsichtigen Materialien, wie zum Beispiel Pappe, abgeschirmt werden. Die Auswahl an künstlichen Lichtquellen ist enorm, z.B. LED-Lampen, Glühbirnen, Halogenlampen, uvm. Des Weiteren kann die Leuchte aus einer oder mehreren Lichtquellen bestehen. Mit einer Streuscheibe aus milchigem Acrylglas kann das Licht noch gleichmäßiger und diffuser verteilt werden. Mit einem Polarisationsfilter hingegen kann das aus der Quelle strömende Licht parallelisiert werden.

Da es schwierig und teuer ist eine Halogenleuchte zu organisieren, welche kleiner als 30 cm ist und da Halogenlampen sehr heiß werden, können diese sowieso nicht eingesetzt werden. Für die LED-Lampen werden drei Vertreter eingesetzt. Die Gemeinsamkeit der LEDs ist die Lichtfarbe, welche als Weiß bezeichnet wird. Durch die Bauform ist der Abstrahlwinkel der LEDs sehr gering, wodurch die Neigung zw. benachbarten Lichtstrahlen als vernachlässigbar gegenüber der Gesamtentfernung einzustufen ist. Als Folge dessen können die Strahlen als parallel angesehen werden. Die erste gewählte LED-Leuchte erzeugt einen kleinen, konzentrierten Lichtpunkt. Vertreter Nummer Zwei besteht aus einer geraden Linie aus acht LEDs. Nummer drei ist eine Matrix aus 36 einzelnen LEDs pro 140 x 140 mm großer Platine. Insgesamt bilden zwei solcher Platinen die gesamte dritte Beleuchtungsvariante. Jede Platine erzeugt ein Licht der Stärke 756 lm. Dies ergibt eine Gesamtstrahlkraft von 1512 lm. Dabei benötigen die LEDs zusammen nur 18 W.

In alten Beständen Glühbirnen in drei verschiedenen Farben (Weiß, Gelb, Grün) gefunden werden. Die 3V-Glühbirnen erzeugen ein gleichmäßiges Licht, welches gedacht war, um Puppenhäuser zu beleuchten. Jedoch produzieren diese Lampen eine beachtliche Abwärme

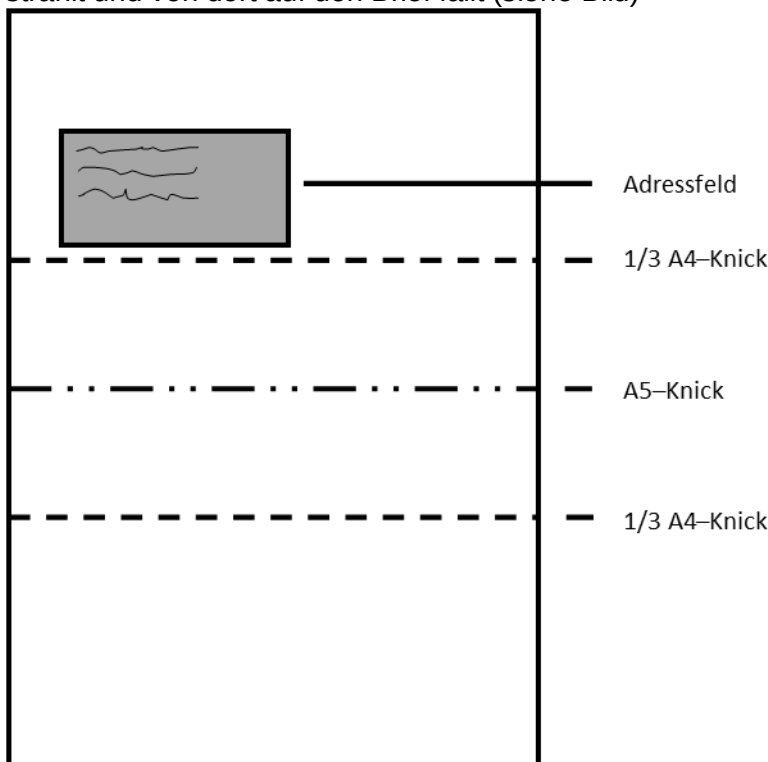
Die erste LED-Variante (1 Lichtpunkt) ist ein Beispiel für paralleles, konzentriertes Licht, da für das Experiment kein Polarisationsfilter zur Verfügung stand. Die LED-Matrix erzeugt ein starkes, großflächiges, jedoch relativ paralleles Lichtfeld. Die Glühbirnen hingegen sind für ihr diffuses Licht bekannt. Ihr Nachteil ist die Wärmeentwicklung. Des Weiteren sind kaum noch Glühbirnen im Handel erhältlich. Energiesparlampen gibt es nicht in so kleiner Form, dass diese in die Testbox passen. Des Weiteren sind mit der Arbeit mit Netzspannung verschiedene Anforderungen zu beachten. So dürfen keine Leiter direkt neben leicht entzündlichen Materialien, wie zum Beispiel Pappe, verlegt werden. Die Kosten für Abschirmung, Belüftung, bzw. Kühlung sind für dieses Experiment nicht angemessen. Daher wird an dieser Stelle auf den Einsatz von Energiesparlampen verzichtet. Ob Polarisations- oder Streufilter eingesetzt werden sollten, wird nach Auswertung dieses Experimentes entschieden werden.

