

Fakultät für Elektrotechnik und Informationstechnik



Lehrstuhl für
Kommunikations-
technik

Bachelorarbeit B 14-2015

**Tasty Kanalmodell
für die drahtlose Kommunikation
zwischen Gebäuden und
Außeninstallationen**

von

Käpt'n Kevin Blaubär

Abgabedatum: 23. Januar 2018

Prof. Dr.-Ing Rüdiger Kays • Lehrstuhl für Kommunikationstechnik • TU Dortmund

Inhaltsverzeichnis

1 Einleitung	1
1.1 Section	1
2 DaViD	3
2.1 Einführung des DaViD	3
2.2 Systemmodell	4
2.2.1 Modulationsverfahren	5
2.2.2 DatenBlock	8
2.3 Anwendungsgebiete	8
3 ErsteErfahrung	10
3.1 TexLipse spell checking	10
3.2 Enable tikzexternalize for PdfLatex	10
3.3 Forward search with TeXlipse and Sumatra PDF	11
4 ZweiteErfahrung	14
4.1 Allgemeine Struktur	14
4.2 Bildregistrierung	14
4.2.1 SURF	16
4.2.2 RANSAC	20
4.2.3 Kamerakalibrierung	24
4.3 Differenzbild	33
4.4 Bildverarbeitung	35
4.5 QR Musters Detektion	36
5 Auswertung	41
5.1 Section	41
6 Zusammenfassung	42
6.1 Section	42
A Erster Anhang	43
A.1 Section	43

Abbildungsverzeichnis	45
Tabellenverzeichnis	46
Quellcodeverzeichnis	47
Literatur	48

1 Einleitung

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1.1 Section

Jetzt nur noch schreiben! :)

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacinia tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacinia congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacinia commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacinia. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacinia vel est. Curabitur consectetur.

2 DaViD

In diesem Kapitel werden das **Data transmission using Video Devices (DaViD)** System beschrieben. Zuerst läuft die Vorstellung des DaViD Systems. Die Systemmodell und Arbeitsprinzip des Systems werden in anschließenden Abschnitt erläutert. Schließlich folgt die mögliche Anwendungsgebiete des Systems. [1]–[4]

2.1 Einführung des DaViD

DaViD ist ein neuartiges Verfahren zur optischen Freiraum-Datenübertragung zwischen einem Display als Sender und einer Kamera als Empfänger. Ein grundlegendes Übertragungskonzept von DaViD wird in Abbildung 2.1 gezeigt. Ein flaches Display wie ein OLED- oder LCD-Bildschirm zeigt ein Live-Video. Gleichzeitig werden die Daten hinter dem Bild auf die Pixel moduliert. Während die zusätzliche Datenmodulation für menschliche Betrachter nahezu unsichtbar ist, der Benutzer leitet ein hochauflösende Kamera oder ein Smartphone zur Bildschirm, um die Szene aufzunehmen. Durch der eingebaut Prozessor können die Signale decodiert werden.

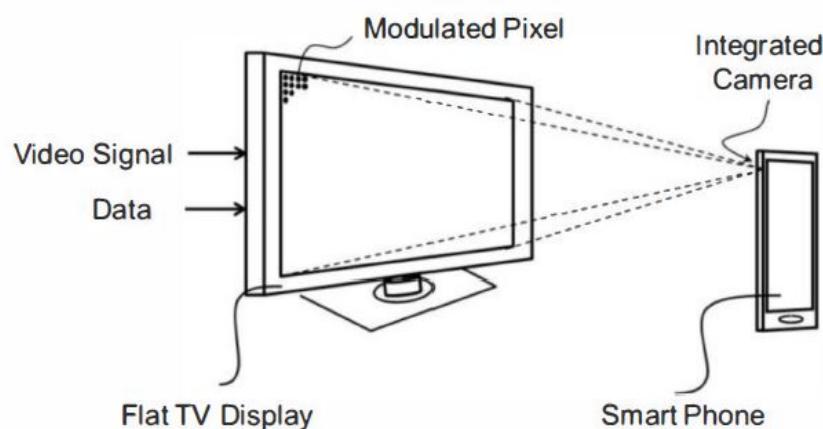


Abbildung 2.1: Eine beispielhafte Implementierung des DaViD-Systems.

2.2 Systemmodell

Bild in Display enthalten eine große Anzahl von Pixeln, die jeweils aus einer spezifischen Anordnung von Subpixelen für die RGB-Farbraum bestehen. Jeder einzelne Frame des Videos wird nämlich durch eine Matrix von Subpixelwerten dargestellt. DaViD System verwendet eine differentielle Modulationsmethod d.h. Teil der Videoinformationen muss wiederholt werden, indem Daten als ein symmetrischer Manchester-Code moduliert und zu den Videosignalkomponenten hinzugefügt werden. In Empfängerseite durch eine zeitliche Synchronisation können die zeitliche inter symbol interference (ISI) vermieden werden. Dann nach Verwendung einer örtlichen Synchronisation enthalten einen Differenzbild. Weil die Randbereich des Differenzbilds ungültig ist, verlässt sich die Modulationsgebiet durch die Verfahren in diese Arbeit entdecken. Danach werden die überlagerten Datensequenz durch eine Reihe von Behandlungen vom Videoinhalt getrennt. Abbildung 2.2 zeigt die schematische Darstellung des DaViD-Systems.



Abbildung 2.2: Schematische Darstellung von DaViD System.

2.2.1 Modulationsverfahren

Ein Modulationsverfahren, das die Videoqualität nicht offensichtlich reduziert garantiert, ist sehr wichtig für ein auf Videogerät basierendes Datenübertragungssystem. Die möglichen Modulationsverfahren in DaViD-System sind:

- Zeitliche differentielle Modulation der Luminanz
- Zeitliche differentielle Modulation der Chrominanz
- Örtliche differentielle Modulation der Luminanz
- Örtliche differentielle Modulation der Chrominanz

Zeitliche differentielle Modulationsverfahren lässt kontinuierliches Paar Frames den gleichen Luminanz- bzw. Chrominanz-Videoinhalt enthalten, d.h. durch Subtrahieren die mit daten addiert Kanal der Paar Frames die Differenzbild erhalten lassen können. Dagegen in örtlichen sind die benachbarte Pixel mit den gleichen Videoamplituden. Hier wird in dieser Arbeit nur zeitliche differentielle Modulation der Chrominanz verwendet. Abbildung 2.3 zeigt ein Blockdiagramm einer typischen Senderimplementierung durch zeitliche differentielle Modulation.

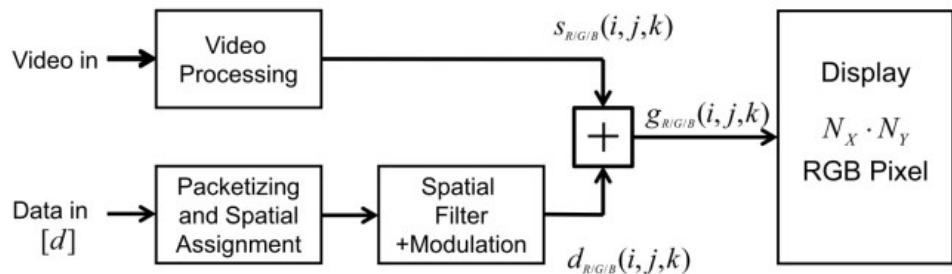


Abbildung 2.3: Blockschaltbild der Signalverarbeitung in zeitlicher differentieller Modulation.