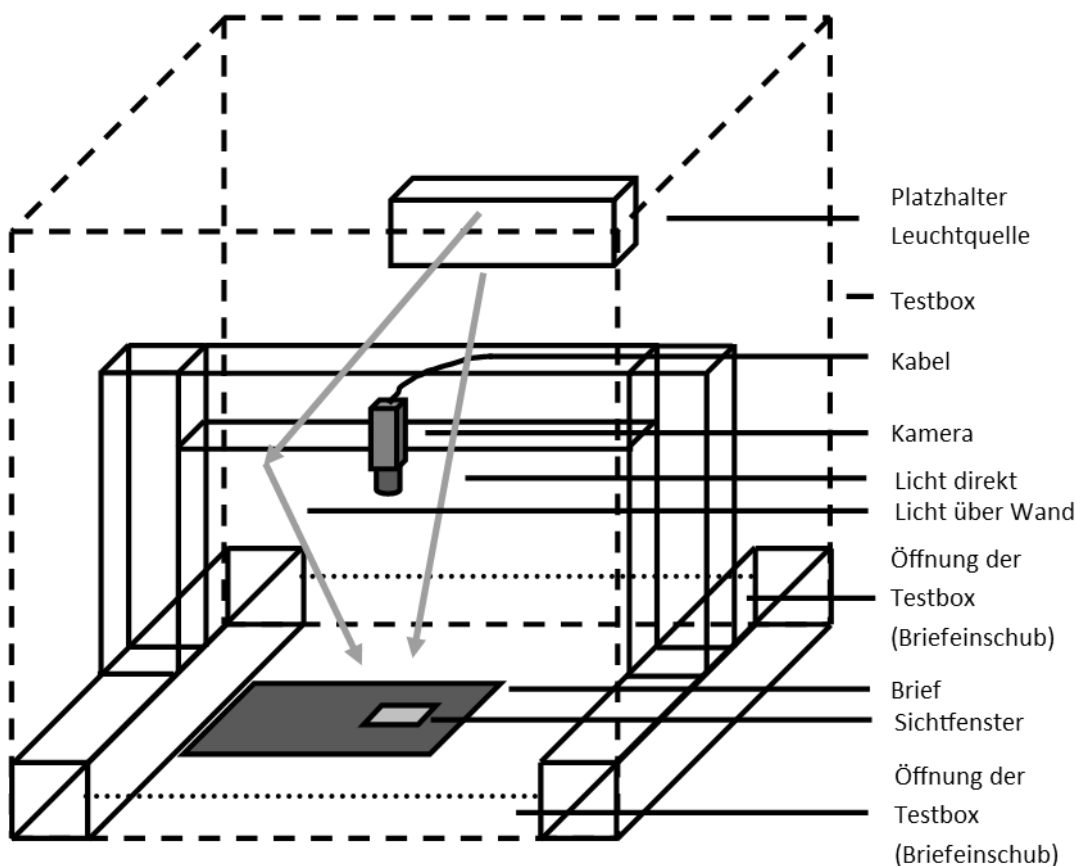
Die durch definierte Beleuchtung erzeugten und von der Kamera aufgenommenen Bilder müssen gespeichert und visualisiert werden. Außerdem müssen die Parameter der Kamera eingestellt werden können. Dazu wird ein Computer mit Ethernet- (LAN-) Anschluss und ein Ethernetkabel benötigt.

Zur Erzeugung der Testbilder wird die Software „IC Capture 2.4“ verwendet. Anschließend werden die Bilder mit dem menschlichen Sehsinn analysiert. Werden diese für tauglich befunden, so werden sie dem Texterkennungsprogramm „Tesseract“ zugeführt. Dieses Programm erzeugt eine Textdatei, welche mit dem Original verglichen wird.

5.3 Aufbau

Auf der Suche nach einer preiswerten Abschirmung des in Kapitel 4 erwähnten Gestells stand eine Pappkartonverpackung im Weg. Solch ein Quader aus Pappe schien genau das Richtige zu sein. Schnell war einer in der benötigten Größe gefunden und das Gestell samt Papier drinnen installiert. Durch ein Loch im Deckel des Kartons wurden die Kabel der Kamera herausgeführt. Anschließend wurden zwei Schlitze an der unteren Seite zweier gegenüberliegender Seiten hineingeschnitten, durch die später die Testbriefe geschoben werden sollen. Auf einer der zwei Seiten entsteht in der Mitte ein Loch für die Beleuchtungen. Für den ersten Teilversuch wird das Papier mit der kleinen aufgedruckten Adresse gefaltet und in den kleinsten glatten Briefumschlag gesteckt. Anschließend wird es durch einen der Schlitze in den Karton gegeben, welcher noch völlig ohne Beleuchtung ist. Jede Beleuchtung wird einmal so angebracht, dass das Licht direkt auf das Sichtfenster des Briefes scheint, und einmal so, dass es zuerst die gegenüberliegende Wand anstrahlt und von dort auf den Brief fällt (siehe Bild)





5.4 Durchführung

Zu Beginn wird das Metallgerüst mit der Kamera auf das Papier mit den vorgezeichneten Umrissen (siehe Kapitel 4) gestellt. Anschließend werden die Briefe vorbereitet. Dazu werden die Briefblätter wie in der oben abgebildeten Zeichnung vorgefaltet. Nun wird die Kamera an den Laptop angeschlossen und das Computerprogramm IC Capture 2.4 gestartet. Folgend wird die Auflösung der Kamera auf 1920 x 1080 gestellt.

Nun wird das erste Briefpapier (Schriftgröße 12) in den glatten C5/6-Umschlag geschoben. Dieser wird unter die Kamera gelegt und im Computerprogramm werden die Parameter nachgestellt, bis ein Bild entsteht, welches Eine schwarze Schrift auf einem fast bis ganz weißem Hintergrund zeigt. Wurden die Parameter kalibriert, werden das entstandene Bild und die Parametereinstellungen gespeichert. Die Parametereinstellungen werden beibehalten, nur das Papier im Briefumschlag wird gewechselt. Wieder wird das Bild gespeichert. Dies erfolgt für die Schriftgrößen 12 bis 6.

Als nächstes wird der vorangehende Abschnitt für den zerknitterten C5/6-Umschlag wiederholt. C5- und C4-Umschläge folgen nach gleichem Muster.

Wurden alle Bilder für das Tageslicht gemacht, wird das Gestell der Kamera in einen Pappkarton gestellt. Dieser schirmt das Tageslicht ab und garantiert fehlerfreie Ergebnisse. Als erste Beleuchtung wird eine Lampe bestehend aus drei LEDs, welche das Licht

punktförmig abstrahlt, verwendet. Diese wird direkt auf das Sichtfenster des Briefes gerichtet. Anschließend wird für jede Schriftgröße und jede Briefart ein Bild gemacht und gespeichert. Nun wird die Lichtquelle auf die gegenüberliegende Wand gerichtet, damit das Licht dort zerstreut in alle Richtungen reflektiert wird. Wieder werden Bilder von allen Beispielobjekten gemacht. Dies wird für die Stableuchte, Die LED-Matrizen und die Glühbirnen wiederholt.

Nachdem alle Bilder gemacht wurden, werden die Ergebnisse analysiert und verglichen. Anschließend werden die Bilder dem Texterkennungsprogramm zugeführt. Die zurückgelieferten Texte werden ebenfalls analysiert, um eine Aussage über die Qualität der Lichtquelle abgeben zu können.

5.5 Ergebnisse

Als erstes ist zu beachten, dass bei den meisten Leuchtquellen die Kameraparameter nicht so eingestellt werden konnten, dass ein Bild ohne Graustufen entsteht, auf dem die Schrift Schwarz und der Hintergrund weiß ist. Bei Tageslicht konnten die besten Ergebnisse erzielt werden, jedoch ist die Intensität des Tageslichts zeit- und wetterabhängig. Die punktförmige Lichtquelle war so schwach, dass durch das vergrößern der Parameterwerte Pixelfehler entstanden. So konnte das Bild nicht scharf gestellt werden. Bei direkter Bestrahlung kam es zu Lichtstreifenbildung in der Nähe der Knicke. Aber auch die Wölbungen sorgten bereits für Reflektionen. Die Stableuchte hatte mehr Kraft, jedoch reichte es auch hier nicht für eine vollständige Scharfstellung. Bei indirekter Beleuchtung waren die Lichtirritationen durch Wölbungen bereits verschwunden. Die Knicke jedoch erzeugten weiterhin Fehler. Das breite Lichtfeld der Matrizen reduzierte diese Fehler weiter. Um die Fehler ganz zu unterbinden bedarf es noch eines Streufilters. Das diffuse, von den Glühbirnen ausgestrahlte Licht erzielte nach dem Tageslicht die besten Ergebnisse. Aber die Glühbirne allein war zu schwach für so einen großen Raum. Des Weiteren gibt es kaum noch Glühbirnen zu kaufen, daher siegen die LED-Matrizen. Ihr Licht muss noch mit einer vorgebauten Streuscheibe, welche eine einfache milchige Acrylglasscheibe sein kann, zerstreut werden. Es sollten zwei Matrizen verbaut werden, welche an gegenüberliegenden Wänden montiert werden und sich gegenseitig bescheinen. So entsteht ein nahezu so diffuses Licht, wie das Tageslicht. Aber die Intensität des Lichtes bleibt konstant.

Wichtige Worte werden teilweise unterstrichen. Um dem Kunden nichts falsches zu versprechen, wird auch diese Möglichkeit geprüft. Da „Tesseract“ nach zusammenhängenden Bildpunkten einer Farbe sucht, jedoch keine Zeichengröße / Zeilengröße kennt, ist davon auszugehen, dass „Tesseract“ doppelte Unterstriche als „=-“-Zeichen auffasst.

In Firmenbezeichnungen kommen die verschiedensten Grafikelemente vor. So kann es sein, dass im Firmenname durchgestrichene Buchstaben vorkommen. In diesem Versuch soll ebenfalls herausgefunden werden, ob solche Namen lesbar sind oder ob der Firmenname von der Texterkennung in so einem Fall ausgeschlossen werden muss.

Bei den Attributen „hochgestellt“ und „tiefgestellt“ verhält es sich ebenso, wie mit den Durchstreichungen.

Da die Kamera nur Schwarz-Weiß sieht, sollte auch nur schwarze Schrift auf weißem Hintergrund stehen. Die Kamera könnte auch weiße Schrift von schwarzem Hintergrund trennen, jedoch denkt „Tesseract“, dass die schwarzen Bildpunkte die Informationen enthalten. Intensive dunkle Farben können eingeschränkt eingesetzt werden, da Monochrom-Kameras alle Farben in Grautönen darstellen. Das Programm „Tesseract“ benötigt jedoch einen hohen Kontrast zw. Text und Hintergrund.

So wird ein Adressfeld in 8 verschiedenen Schriftarten und mit durchgestrichenem Firmennamen in der Schriftgröße „6“ und „20“ benötigt. Des Weiteren wird jede Schriftart noch einmal in kursiv, fett sowie in kursiv und fett benötigt. Hierfür ist es ausreichend, diese Stilarten auf die Schrift der Größe „6“ anzuwenden. Die Attribute hoch- und tiefgestellt werden in einige Zeichenketten eingearbeitet, sodass keine eigenen Schriftsätze nötig sind.

6.3 Aufbau / Durchführung

Es wird der gleiche Aufbau wie in Kapitel 5 benutzt. Als Beleuchtung werden die LED-Matrizen eingesetzt. Unter die Kamera werden in diesem Experiment Briefblätter mit unterschiedlichen Schriftarten und Größen gelegt. Von jedem Papier wird ein Bild gemacht. Nachdem alle Bilder gemacht wurden, werden diese durch das Texterkennungsprogramm Tesseract in Text umgewandelt. Anschließend wird berechnet, wie sehr der ermittelte Text dem Original entspricht.

6.4 Ergebnisse

Arial konnte problemlos bis Schriftgröße 6 erkannt werden. Courier New stellte ebenfalls kein Hindernis dar. Von allen anderen Schriftstilen konnten nur einige Buchstaben richtig erkannt werden. Soll eine dieser Schriften verwendet werden, so müsst sie Tesseract erst gelernt werden.

7 Experiment 3 "Sonderlaute":

7.1 Ziel

Das deutsche Alphabet besteht aus 26 Buchstabe, zehn Ziffern und Sonderzeichen wie zum Beispiel ä, ö, ü, ß. Diese kommen jedoch auch in Namen vor. Durch die Experimente in den vorhergehenden Kapiteln wurde bereits ermittelt, dass das reguläre Alphabet problemlos gelesen werden kann. Ob auch die Sonderzeichen korrekt erkannt werden, wird dieses Experiment zeigen. Ist dies nicht der Fall, so muss ein Lerndokument für das Texterkennungsprogramm entwickelt werden.