Wir nehmen eine Diaplay an, die in horizontale Richtung N_x Pixel stehen, dagegen in vertikale Richtung N_y Pixel. Das Videoeingangssignal wird verarbeitet, um eine Anzeigeeingabe $s(i, j, k)$ zu liefern. Mit zeitliche differenziell Modulation ist die Videoinhalt des kommenden Frame dasselbe. Die Indizes i und j bedeuten die horizontale und vertikale Pixelposition auf dem Bildschirm, während k die Nummer des reproduzierten Bildes ist. Indiz m heißt den Zähler des Frames in einer Videosequenz. Der Amplitudengrenzenbereich des Videosignals sollte begrenzt sein, um die Addition kleiner Datenamplituden ohne Übersteuern zu ermöglichen.

$$s_{R/G/B}(i, j, k + 1) = s(i, j, k) \quad (2.1)$$

for $0 \leq i < N_x, 0 \leq j < N_y, k = 2 \cdot m, m \in \mathbb{Z}$

Vor Datenübertragung muss der Datenstrom in Schichten der Länge L aufgeteilt werden. Indiz L bedeutet die Menge der Daten, die in einem Framepaar übertragen werden können. Ein direkter Ansatz ist eine direkte Zuordnung von Datenbits zu Pixeltripeln Zeile für Zeile.

$$\begin{aligned}
 d(l) &\rightarrow d(i, j, m) \quad d(l) \in \{-1, 1\} \\
 0 \leq l < L, L &= N_x \cdot N_y \\
 i = l \bmod N_x &\quad j = \lfloor l/N_x \rfloor
 \end{aligned} \tag{2.2}$$

Die Modulationsamplitude A ist ein wichtiger Parameter für Datenübertragung. Im Prinzip kann die Amplitude in verschiedenen Kanal unabhängig gewählt werden, um die Systemleistung zu optimieren. In diesen Arbeiten setzen die Amplitude gleichwertig.

$$A_R = A_G = A_B = A \tag{2.3}$$

Das differentielle Modulationsverfahren ordnet jede Sequenz von $\{-A, A\}$ zu $d = 0$ bzw. $\{A, -A\}$ zu $d = 1$ zu. Modulierte Datensymbole und verarbeitete Videoamplituden werden addiert, um die Anzeigeeingabe $g(i, j, k)$ zu liefern:

$$\begin{aligned}
 s_{R/G/B}(i, j, k) &= s_{R/G/B}(i, j, m) + A_{R/G/B} \cdot (2 \cdot d(i, j, m) - 1) \\
 s_{R/G/B}(i, j, k+1) &= s_{R/G/B}(i, j, m) - A_{R/G/B} \cdot (2 \cdot d(i, j, m) - 1)
 \end{aligned} \tag{2.4}$$

Ein Beispiel einer modulierten Bildfolge ist in Abbildung 2.4 gezeigt. Das Hinzufügen der modulierten Daten (hier mit $A = 4$) zu dem Videoeingang ergibt die Anzeigearamplituden in der rechten Spalte.



Abbildung 2.4: Ein Beispiel einer modulierten Bildfolge.

Im Vergleich zu Luminanzteil Y die Anzeigequalität in U und V Komponente ist signifikant besser, wenn Informationen in Chrominanz wiederholt und moduliert werden. Auf diese Weise wird die Gesamtleuchtdichte eines Pixel-Triple durch die Datenmodulation nicht beeinflusst. Die Umwandlungsmatrix, angegeben von test (ITU-R BT.709) für **high definition television** (HDTV) Display von Standard (R, G, B) zur Standard (Y, U, V) läuft:

$$T = \begin{pmatrix} 0,213 & 0,715 & 0,072 \\ -0,115 & -0,385 & 0,5 \\ 0,5 & -0,454 & -0,0458 \end{pmatrix} \quad (2.5)$$

Das Videosignal $s(i, j, k)$ muss vor dem Anwenden der Modulation in Y-, U- und V-Komponenten umgewandelt werden. Die nachfolgende inverse Konvertierung erklärt das Display-Eingangssignal in Abbildung 2.3:

$$\begin{aligned} \begin{pmatrix} g_R(i, j, k) \\ g_G(i, j, k) \\ g_B(i, j, k) \end{pmatrix} &= T^{-1} \cdot \left(T \cdot \begin{pmatrix} S_R(i, j, m) \\ S_G(i, j, m) \\ S_B(i, j, m) \end{pmatrix} + \begin{pmatrix} 0 \\ A_U \cdot d(i, j, m) \\ A_V \cdot d(i, j, m) \end{pmatrix} \right) \\ \begin{pmatrix} g_R(i, j, k+1) \\ g_G(i, j, k+1) \\ g_B(i, j, k+1) \end{pmatrix} &= T^{-1} \cdot \left(T \cdot \begin{pmatrix} S_R(i, j, m) \\ S_G(i, j, m) \\ S_B(i, j, m) \end{pmatrix} - \begin{pmatrix} 0 \\ A_U \cdot d(i, j, m) \\ A_V \cdot d(i, j, m) \end{pmatrix} \right) \end{aligned} \quad (2.6)$$

Diese Art der Modulation kann als eine Modulation des roten und des blauen Subpixels betrachtet werden, während das grüne Subpixel verwendet wird, um die Änderung der Luminanz des Pixel-Tripels zu kompensieren. Durch Definition korreliert A_U mit A_B und A_V mit A_R .

$$\begin{pmatrix} A_R \\ A_G \\ A_B \end{pmatrix} = T^{-1} \cdot \begin{pmatrix} A_Y \\ A_U \\ A_V \end{pmatrix} \quad (2.7)$$

2.2.2 DatenBlock

Um die Anforderungen an die Kameraauflösung zu lockern, Ein einfaches und unkompliziertes Verfahren ist Zuordnung jedes Datenbits zu einem Block von $B_X \times B_Y$ Pixeln.

$$\begin{aligned} d(l) &\rightarrow d(x, y, k) \quad 0 \leq l < L \\ L &= \lfloor N_X/B_X \rfloor \cdot \lfloor N_Y/B_Y \rfloor \\ x &= (l \cdot B_X) \bmod N_X + r_X, \quad r_X = 0 \dots (B_X - 1) \\ y &= \lfloor l/\lfloor N_X/B_X \rfloor \rfloor \cdot B_Y + r_Y, \quad r_Y = 0 \dots (B_Y - 1) \end{aligned} \quad (2.8)$$

Wenn die Anzahl der Pixel pro Zeile kein Vielfaches von B_X ist oder wenn die Anzahl der Pixel kein Vielfaches von B_Y ist, muss die Anzahl der Pixel und Zeilen, die für die Modulation in Gleichung (2.5) verwendet werden, ersetzt werden durch:

$$\begin{aligned} N_X &= \lfloor N_X/B_X \rfloor \cdot B_X \\ N_Y &= \lfloor N_Y/B_Y \rfloor \cdot B_Y \end{aligned} \quad (2.9)$$

In dieser Arbeit werden das DatenBlock für quadratische Blöcke gesetzt.

$$B_X = B_Y = B. \quad (2.10)$$

2.3 Anwendungsgebiete

Die Datenübertragungsrate des DaViD-Systems wird voraussichtlich erreicht bis zu 100 Mbit/s. Es gehöre zu einer Sichtlinienübertragung für kurze Verbindungen. Geeignete Abdeckungsbereichen hängen von der Größe des Displays und der Kameraoptik ab. Obwohl im Vergleich zum letzten WLAN- Versionen Institute of Electrical and Electronics Engineers (IEEE) 802.11,

die Leistung scheint nicht so attraktiv. Der Vorteil liegt nicht nur in der wachsenden Leistungsfähigkeit von Video-Display und Kamera, aber auch die Option zur Wiederverwendung der bestehenden Hardware, die zum Zeigen des Videos installiert wurde. Ein empfohlene praktische Anwendungsbereich des DaViDs ist öffentlicher Ort, z.B. U-Bahn-Station, großes Stadion und so weiter. Annehmen eine Situation, wenn die Leute auf ihre U-Bahn warten, sie können ihre eigene Software aktualisieren, indem Sie einfach auf die zeigende Werbung in dem Bildschirm leiten.

Berücksichtigen der Eigenschaften des DaViDs, d.h. die Synchronisation von Videospielen und Datenübertragung. Viele Anwendungsszenarien können in Betracht gezogen werden und scheinen sehr attraktiv zu sein. Die drei Hauptszenarien sind:

- Indoor-individuelle Kommunikation: Kurzstreckenverbindungen basieren auf relativ kleinen (Tablet-Größe) Bildschirm, Anwendungen z.B. die Übertragung von Hintergrundinformationen an Besucher im Museum, Kiosk.
- Indoor-Multicast-Kommunikation: Streckabstand ist länger als ersten Fall auf relativ größer (40-100") Bildschirm, Anwendungen z.B. während Videoabspielen Besucher die Anwendungsdaten oder Mediendateien herunterladen können im Kiosk, Restaurant.
- Freien Kommunikation: Größter Bildschirm wie im Einkaufszentren oder Sport-Arenen, Anwendungen können denen des zweiten Szenarios ähneln.

Sobald die Dienste auf öffentlichen Bildschirmen implementiert werden, kann Leute mit Hilfe eines modernen Smartphones, die mit einer geeigneten Kamera eingebaut ist, nach der Installation einer neuen App innovative wahrnehmen.