7.2 Material / Aufbau

Als Material wird nur das Gestell aus Kapitel 5, Papier und Tinte benötigt. Auf dem Papier werden 90 x 45 mm große Bereiche abgegrenzt. In einen dieser Bereiche werden die einzelnen Zeichen untereinander geschrieben. Ein weiterer wird mit Vornamen gefüllt. Abgerundet wird es von einem Bereich für Städtenamen. Das Papier, auf dem all diese Bereiche sind wird dann Stück für Stück unter die Kamera gelegt, sodass jeder Name auf einem von der Kamera fotografierten Bild zu sehen ist. Diese Bilder werden dann „Tesseract“ zugeführt und dort in Text umgewandelt. Anschließend wird der Text mit dem Originalwort verglichen.

7.3 Durchführung

Das Papier wird so unter die Kamera gelegt, dass nur ein Bereich im Sichtfeld der Kamera ist. Anschließend werden die Parameter wie in Kapitel 5 für die LED-Matrizen eingestellt. Nun wird ein Bild gemacht und durch das Texterkennungsprogramm bearbeitet. Danach wird das Papier verschoben und der nächste Bereich untersucht.

7.4 Ergebnisse

Tesseract kann keine einzelnen Sonderbuchstaben erkennen. Wenn diese aber in bekannten Wörtern vorkommen, so ist es möglich, dass Tesseract dieses richtig errät. Kommen diese jedoch in Namen vor, so werden diese fehlinterpretiert. Da es viele Namen mit Sonderzeichen gibt muss Tesseract auch auf Sonderlaute trainiert werden.

8 Programmierung

Die Firma TheImageSource liefert zu ihren Kameras auch ein Programm und Quellcode um eigene Programme zu schreiben. Ein Beispielprogramm bildet die Grundlage, denn viele der dort enthaltenen Codezeilen können gleich verwendet werden, was viel Zeit spart. Der selbstgeschriebene Code wird in einem ähnlichen Kommentar wie folgenden eingebettet:

```
//-- |* ',,__,,' *| -- Thread.Variablen -- |* ',,__,,' *| --//
```

Zuerst wurden die Grenzen und Möglichkeiten des Beispielprogramms erörtert. Dabei stellt sich heraus, dass einige Funktionen für die Texterkennung unnötig sind. Diese wurden sofort gelöscht. Anschließend wurden die Anforderungen an das neue Programm ermittelt.

Die Bildspeicherung und die Texterkennung sind zeitkritisch, dies bedeutet, die langwierigen Prozesse müssen innerhalb von 2 s erfolgen. Dauert es länger, kommt es zu einer falschen Zuordnung. Daher wurden diese zwei Teilaufgaben in einen neuen Thread ausgelagert. Aber auch das Hauptprogramm wurde schlang gehalten.

Die Erkennung sollte mit einem Observer statt einem Thread erfolgen, jedoch konnte der Observer nicht mit dem Element ICLicensingControl arbeiten. Dieses wird aber benötigt, um die Bildinformationen von der Kamera zu holen. Den Code für diesen Versuch befindet sich in den Anlagen, Teil 3.

Der Thread kann über einen Menüeintrag gestartet werden. Dabei wird das ICLicensingControl des Hauptprogrammes geschlossen und dessen Parameter an das Control des Threads übergeben. Des Weiteren wird auch der Speicherpfad übergeben, welcher vom Programmbenutzer über einen SaveFileDialog eingegeben werden kann.

```
//-- |* ',,__,,' *| -- startet den Thread -- |* ',,__,,' *| --//
private void startToolStripMenuItem_Click(object sender, EventArgs e)
{
    string devstr = icImagingControl1.SaveDeviceState();
    label2.Text = devstr;
    Work.label3.Text = label2.Text; //--Daten ueber Kameraeigenschaften
    Work.textbox2 = textbox1; //-- Speicherort --//

    Bildsuche = new Thread(new ThreadStart(Work.SavePicture));
    icImagingControl1.LiveStop(); //-- beendet Kameraverbindung des --//
    //-- Hauptthreads, damit Nebenthread eine Verbindung zur Kamera --//
    //-- aufbauen kann; --//
    flag = true; //-- schaltet Threadschleife frei --//
    Bildsuche.Start();
}
```

Im Thread für die Erkennung befindet sich eine Endlosschleife. In diese müsste noch ein Eventhandler eingebaut werden. Der Trigger würde dann den Event auslösen. Mit einem

Flag (Ja / Nein) kann die Schleife abgebrochen werden, damit der Thread beendet werden kann. Dies geschieht über einen weiteren Menüeintrag im Hauptprogramm.

```
//-- |* '','_,_,_' *| -- beendet den Thread -- |* '','_,_,_' *| --//  
private void stopToolStripMenuItem1_Click(object sender, EventArgs e)  
{  
    label2.Text = Work.label3.Text;//-- Devicestate --//  
    label1.Text = Work.label4.Text;//-- Text auf dem Bild --//  
    flag = false;  
}
```

Im Thread selbst werden zur Sicherheit Standardspeicherort und Standarddateiname vergeben und ein ICImagingControl mit den Parametern des Hauptthreads gestartet. Es schließt sich die Endlosschleife an. Über einen Zähler wird eine Nummer an den Dateinamen angefügt, damit alle Bilder zur nachträglichen Kontrolle erhalten bleiben. Nach dem Abholen und dem Speichern der Bilder wird das Programm Tesseract aufgerufen. Dieses liefert einen Text zurück, welcher dann mit den Informationen der ICS-Mailing-Maschine abgeglichen werden kann.

Das Gesamtprogramm findet sich in den Anlagen, Teil 1 + 2.