3 ErsteErfahrung

In diesem Kapitel werden die erste Erfahrung beschrieben. Zuerst läuft die Vorstellung des DaVid-Systems. Die Systemmodell und Arbeitsprinzip des Systems werden in anschließenden Abschnitt erläutert. Schließlich folgt die mögliche Anwendungsgebiete des Systems.

Nicht vergessen, dass Überschriften nicht aufeinander folgen dürfen...

3.1 TexLipse spell checking

To enable spell checking in TeXLipse, download the respective dictionaries from <https://sourceforge.net/projects/texlipse/files/dictionaries/>.

Save the dictionaries at a local location and enter the path in Window->Preferences->Texlipse->Spell Checker (see Fig. 3.1).

To synchronize the user dictionaries between multiple machines, it might be useful to save the dictionaries in your google drive or drop box.

3.2 Enable tikzexternalize for PdfLatex

Go to Window->Preferences->Texlipse->Builder Settings and add

--shell-escape

to the command arguments (see Fig. 3.2).

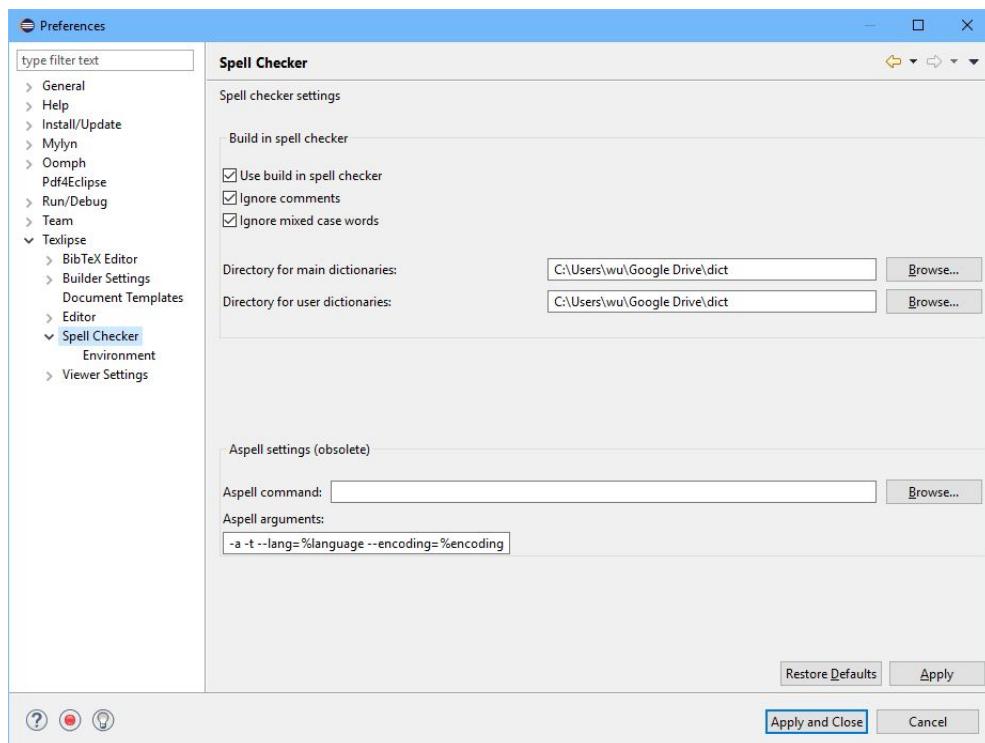


Figure 3.1: TeXlipse Spell Checker preferences.

3.3 Forward search with TeXlipse and Sumatra PDF

Download and install SumatraPDF: <https://www.sumatrapdfreader.org/>.

Then edit the viewer settings for SumatraPDF in Window->Preferences->Texlipse->Viewer Settings.

Change the viewer arguments to

```
-reuse-instance %fullfile -forward-search %texfile %line
```

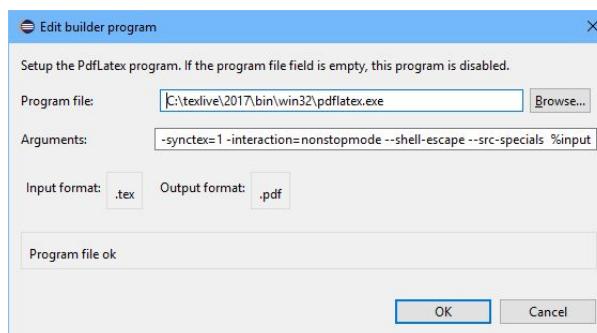


Figure 3.2: PdfLatex Builder Settings.

and leave all DDE message field empty. Change the inverse search support to "Viewer runs external command" and enable "Viewer supports forward search".

Figure 3.3 displays the dialog window.

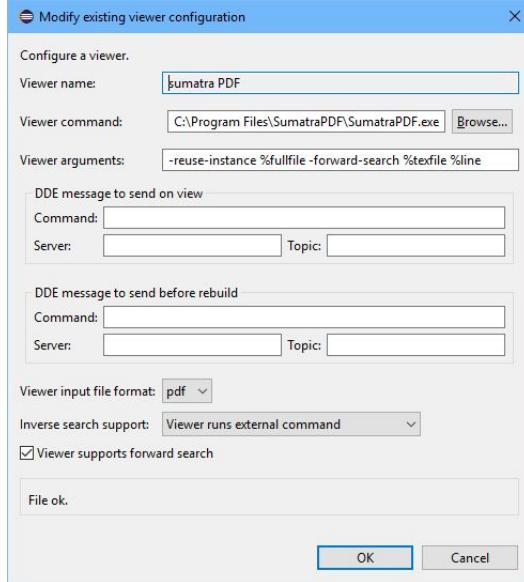


Figure 3.3: TeXlipse Viewer Settings.

In SumatraPDF configure the inverse search command via the `Settings->Options` menu (see Fig. 3.4).

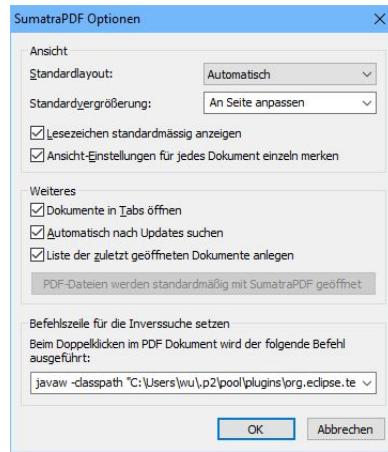


Figure 3.4: SumatraPDF Options.

If you have install TeXlipse 1.5.0, the inverse search command will look like this:

```
javaw -classpath "C:\Users\wu\.p2\pool\plugins\net.sourceforge.texlipse_1.5.0\
texlipse.jar" net.sourceforge.texlipse.viewer.util.FileLocationClient -p
55000 -f "%f" -l %l
```

Let the path point to your eclipse share pool. Or if you do not have a shared pool, choose the plugins directory of your eclipse installation.

For TeXLipse 2.0.X the FileLocationClient is relocated to org.eclipse.texlipse making the inverse search command look like the following.

```
javaw -classpath "C:\Users\wu\.p2\pool\plugins\org.eclipse.texlipse_2
.0.1.201801202105\texlipse.jar" org.eclipse.texlipse.viewer.util.
FileLocationClient -p 55000 -f "%f" -l %l
```

4 ZweiteErfahrung

In diesem Kapitel wird die Realisierung der zweiten Verfahren eingegangen werden. Die Bildregistrierung wird zuerst im Detail gegeben. Anschließend läuft die Differenzbild Optimierung und Bildverarbeitung. QR Muster Detektion und die entspricht Matlab Code werden schließlich erläutert. Zur Implementierung dieser Verfahren wird Matlab unter der Lizenz TU Dortmund verwendet.

4.1 Allgemeine Struktur

Diese Verfahren verwendet die Charakteristiken der Datenmodulation des DaViD Systems, d.h. an jedem Eck des Datenebene ein QR Muster hinzugefügt und dann mit den Daten zusammen hinter dem Bild moduliert. Am Empfängen durch eine Reihe von Verarbeitungen wird QR Muster detektiert und dann schließlich bestimmt der Modulationsbereich. Diese Methode löst effektiv den unschönen Effekt, der das QR-Modul direkt zum Bild hinzugefügt wird, und kann das Problem schnell und effektiv durch die QR-Mustereigenschaft lösen. Abbildung 4.1 zeigt die Strukturdiagramm des Verfahrens.

In dieser Verfahren läuft zuerst Bildregistrierung. Durch diese Schritt kann die Punkte über den zwei Bildern in dasselbe Koordinatensystem konvertieren. Anschließend mit Hilfe einer Optimierung für die enthältet Differenzbilder wird eine detektierend Bild bekommen, die die Erkennung des QR Musters erleichtern können. Schließlich wird es durch die Charakteristik des QR Musters, das Breiteverhältnis $1 : 1 : 3 : 1 : 1$ beträgt, um QR Muster zu detektieren.

4.2 Bildregistrierung

Annehmen eine Szene, wenn eine Rhein Fotos mit einer Handheld-Kamera aufnehmen, kommt es aufgrund von Handbewegungen zu einer leichten Verschiebung zwischen den beiden benachbarten Fotos. Wenn Sie diese Fotos abziehen, um Differenzbild zu erhalten, wird der Ergebnisse sehr schlecht sein. Um diese Problem zu lösen, wird hier Bildregistrierung eingeführt. Ein Flussdiagramm der Bildregistrierung wird in Abbildung 4.2 gezeigt.

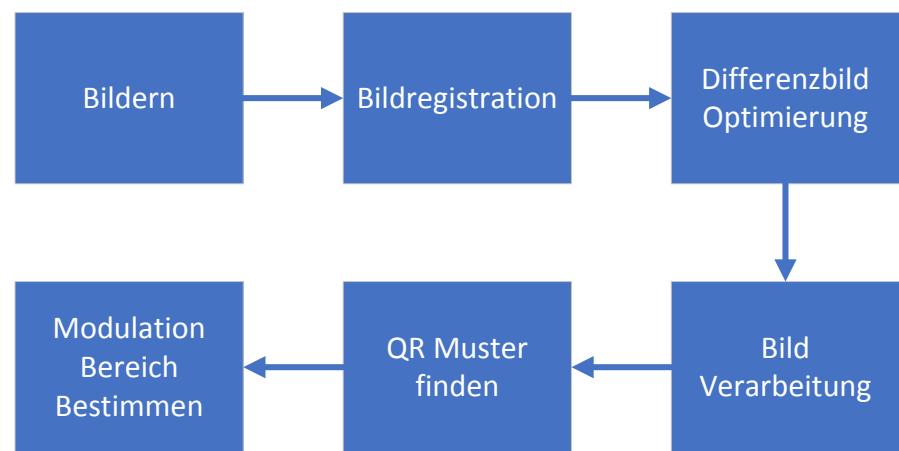


Abbildung 4.1: Strukturdiagramm.

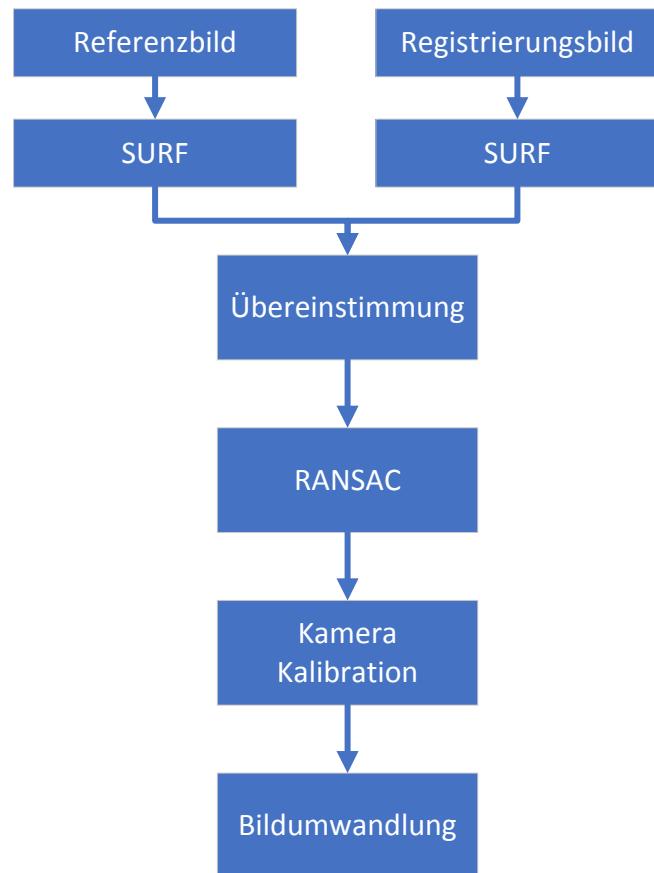


Abbildung 4.2: Flussdiagramm der Bildregistrierung.

4.2.1 SURF

In Bildregistrierung wird zuerste Merkmalserkennung eingegangen. Hier in dieser Arbeit lass die SURF [6] nutzen. Es ist eine verbesserte Version von SIFT, die Haar-Wavelet verwendet, um die Gradientenoperation in der SIFT-Methode anzunähern und gleichzeitig verwendet eine Integralgraph-Technik für schnelle Berechnungen. Die Faltung bezieht sich nur auf das vorherige Bild, mit Erhöhung der Größe des Bildkerns können das Heruntertaktung-Verfahren realisiert werden. Die Geschwindigkeit von SURF ist 3-7 mal die von SIFT mit der in den meisten Fällen entspricht Leistung von SIFT. Daher wurde es in vielen Anwendungen eingesetzt, insbesondere in Anwendungen, in denen die Laufzeitanforderungen hoch sind. Der Verlauf einer SURF Merkmalserkennung ist wie folgend:

- **Aufbau einer hessischen Matrix.**

Die Hesse-Matrix stellt den Kern des SURF Algorithmus dar. Zur Vereinfachung der Operation wird die Funktion $f(z, y)$ angenommen, dass die Hesse-Matrix H setzt sich aus Funktionen und partiellen Ableitungen zusammen:

$$H(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \cdot \partial y} \\ \frac{\partial^2 f}{\partial x \cdot \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \quad (4.1)$$

Diskriminante der H-Matrix läuft:

$$\det(H) = \frac{\partial^2 f}{\partial x^2} \cdot \frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \cdot \partial y} \right)^2 \quad (4.2)$$

Der Wert der Diskriminante ist der Eigenwert der H-Matrix. Durch dessen positiven und negativen wird bestimmt, ob der Punkt ein Extrempunkt ist oder nicht. Im SURF Algorithmus wird das Bildpixel $I(x, y)$ anstelle des Funktionswertes $f(x, y)$ verwendet. Nutzen eine Zweite-Order Gaussian Function als Filter. Die zweiten Partielle Ableitungen können durch Faltung zwischen bestimmten Kernen berechnet werden. Dadurch können die Werte der drei Matrixelemente der H-Matrix auch berechnet werden, nämlich die H-Matrix berechnet:

$$\begin{aligned} H(\mathbf{x}, \sigma) &= \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix} \\ L(\mathbf{x}, \sigma) &= G(\sigma) * I(\mathbf{x}) \\ G(\sigma) &= \frac{\partial^2 g(\sigma)}{\partial x^2} \end{aligned} \quad (4.3)$$

Hier $L_{xx}(\mathbf{x}, \sigma)$ bedeutet die Faltung der zweiter Gaussian Ableitung $G(\sigma)$ mit dem Bild I in Punkt $\mathbf{x}(x,y)$, ähnlich für $L_{xy}(\mathbf{x}, \sigma)$ und $L_{yy}(\mathbf{x}, \sigma)$. Auf diese Weise kann der Wert der Determinante für jedes Pixel in dem Bild berechnet werden, und dieser Wert kann verwendet werden, um den Merkmalspunkt zu feststellen. Zur einfacheren Anwendung schlägt Herbert Bay[6] vor, L mit einer Approximation ersetzen. Um den Fehler zwischen dem genauen Wert und der Approximation auszugleichen, kann die H-Matrix-Diskriminante wie folgt ausgedrückt werden:

$$\det(\mathcal{H}_{\text{Approx}}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad (4.4)$$

- **Erstellen Maßstab Raum**

Der Maßstabsraum $L(\mathbf{x}, \sigma)$ des Bildes ist die Darstellung dieses Bildes bei unterschiedlichen Auflösungen(Skalierung). Im Bereich der Computer Vision wird der Maßstabsraum symbolisch als Bildpyramide ausgedrückt, wobei die Eingangsbildfunktion wiederholt mit dem Kern der Gaußschen Funktion gefaltet und wiederholt unterabgetastet wird. Diese Methode wird hauptsächlich für die Implementierung des SIFT Algorithmus verwendet. Jede Bildschicht hängt jedoch von der vorherigen Bildschicht ab, und das Bild muss in der Größe angepasst werden. Daher hat diese Berechnungsmethode eine große Kosten in Berechnung. Im Vergleich dazu ist es in SURF durch die Erhöhung der Größe des Bildkerns. Diese ist ein Unterschied zwischen dem SIFT Algorithmus und dem SURF Algorithmus bei der Verwendung des Pyramidenprinzips. Der Algorithmus ermöglicht, dass mehrere Bilder des Maßstabsraums gleichzeitig verarbeitet werden, ohne dass das Bild unterabgetastet wird, wodurch die Leistung des Algorithmus verbessert wird. Das linke Bild in Abbildung 4.3 ist eine Pyramidenstruktur, die auf herkömmliche Weise erstellt wird, die Größe des Bildes wird geändert, und die Operation wird die Unterebene unter Verwendung der Gaußschen Funktion wiederholt glätten. Der Surf Algorithmus auf der rechten Seite in Abbildung 4.3 behält das ursprüngliche Bild unverändert und ändert nur die Filtergröße.