Literatur

- [Abbe2016] Abbey: OCR. URL: <<http://www.abbyy.com/finereader/about-ocr/what-is-ocr/>>, verfügbar am 05.02.2016
- [ArcG2015] ArcGis for Desktop: Tabelle. URL: <<http://desktop.arcgis.com/de/desktop/latest/manage-data/tables/what-are-tables-and-attribute-information.htm>>, verfügbar am 10.02.2016
- [Auer2015] Auer, Jürgen: Datenbank versus Datenbank-Management-System. URL: <<http://www.sql-und-xml.de/sql-tutorial/datenbank-grundbegriffe.html>>, verfügbar am 10.02.2016
- [Bamb2016] Universität Bamberg: Texterkennung für gescannte Dokumente. URL: <https://www.uni-bamberg.de/fileadmin/uni/wissenschaft_einricht/universitaetsbibliothek/Digitalisierung/pdf2txt-doku.pdf>, verfügbar am 05.02.2016
- [Betr2014] Vorlesung Betriebssysteme von Herrn Prof.
- [Born1985] Born, Max: Optik. – 3.Aufl. – Berlin: Springer-Verlag Berlin Heidelberg, 1985
- [Bras2005] Brass, Stefan: Datenbanken I. URL: <http://users.informatik.uni-halle.de/~brass/db05/da_tabdf.pdf>, verfügbar m 10.02.2016 (Universität Halle)
- [Dude2016] Duden; Bibliographisches Institut GmbH: Parameter. URL: <<http://www.duden.de/rechtschreibung/Parameter>>, verfügbar am 04.02.2016
- [Dude2016-2] Duden: Software. URL: <<http://www.duden.de/rechtschreibung/Software>>, verfügbar am 28.01.2016
- [Dude2016-3] Duden: Datei. URL: <<http://www.duden.de/rechtschreibung/Datei>>, verfügbar am 10.02.2016

- [Dude2016-4] Duden: Hardware. URL: < <http://www.duden.de/rechtschreibung/Hardware>>, verfügbar am 10.02.2016
- [Gabl2016] Gabler Wirtschaftslexikon: Software. URL: <<http://wirtschaftslexikon.gabler.de/Archiv/56933/software-v1.html>>, verfügbar am 28.01.2016
- [Gamm2001] Gamma, Erich: Entwurfsmuster. – 5.Aufl. – Bonn: Pearson Education Deutschland GmbH, 2001
- [Gran1989] Granger, Pierre Marie: Die Optik in der Bildgestaltung. – 1.Aufl. – Würzburg: Vogel Buchverlag, 1989
- [ITWi2016] ITWissen: Datei. URL: <<http://www.itwissen.info/definition/lexikon/Datei-file.html>>, verfügbar am 10.02.2016
- [ITWi2016-2] ITWissen: Array. URL: <<http://www.itwissen.info/definition/lexikon/Array-array.html>>, verfügbar am 10.02.2016
- [Koch2009] Kochan, Stephen G.: Objective-C 2.0. – 1.Aufl. - : Pearson Education Deutschland GmbH, 2009
- [Krat2014] Kratzenberg, Marco: OCR Freeware für die automatische Texterkennung. URL: <<http://www.giga.de/software/office-business/automatische-texterkennung-mit-ocr-freeware/>>, verfügbar am 05.02.2016
- [Kühl2004] Kühlke, Dietrich: Optik. – 2.Aufl. – Frankfurt am Main: Wissenschaftlicher Verlag Harri Deutsch GmbH, 2004
- [Küpp1992] Küppers, Harald: Schule der Farben. – 1.Aufl. – Köln: DuMont Buchverlag, 1992
- [LiTa1997] Lipson, Stephen G.; Lipson, Henry S.; Tannhauser, David S.: Optik. 1.Aufl. – Berlin: Springer-Verlag Berlin Heidelberg, 1997
- [Nemc1993] Nemcsics, Antal: Farbenlehre und Farbdynamik. – 1.Aufl. – Göttingen: Muster-Schmidt Verlag, 1993

- [PeBa2005] Pedrotti, Frank S.; Pedrotti, Leno S.; Bausch, Werner; Schmidt Hartmut: Optik für Ingenieure. – 1.Aufl. – Heidelberg: Springer-Verlag Berlin Heidelberg, 2005
- [Ubun2015] Ubuntu documentation: OCR. URL: <<https://help.ubuntu.com/community/OCR>>, verfügbar am 05.02.2016
- [Vett1998] Vetter, Max: Objektmodellierung. – 2.Aufl. – Stuttgart: Teubner, 1998
- [Wiki2015] Wikipedia: Software. URL: <<https://de.wikipedia.org/wiki/Software>>, verfügbar am 28.01.2016
- [Youn1997] Young, Matt: Optik, Laser, Wellenleiter. – 1.Aufl. – Berlin: Springer-Verlag Berlin Heidelberg, 1997
- [Wiki2015-2] Wikipedia: Zellophan. URL: <<https://eu.wikipedia.org/wiki/Zelofan>>, verfügbar am 12.02.2016
- [Spek2016] Spektrum: Kontrast. URL: <<http://www.spektrum.de/lexikon/optik/kontrast/1639>>, verfügbar am 08.04.2016

Anlagen

Teil 1	I
Teil 2	XI
Teil 3	XII

Anlagen, Teil 1

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using TIS.Imaging;                                //++For using camera tools

namespace demoapp
{
    public partial class Form1 : Form
    {
        //-- |* ',,__,,' *| -- Thread.Variablen -- |* ',,__,,' *| --//
        Thread Bildsuche = null;
        static bool flag;

        //-- |* ',,__,,' *| -- Methoden -- |* ',,__,,' *| --//
        public Form1()
        {
            InitializeComponent();
            label2.Hide();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            LoadLastUsedDevice();
            UpdateControls();
        }

        /// <summary>
        /// exitToolStripMenuItem_Click
        ///
        /// Exit the demo application
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Close();
        }

        /// <summary>
        /// deviceToolStripMenuItem_Click
        ///
        /// Show the device selection dialog.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void deviceToolStripMenuItem_Click(object sender, EventArgs e)
        {
            SelectDevice();
        }

        /// <summary>
```

```

    /// playToolStripMenuItem_Click
    ///
    /// Handler for the play menu item. StartLiveVideo() sub is called.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void playToolStripMenuItem_Click(object sender, EventArgs e)
    {
        StartLiveVideo();
    }

    /// <summary>
    /// stopToolStripMenuItem_Click
    ///
    /// Stop the live video display by a call to StopLiveVideo() sub.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void stopToolStripMenuItem_Click(object sender, EventArgs e)
    {
        StopLiveVideo();
    }

    private void Form1_Resize(object sender, EventArgs e)
    {
        icImagingControl1.Size = ClientSize;
    }

    /// <summary>
    /// imageToolStripMenuItem_Click
    ///
    /// Show the video and camera properties dialog.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void imageToolStripMenuItem_Click(object sender, EventArgs e)
    {
        ShowProperties();
    }

    /// <summary>
    /// saveImageToolStripMenuItem_Click
    ///
    /// Handler for the SaveImage menu. SaveImage() Sub is called.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void saveImageToolStripMenuItem_Click(object sender, EventArgs e)
    {
        SaveImage();
    }

    private void externalTriggerToolStripMenuItem_Click(object sender, EventArgs
e)
    {
        ToggleTrigger();
    }

    /// <summary>
    /// UpdateControls
    ///
    /// UpdateControls enables and disables the menuitems of the application de-
pending of

```

```

/// camera is connected to the computer or not.
/// </summary>
private void UpdateControls()
{
    bool IsDeviceValid = false;

    IsDeviceValid = icImagingControl1.DeviceValid;
    imageToolStripMenuItem.Enabled = IsDeviceValid;
    saveImageToolStripMenuItem.Enabled = IsDeviceValid;
    playToolStripMenuItem.Enabled = IsDeviceValid;
    stopToolStripMenuItem.Enabled = IsDeviceValid;
    ToolbarPlayButton.Enabled = IsDeviceValid;
    ToolbarStopButton.Enabled = IsDeviceValid;

    externalTriggerToolStripMenuItem.Enabled = false;
    ToolbarTriggerButton.Enabled = false;

    if (IsDeviceValid)
    {
        playToolStripMenuItem.Enabled = !icImagingControl1.LiveVideoRunning;
        stopToolStripMenuItem.Enabled = !playToolStripMenuItem.Enabled;
        ToolbarPlayButton.Enabled = !icImagingControl1.LiveVideoRunning;
        ToolbarStopButton.Enabled = !ToolbarPlayButton.Enabled;

        if (icImagingControl1.DeviceTriggerAvailable)
        {
            externalTriggerToolStripMenuItem.Enabled = true;
            ToolbarTriggerButton.Enabled = true;
            externalTriggerToolStripMenuItem.Checked = icImagingControl1.De-
viceTrigger;
            ToolbarTriggerButton.Checked = icImagingControl1.DeviceTrigger;
        }

        if (icImagingControl1.InputChannelAvailable)
        {
            //Create the sub menus that allow to select the input channels.
            //ToolBarInputChannel.DropDownItems.Clear();
            //inputChannelsToolStripMenuItem.DropDown.Items.Clear();
            //inputChannelsToolStripMenuItem.Enabled = true;
            //ToolBarInputChannel.Enabled = true;

            foreach (InputChannel s in icImagingControl1.InputChannels)
            {
                System.Windows.Forms.ToolStripItem mitem = null;

                // Add the input channel as menu item to the main menu
                //mitem = inputChannelsToolStripMenuI-
tem.DropDown.Items.Add(s.ToString());

                //mitem = inputChannelsToolStripMenuItem.DropDownI-
tems.Add(s.ToString());

                if (icImagingControl1.InputChannel == s.ToString())
                {
                    ToolStripMenuItem i = (ToolStripMenuItem)mitem;
                    i.Checked = true;
                }
                mitem.Click += new System.EventHandler(mnuInputChannel-
Child_Click);

                // Add the input channel as menu item to the toolbar but-
ton's context menu

```

```

        //mitem = ToolBarInputChan-
nel.DropDown.Items.Add(s.ToString());
        if (icImagingControl1.InputChannel == s.ToString())
        {
            ToolStripMenuItem i = (ToolStripMenuItem)mitem;
            i.Checked = true;
        }
        mitem.Click += new System.EventHandler(mnuInputChannel-
Child_Click);
    }
}
else
{
    // Remove the input channels from the submenus.
    //ToolBarInputChannel.DropDownItems.Clear();
    //inputChannelsToolStripMenuItem.DropDown.Items.Clear();
    //inputChannelsToolStripMenuItem.Enabled = false;
    //ToolBarInputChannel.Enabled = false;
}

}
ToolBarSnapButton.Enabled = IsDeviceValid;
ToolBarPropertiesButton.Enabled = IsDeviceValid;
}

/// <summary>
/// toolStrip1_ItemClicked
///
/// Handler for the toolbar play button. StartLiveVideo sub is called.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void toolStrip1_ItemClicked(object sender, ToolStripItemClickedE-
ventArgs e)
{
    // Evaluate the Button property to determine which button was clicked.
    switch (toolStrip1.Items.IndexOf(e.ClickedItem))
    {
        case 0:
            StartLiveVideo();
            break;
        case 1:
            StopLiveVideo();
            break;
        case 3:
            SaveImage();
            break;
        case 5:
            SelectDevice();
            break;
        case 6:
            ShowProperties();
            break;
        case 8:
            ToggleTrigger();
            break;
    }
}

//__##_ausgelagerte Funktionen__##_//
/// <summary>
/// StartLiveVideo
///

```

```

/// Start the live video display and change the button states of the
/// play and stop button.
/// </summary>
private void StartLiveVideo()
{
    if (!flag)
    {
        if (icImagingControl1.DeviceValid)
        {
            icImagingControl1.LiveStart();
            playToolStripMenuItem.Enabled = false;
            stopToolStripMenuItem.Enabled = true;
            ToolbarPlayButton.Enabled = false;
            ToolbarStopButton.Enabled = true;
            ToolbarSnapButton.Enabled = true;
        }
    }
}

/// <summary>
/// StopLiveVideo
///
/// Stop the live video display and change the button states of the
/// play and stop button.
/// </summary>
private void StopLiveVideo()
{
    if (!flag)
    {
        if (icImagingControl1.DeviceValid)
        {
            icImagingControl1.LiveStop();
            playToolStripMenuItem.Enabled = true;
            stopToolStripMenuItem.Enabled = false;
            ToolbarPlayButton.Enabled = true;
            ToolbarStopButton.Enabled = false;
            ToolbarSnapButton.Enabled = false;
        }
    }
}