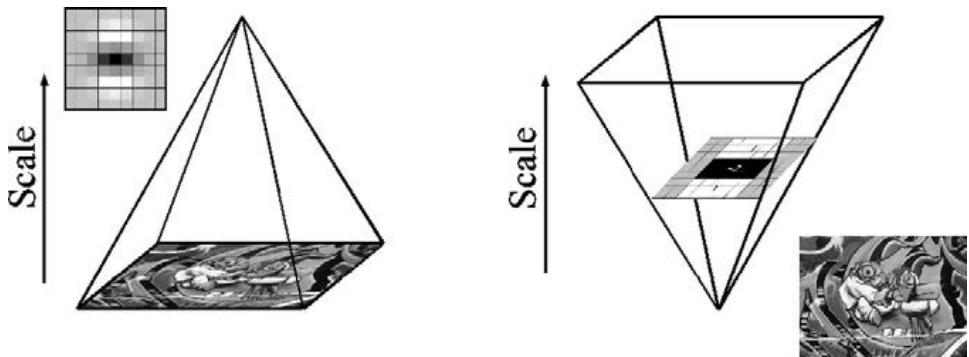


Abbildung 4.3: Scale space.

- **Präzise Lokalisierung von Feature-Punkten**

Vergleichen die Größe jedes Pixel, die von der hessischen Matrix verarbeitet wird, mit die 26 Punkten in seiner drei Dimensionen Raum, wie in Abbildung 4.4 zeignt. Wenn es das Maximum oder Minimum dieser 26 Punkte ist, wird es als vorläufiger Merkmalspunkt beibehalten. Das dreidimensionale lineare Interpolationsverfahren wird verwendet, um die Merkmalspunkte des Subpixel-Niveaus zu erhalten, und die Punkte, deren Werte kleiner als ein bestimmter Schwellenwert sind, werden ebenfalls entfernt.

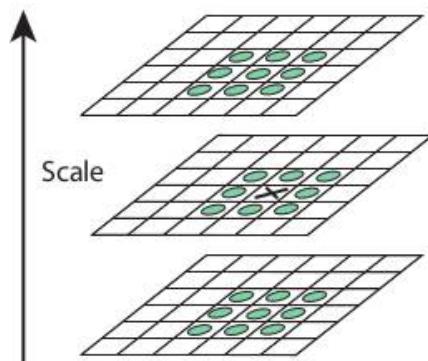


Abbildung 4.4: Extreme Wert Erkennung.

• Hauptrichtungsermittlung

SIFT wählt die Hauptrichtung des Merkmalspunkts unter Verwendung des Gradientenhistogramms im Merkmalspunktbereich aus. Die Richtung, in der der Bin-Wert des Histogramms der größte und oder 80% maximale Bin-Wert überschreitet, wird als Hauptrichtung des Merkmalspunkts genommen. Dagegen beim SURF wird das Gradientenhistogramm nicht statistiken, sondern das Harr-Wavelet-Eigenschaft im Merkmalspunktbereich wird statistisch analysiert. Das heißt, im Bereich der Merkmalspunkt (zum Beispiel innerhalb eines Kreises mit einem Radius von $6s$, wobei s der Maßstab ist, auf dem der Punkt liegt) die Summe der Horizontal-Haar-Wavelet-Merkmale und der Vertikal-Haar-Wavelet-Merkmale aller Punkte im 60-Grad-Sektor($\pi/3$) werden gezählt. Die Größe des Haar Wavelets stellt als $4s$, so dass für jeden Sektor einen Wert bekommt. Dann wird 60-Grad-Sektor in einem bestimmten Intervall gedreht, schließlich lassen die Richtung des Sektors mit Maximalwert als Hauptrichtung des Merkmalspunkts nehmen. Ein schematisches Diagramm des Prozesses ist wie folgt in Abbildung 4.5.

• Merkmalspunkt Descriptor Generierung

SURF nehmen eine quadratische Rahmen um den Merkmalspunkt. Die Seite der Rahmen ist $20s$ (s ist die Skala, bei der der Merkmalspunkt erkannt wird). Die Richtung des Rahmens ist natürlich die Hauptrichtung, die in vorliegenden Schritt erfasst wird. Die Rahmen wird dann in 16 Unterbereiche unterteilt, von denen jeder die Haar-Wavelet-Merkmale der horizontalen und vertikalen Richtungen von 25 Pixeln berechnen. Hier die horizontalen und vertikalen Richtungen sind relativ zur Hauptrichtung. Das Haar Wavelet-Merkmal ist die Summe der horizontalen

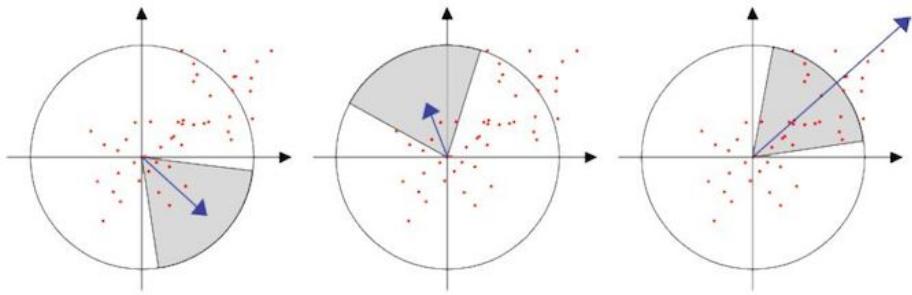


Abbildung 4.5: Dominante Orientierung Feststellen.

Richtungswerte, die Summe der absoluten Werte in der horizontalen Richtung, die Summe der vertikalen Richtungen und die Summe der absoluten Werte in der vertikalen Richtung. Das schematische Diagramm in Abbildung 4.6 zeigt dieses Prozesse.

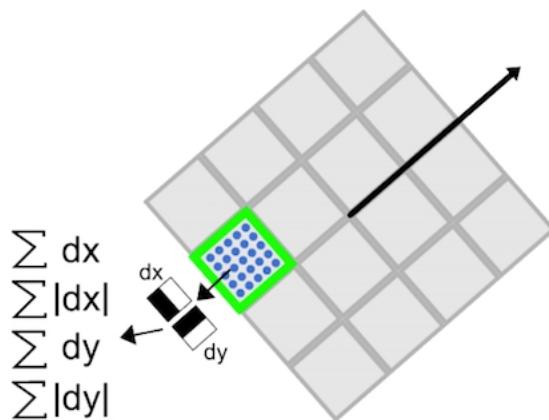


Abbildung 4.6: Merkmalspunkt Deskriptor.

Auf diese Weise hat jeder kleine Bereich 4 Werte, so dass jeder Merkmalspunkt ein $16 \times 4 = 64$ dimensionaler Vektor verfügt, der halb so klein wie Sift(128 Dimension) ist, deswegen den Anpassungsprozess beim Merkmalanpassungsprozess stark beschleunigt. Die folgende Abbildung 4.7 zeigt den Merkmalspunkt, den wir durch den SURF-Algorithmus erhalten haben.

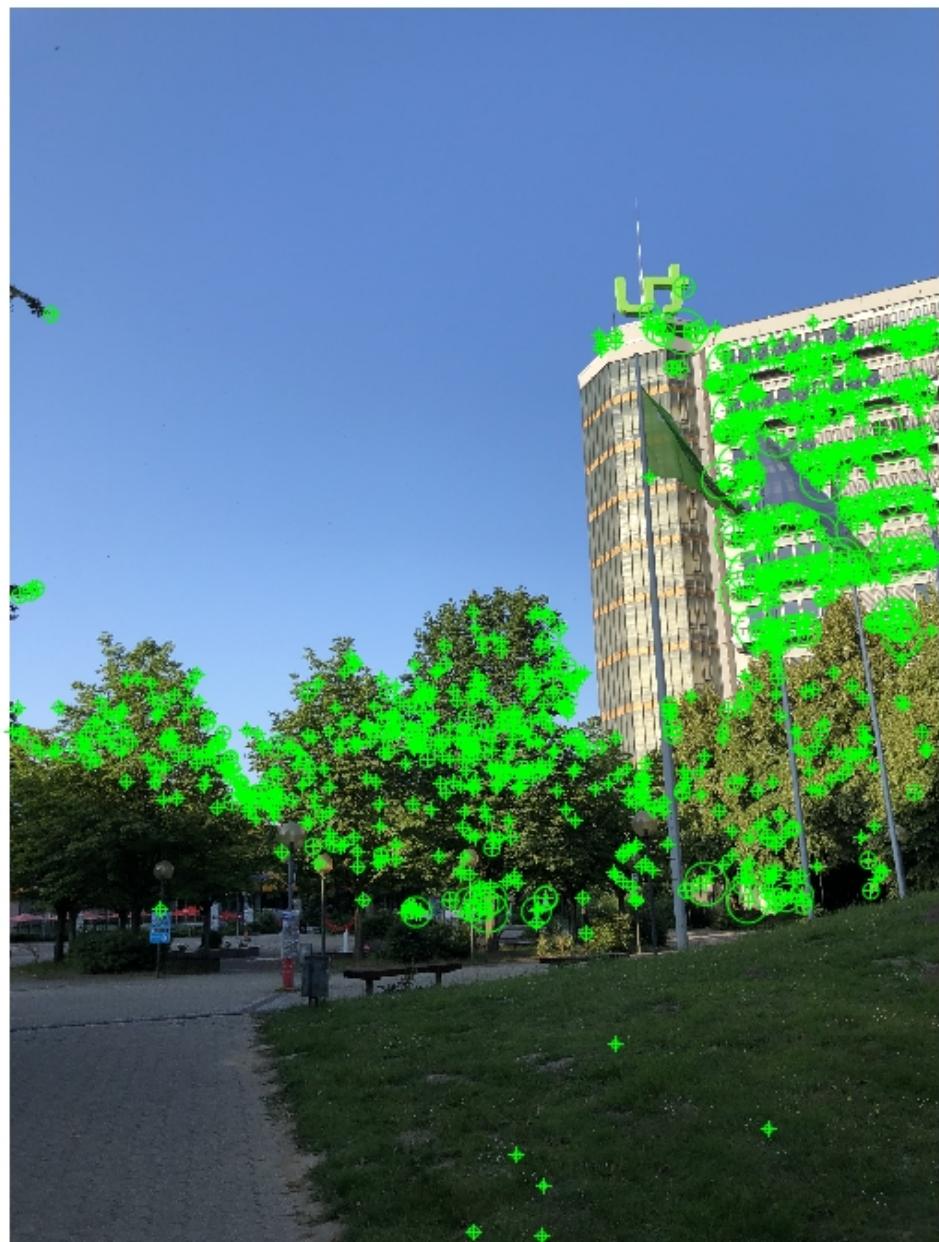


Abbildung 4.7: SURF Merkmal.

4.2.2 RANSAC

Nach SURF Merkmalserkennung wird die Merkmalspunkte von zwei benachbarten Bildern mit Verfahren z.B. **Normalized Cross Correlation** (NCC) übereinstimmt. Leider durch diese Verfahren gibt es noch viele fehlerhafte zusammenpassendes Paar. Deswegen wird **RANdom SAmple Consensus** (RANSAC) eingeführt.

RANSAC Algorithmus, der von Fischler und Bolles [7] vorgeschlagene im Jahr 1981, ist ein allgemeiner Parameterschätzungsansatz, um den großen Anteil von Ausreißern in den Eingabedaten zu bewältigen. Im Gegensatz zu vielen der üblichen robusten Schätzverfahren wie M-Schätzer und kleinsten Quadraten, die von der Computer Vision Community aus der Statistik-Literatur übernommen wurden, wurde RANSAC aus der Computer-Vision-Community entwickelt.

Ein einfaches Beispiel ist in der Abbildung 4.7 dargestellt. Das Ziel besteht darin, die am besten geeignete Linie unter einer Menge von Datenpunkten zu finden. Wenn es die einfache Methode der kleinsten Quadrate verwenden, um diese Linie zu finden, wie auf der linken Seite gezeigt, kann es leider nicht richtig finden, da die Methode der kleinsten Quadrate von allen Datenpunkten beeinflusst wird. Dagegen mit RANSAC kann das Modell nur von den inlierer Punkten berechnet werden und die Ergebnisse wie auf der rechten Seite zeigt.

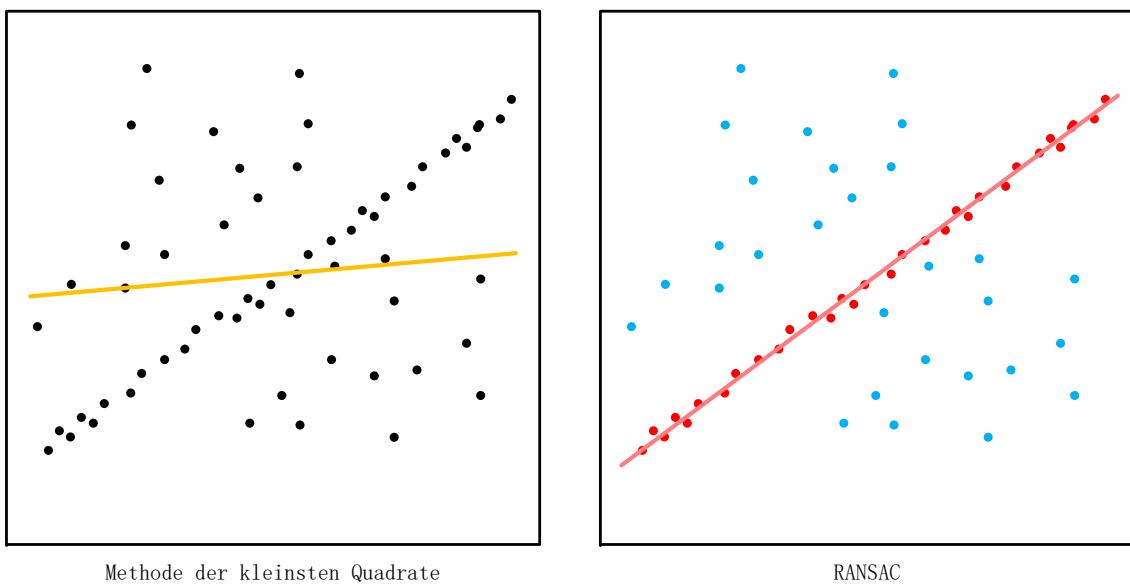


Abbildung 4.8: Linien Detektion.

RANSAC ist ein Wiederholungsprobennahme Verfahren, das durch die minimalen Anzahl von Beobachtungspunkten (Datenpunkten) die Kandidatenlösungen generiert. Diese Datenpunkten sind die erforderlich, um die zugrunde liegenden Modellparameter zu schätzen. Darauf haben Fischler und Bolles [7] hingewiesen, zur Erhalten einer anfängliche Lösung und Beschneidung der Ausreißern RANSAC Verfahren braucht nicht so viele Daten, sondern verwendet die kleinste mögliche Menge und fährt fort, diese Menge mit eine konsistenten Datenpunkten zu vergrößern.

Der grundlegende Algorithmus ist wie folgt zusammengefasst:

- Zufällig wählen die Mindestanzahl der Punkten aus, die erforderlich sind, zum Bestimmen der Modellparameter.
- Lösen die Parameter des Modells.
- Bestimmen wie viele Punkte aus der Menge aller Punkte mit einer vordefinierten Toleranz ϵ übereinstimmen
- Wenn der Bruchteil der Anzahl von Inlieren über die Gesamtzahl der Punkte in dem Satz einen vordefinierten Schwellenwert τ überschreitet, schätzen die Modellparameter mit allen identifizierten Inlieren und terminieren wieder.
- Ansonsten wiederholen die Schritte 1 bis 4 (maximal N-mal).