/// <summary>
/// SelectDevice
///
/// Show the device selection dialog.
/// </summary>
private void SelectDevice()
{
    if (!flag)
    {
        if (icImagingControl1.LiveVideoRunning)
        {
            icImagingControl1.LiveStop();
            icImagingControl1.ShowDeviceSettingsDialog();
            if (icImagingControl1.DeviceValid)
                icImagingControl1.LiveStart();
        }
        else
            icImagingControl1.ShowDeviceSettingsDialog();

        UpdateControls();
        SetWindowSizeToImagesize();
    }
}

```

```

        DeviceSettings.SaveSelectedDevice(icImagingControl1);
    }
}

/// <summary>
/// ShowProperties
///
/// Show the property dialog of the current video capture device.
/// </summary>
private void ShowProperties()
{
    if (icImagingControl1.DeviceValid)
    {
        icImagingControl1.ShowPropertyDialog();
        UpdateControls();
    }
}

/// <summary>
/// SaveImage
///
/// Snap (capture) an image from the video stream and save it to harddisk.
/// </summary>
private void SaveImage()
{
    if (!flag)
    {
        if (icImagingControl1.DeviceValid)
        {
            // Snap (capture) an image to the memory
            icImagingControl1.MemorySnapImage();

            // Call the save file dialog to enter the file name of the image
            SaveFileDialog saveFileDialog1 = new SaveFileDialog();
            saveFileDialog1.Filter = "bmp files (*.bmp)|*.bmp|All files
(*.*)|*.*.txt";

            saveFileDialog1.FilterIndex = 1;
            saveFileDialog1.RestoreDirectory = true;

            if (saveFileDialog1.ShowDialog() == DialogResult.OK)
            {
                // Save the image.
                icImagingControl1.MemorySaveImage(saveFileDialog1.FileName);
            }
        }
    }
}

/// <summary>
/// ToggleTrigger
///
/// Enable or disable the external trigger.
/// </summary>
private void ToggleTrigger()
{
    if (!flag)
    {
        if (icImagingControl1.DeviceValid)
        {
            if (icImagingControl1.DeviceTriggerAvailable)
            {
                bool trigger = !icImagingControl1.DeviceTrigger;
            }
        }
    }
}

```

```

        if (icImagingControl1.LiveVideoRunning)
        {
            icImagingControl1.LiveStop();
            icImagingControl1.DeviceTrigger = trigger;
            icImagingControl1.LiveStart();
        }
        else
        {
            icImagingControl1.DeviceTrigger = trigger;
        }

        externalTriggerToolStripMenuItem.Checked = trigger;
        ToolbarTriggerButton.Checked = trigger;
    }
}

/// <summary>
/// SetWindowSizeToImagesize
///
/// Resize the application to the size of the video format.
/// </summary>
private void SetWindowSizeToImagesize()
{
    if (!flag)
    {
        if (icImagingControl1.DeviceValid)
        {
            // Change the width to a minimum size of 230 pixel. If the
            // less than 230 pixel, the toolbar can not be seen completely.
            Width = Math.Max(icImagingControl1.ImageWidth + Width - ClientSize.Width, 230);
            Height = icImagingControl1.ImageHeight + Height - ClientSize.Height + toolStrip1.Size.Height;
        }
    }
}

/// <summary>
/// mnuInputChannelChild_Click
///
/// If the video capture device supports input channels, submenus
/// for the input channel selection are created. This is the handler
/// sub for the input channel submenus.
/// The menu text if the clicked submenu item is retrieved and assigned
/// to the InputChannel property of IC15Control.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void mnuInputChannelChild_Click(object sender, System.EventArgs e)
{
    System.Windows.Forms.ToolStripItem mitem = null;
    mitem = ((System.Windows.Forms.ToolStripItem)(sender));

    /*
        foreach (System.Windows.Forms.ToolStripItem Item in inputChannelsToolStripMenuItem.DropDownItems)
        {
            if (Item.Text == mitem.Text)
                Item.Checked = true;
            else
                Item.Checked = false;
        }
    */
}

```

```

    }

    foreach (System.Windows.Forms.ToolStripItem Item in ToolBarInputChannel.DropDown.Items)
    {
        if (Item.Text == mitem.Text)
            Item.Checked = true;
        else
            Item.Checked = false;
    }
    */

    icImagingControl1.InputChannel = mitem.Text;
    // Save the new input channel to the registry.
    DeviceSettings.SaveSelectedDevice(icImagingControl1);
}

/// <summary>
/// LoadLastUsedDevice
///
/// Retrieve the last used video capture device from the registry and
/// open it. The implementation of these functionality is in the
/// modul devicesettings.vb
/// </summary>
private void LoadLastUsedDevice()
{
    if (!flag)
    {
        DeviceSettings.OpenSelectedDevice(icImagingControl1);
        SetWindowSizeToImageSize();
    }
}

//-- |* ',,__,,' *| -- Nebenthread -- |* ',,__,,' *| --//
//-- fuer das sammeln der Bilder, die anschließende Texterkennung, --//
//-- und der Textvergleich --//
public class Work
{
    public static TextBox textbox2 = null;
    public static Label label3 = new Label();
    public static Label label4 = new Label();

    public static void SavePicture()
    {
        int zaehler = 0;
        string Ordner = "";
        const string Bild = "/Beispiel.bmp";
        int Index_Ending = 0;
        ICIImagingControl icImagingControl2 = new ICIImagingControl();
        icImagingControl2.LoadDeviceState(label3.Text, true);
        icImagingControl2.LiveStart();

        int Pwidth=1920, Pheight=1080;
        while (flag)
        {
            if (icImagingControl2.DeviceValid)
            {
                //--Speicerort festlegen--//
                Ordner = @textbox2.Text;
                if (Ordner == "") Ordner = "C:";
                Index_Ending = Ordner.IndexOf(".bmp");
                if (Index_Ending < 0) Ordner = Ordner + Bild;
                Ordner = Ordner.Replace(".bmp",zaehler + ".bmp");
            }
        }
    }
}

```

```

        zaehler = zaehler + 1;