N bedeutet die Anzahl der Iterationen. Es wird hoch genug gewählt, um die Wahrscheinlichkeit p (normalerweise auf 0,99 gesetzt) sicherzustellen, dass mindestens eine der Gruppen von Stichproben keinen Ausreißer enthält. Dann die Wahrscheinlichkeit, dass bei N Mal Iterationen mit erforderlich minimalen Anzahl Punkte (hier m annahmen) mindestens ein Ausreißer mit ausgewählt wird, läuft:

$$1 - p = (1 - u^m)^N \quad (4.5)$$

Hier u stellen die Wahrscheinlichkeit dar, dass jeder ausgewählte Datenpunkt ein Inlier ist. Dagegen $v = 1 - u$ heißt die Wahrscheinlichkeit, dass jeder ausgewählte Datenpunkt ein Ausreißer ist. Durch einige Gleichheitsumwandlung können die Anzahl der Iterationen ausgedrückt werden als:

$$N = \frac{\log(1 - p)}{\log(1 - (1 - v)^m)} \quad (4.6)$$

Abbildung 4.8 zeigt die passende Punkt durch SURF Merkmalserkennung. Es ist ersichtlich, dass es viele fehlerhafte Kombinationen gibt. Durch die Anwendung von RANSAC kann dieses Problem effektiv lösen und die übereinstimmenden Punkte verfeinern, wie in Abbildung 4.9 zeigt.

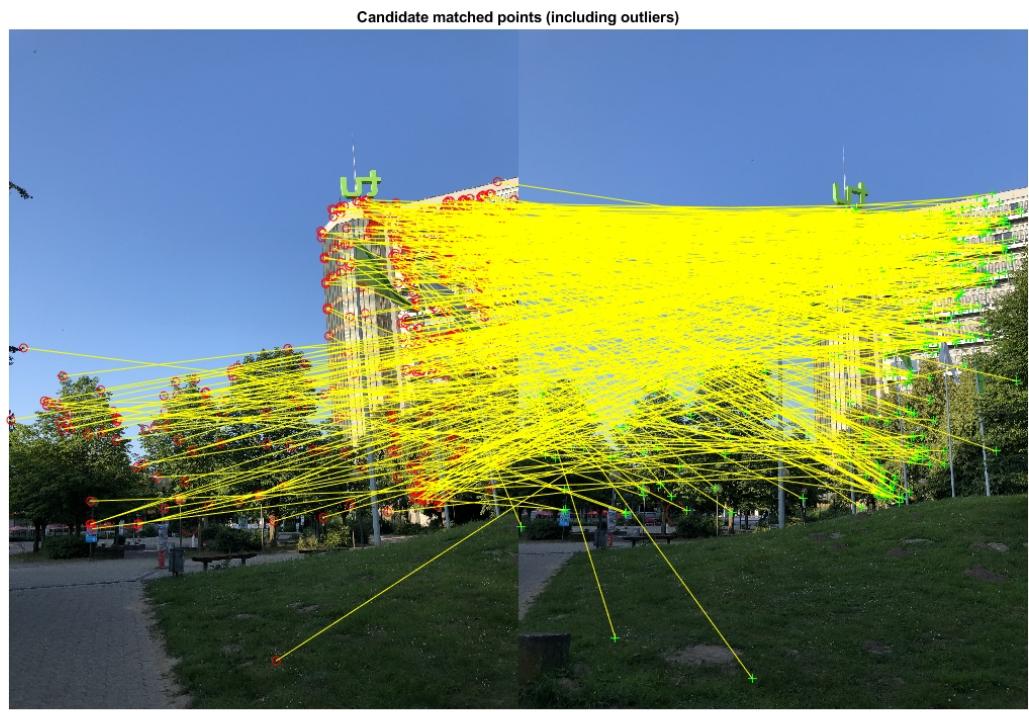


Abbildung 4.9: Ohne RANSAC.

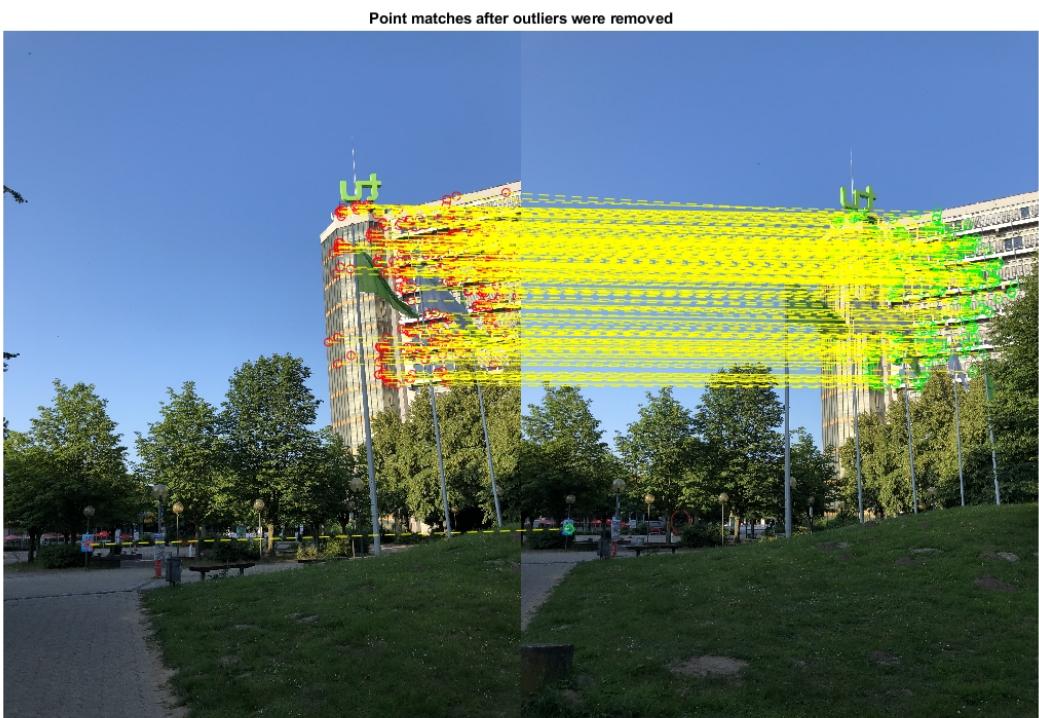


Abbildung 4.10: Mit RANSAC.

4.2.3 Kamerakalibrierung

Wie in den ersten beiden Abschnitten vorgestellt, finde die übereinstimmende Punkte mit Verwenden des SURF in aufeinanderfolgenden Bildern, dann durch RANSAC lassen die Ausreißer verwerfen. Als nächstes soll die Kamera kalibriert werden, um die Punkte über den zwei Bildern in dasselbe Koordinatensystem konvertieren. In diesem Abschnitt wird zuerst das Kameramodell vorstellen.

Kamera Model

Das Modell der Lochkamera ist in Abbildung 4.10 dargestellt. In dem Modell ist C das optische Zentrum (Fokus), f ist die Kamerabrennweite.

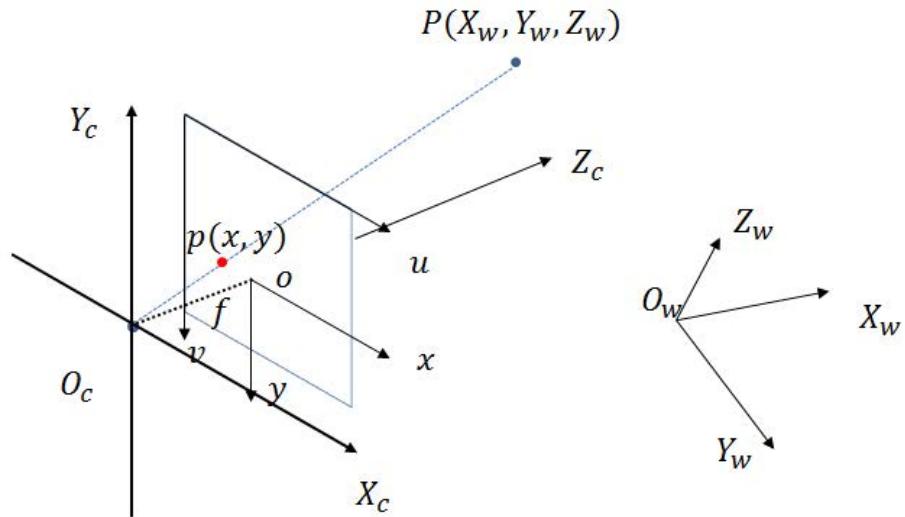


Abbildung 4.11: Kamera Model.