        // Snap (capture) an image to the memory and bmp
        icImagingControl2.MemorySnapImage();
        icImagingControl2.MemorySaveImage(Ordner);

        //--Tesseract_OCR_Verarbeitung--//
        tesseract_ocr_class ocr = new tesseract_ocr_class(
            "C:\\Program Files\\Tesseract-OCR\\tesseract.exe");
        Bitmap beispiel = new Bitmap(Ordner);
        //Bitmap bsp = new Bitmap(Pwidth, Pheight,
        //    icImagingControl2.MemoryPixelFormat);
        //string result = ocr.OCRFromBitmap(bsp);
        string result = ocr.OCRFromBitmap(beispiel);
        label4.Text = result;

        //--Textverarbeitung--//

        string Adresse = @"C:\Program Files\Tesseract-OCR\text.txt";

        //--Textvergleich--//
    }

}
icImagingControl2.LiveStop();
}

}

//-- |* '','','__','' *| -- startet den Thread -- |* '','','__','' *| --//
private void startToolStripMenuItem_Click(object sender, EventArgs e)
{
    string devstr = icImagingControl1.SaveDeviceState();
    label2.Text = devstr;
    Work.label3.Text = label2.Text; //--Daten ueber Kameraeigenschaften
    Work.textbox2 = textbox1; //-- Speicherort --//

    Bildsuche = new Thread(new ThreadStart(Work.SavePicture));
    icImagingControl1.LiveStop(); //-- beendet Kameraverbindung des --//
    //-- Hauptthreads, damit Nebenthread eine Verbindung zur Kamera --//
    //-- aufbauen kann; --//
    flag = true; //-- schaltet Threadschleife frei --//
    Bildsuche.Start();
}

//-- |* '','','__','' *| -- beendet den Thread -- |* '','','__','' *| --//
private void stopToolStripMenuItem1_Click(object sender, EventArgs e)
{
    label2.Text = Work.label3.Text; //-- Devicestate --//
    label1.Text = Work.label4.Text; //-- Text auf dem Bild --//
    flag = false;
}

//-- |* '','','__','' *| -- Definition des Speicherortes -- |* '','','__',''
*| --//
//-- Ermittlung des Speicherpfades per SpeicherOrdnerDialog; --//
//-- und Speicherung in eine Textbox --//

```

```

private void speicherortToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (!flag) //--darf nicht waehrend Nebenthread laufen
    {
        SaveFileDialog saveFileDialog1 = new SaveFileDialog();
        saveFileDialog1.Filter = "bmp files (*.bmp)|*.bmp|All files
(*.*)|*.*";
        saveFileDialog1.FilterIndex = 1;
        saveFileDialog1.RestoreDirectory = true;
        if (saveFileDialog1.ShowDialog() == DialogResult.OK)
        {
            // Save the first image
            icImagingControl1.MemorySnapImage();
            icImagingControl1.MemorySaveImage(saveFileDialog1.FileName);
            textBox1.Text = saveFileDialog1.FileName;
        }
    }
}
}

```

Anlagen, Teil 2

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using System.Drawing;
using System.Diagnostics;

namespace demoapp
{
    public class tesseract_ocr_class
    {
        private string commandpath;
        private string outputpath;
        private string tmppath;

        public tesseract_ocr_class(string commandpath)
        {
            this.commandpath = commandpath;
            tmppath = System.Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + @"\out.tif";
            outputpath = System.Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + @"\out.txt";
        }

        public string analyze(string filename)
        {
            string args = filename + " " + outputpath.Replace(".txt", "");
            ProcessStartInfo startinfo = new ProcessStartInfo(commandpath, args);
            startinfo.CreateNoWindow = true;
            startinfo.UseShellExecute = false;
            Process.Start(startinfo).WaitForExit();

            string ret = "";
            using (StreamReader r = new StreamReader(outputpath))
            {
                string content = r.ReadToEnd();
                ret = content;
            }
            File.Delete(outputpath);
            return ret;
        }

        public string OCRFromBitmap(Bitmap bmp)
        {
            bmp.Save(tmppath, System.Drawing.Imaging.ImageFormat.Tiff);
            string ret = analyze(tmppath);
            File.Delete(tmppath);
            return ret;
        }

        public string OCRFromFile(string filename)
        {
            return analyze(filename);
        }
    }
}
```

Anlagen, Teil 3

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using TIS.Imaging;
using System.Threading;

namespace demoapp
{
    class Class1
    {
        public void DoWork()
        {

        }

    }

    public struct BoolObser
    {
        bool flag;

        public BoolObser(bool flag_)
        {
            flag = flag_;
        }

        public bool Flag
        { get { return this.flag; } }
    }

    public class BoolObserTracker : IObservable<BoolObser>
    {
        public BoolObserTracker()
        {
            observers = new List<IObserver<BoolObser>>();
        }

        private List<IObserver<BoolObser>> observers;

        public IDisposable Subscribe(IObserver<BoolObser> observer)
        {
            if (!observers.Contains(observer))
                observers.Add(observer);
            return new Unsubscriber(observers, observer);
        }

        private class Unsubscriber : IDisposable
        {
            private List<IObserver<BoolObser>> _observers;
            private IObserver<BoolObser> _observer;
        }
    }
}
```

```

        public Unsubscriber(List<IObserver<BoolObser>> observers, IObserver<BoolObser> observer)
        {
            this._observers = observers;
            this._observer = observer;
        }

        public void Dispose()
        {
            if (_observer != null && _observers.Contains(_observer))
                _observers.Remove(_observer);
        }

        public void TrackBoolObser(Nullable<BoolObser> loc)
        {
            foreach (var observer in observers)
            {
                if (!loc.HasValue)
                    observer.OnError(new BoolObserUnknownException());
                else
                    observer.OnNext(loc.Value);
            }
        }

        public void EndTransmission()
        {
            foreach (var observer in observers.ToArray())
                if (observers.Contains(observer))
                    observer.OnCompleted();

            observers.Clear();
        }
    }

    public class BoolObserUnknownException : Exception
    {
        internal BoolObserUnknownException()
        { }
    }
}

```


Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Hainichen, den 21. April 2016

André Baumann