Die vier Koordinatensysteme im Modell sind wie folgt definiert:

- 3D Weltkoordinatensystem $W(X, Y, Z)$
Punktkoordinaten werden durch homogene Koordinaten dargestellt: $\tilde{X}_w \sim (X_w, Y_w, Z_w, 1)^T$
- 3D Kamerakoordinatensystem $C(X_c, Y_c, Z_c)$
Punktkoordinaten werden durch homogene Koordinaten dargestellt: $\tilde{X}_c \sim (X_c, Y_c, Z_c, 1)^T$
- 2D Bildabbildung Koordinatensystem $P(x, y)$
Punktkoordinaten werden durch homogene Koordinaten dargestellt: $\tilde{x} \sim (x, y, 1)^T$

- 2D Bildpixel Koordinatensystem $I(u, v)$

Punktkoordinaten werden durch homogene Koordinaten dargestellt: $\tilde{u} \sim (u, v, 1)^T$

Unter diesem Modell wird ein 3D-Punkt im Weltkoordinatensystem durch drei Koordinaten den 2D-Bildpixelkoordinaten zugeordnet.

(a). 3D-Weltkoordinatensystem zum 3D-Kamera-Koordinatensystem.

Die Transformation vom Weltkoordinatensystem zum Kamerakoordinatensystem ist eine Starrekörpertransformation, d.h. das Objekt verformt sich nicht und nur durch Rotation und Parallelverschiebung. Diese Transformation wird in Abbildung 4.11 gezeigt. R bedeutet Rotationsmatrix und T ist Translationsmatrix.

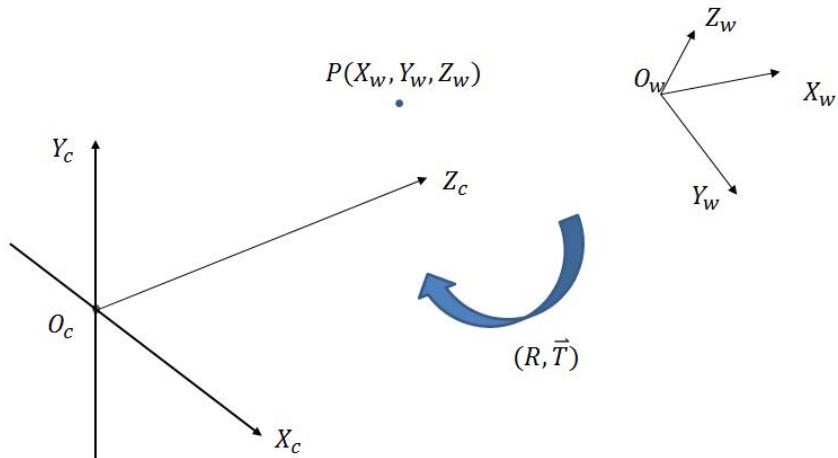


Abbildung 4.12: Transformation vom Weltkoordinatensystem zum Kamerakoordinatensystem.

Um die entsprechende Rotationsmatrix zu erhalten, wird verschiedene Winkel um verschiedene Koordinatenachsen gedreht. Ein simple Beispiel wird in Abbildung 4.12 gezeigt.

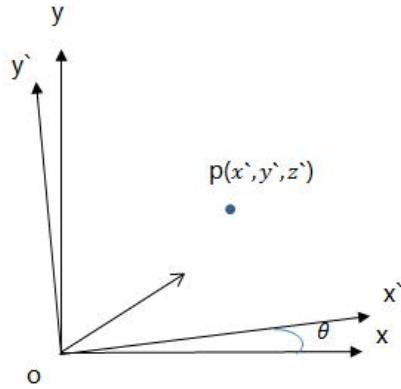


Abbildung 4.13: Rotation um Z-Achse.

Aus dem Bild können wir leicht bekommen:

$$\begin{cases} x = x' \cos \theta - y' \sin \theta \\ y = x' \sin \theta + y' \cos \theta \\ z = z' \end{cases} \quad (4.7)$$

In Matrixform wie folgend ausgedrückt:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_1 \cdot \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (4.8)$$

In ähnlicher Weise, um die x-Achse, y-Achse dreht sich um φ und ω Grad, bekommen:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_2 \cdot \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (4.9)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos \omega & 0 & \sin \omega \\ 0 & 1 & 0 \\ -\sin \omega & 0 & \cos \omega \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_3 \cdot \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (4.10)$$

Dann können die Rotationsmatrix erhalten werden:

$$R = R_1 \cdot R_2 \cdot R_3 \quad (4.11)$$

Kombinieren das Obige, können die Koordinaten von Punkt P im Kamerakoordinatensystem erhalten.

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = R \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \quad (4.12)$$

Im homogenen Koordinatensystem darstellt:

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ \vec{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (4.13)$$

(b). 3D-Kamera-Koordinatensystem zum 2D-Bildabbildung Koordinatensystem.

Die Transformation vom Kamerakoordinatensystem zum Bildkoordinatensystem gehört zur perspektivischen Projektionsbeziehung von 3D zu 2D, wie zeigt in Abbildung 4.13.

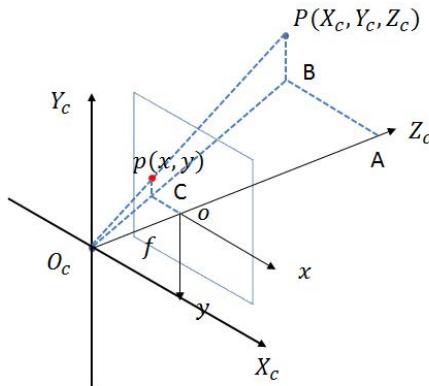


Abbildung 4.14: Transformation vom Kamerakoordinatensystem zum Bild Koordinatensystem.

Es gibt zwei Paare ähnlichen Dreiecken:

$$\begin{aligned} \triangle ABC &\sim \triangle oCO_c \\ \triangle PBC &\sim \triangle pCo_c \end{aligned} \quad (4.14)$$

Aus ähnlichen Dreiecksbeziehungen können diese Gleichung Verfügbar sein:

$$\frac{AB}{oC} = \frac{AO_c}{oO_c} = \frac{PB}{pC} = \frac{X_c}{x} = \frac{Z_c}{f} = \frac{Y_c}{y} \quad (4.15)$$

Durch die Gleichung Transformation können es erhalten:

$$x = f \cdot \frac{X_C}{Z_C}, y = f \cdot \frac{Y_C}{Z_C} \quad (4.16)$$

Im homogenen Koordinatensystem darstellt:

$$Z_C \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} \quad (4.17)$$

Zu dieser Zeit ist die Einheit des Projektionspunkts p noch nicht Pixel, sondern mm und muss weiter in das Pixelkoordinatensystem umgewandelt werden.

(c). 2D-Bildabbildung Koordinatensystem zum 2D-Bildpixel Koordinatensystem.

Das Pixelkoordinatensystem und das Bildkoordinatensystem befinden sich alle auf der Abbildungsebene, aber die jeweiligen Ursprünge und Maßeinheiten sind unterschiedlich. Der Ursprung des Bildkoordinatensystems ist der Schnittpunkt der optischen Achse der Kamera und der Abbildungsebene, üblicherweise der Mittelpunkt der Abbildungsebene oder der Hauptpunkt. Die Einheit des Bildkoordinatensystems ist mm, die zu der physikalischen Einheit gehört, und die Einheit des Pixelkoordinatensystems ist Pixel. Wir beschreiben gewöhnlich, dass ein Pixel welche Zeilen und Spalten ist. So ist der Übergang zwischen den beiden wie folgt in Abbildung 4.14.

Hier dx , dy ist die Größe jedes Pixels in den X- und Y-Achsenrichtungen. Jedes Pixel des Bildes hat die folgende Beziehung 4.7 in zwei Koordinatensystemen.

$$\begin{cases} u = \frac{x}{d_x} + u_0 \\ v = \frac{y}{d_y} + v_0 \end{cases} \quad (4.18)$$

Im homogenen Koordinatensystem darstellt:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{d_x} & 0 & u_0 \\ 0 & \frac{1}{d_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4.19)$$

Das Bildkoordinatensystem ist eine zweidimensionale Ebene(Bildebene). Es ist in praktisch die Oberfläche des Kamera-CCD-Sensors. Jeder CCD-Sensor hat eine bestimmte Größe und

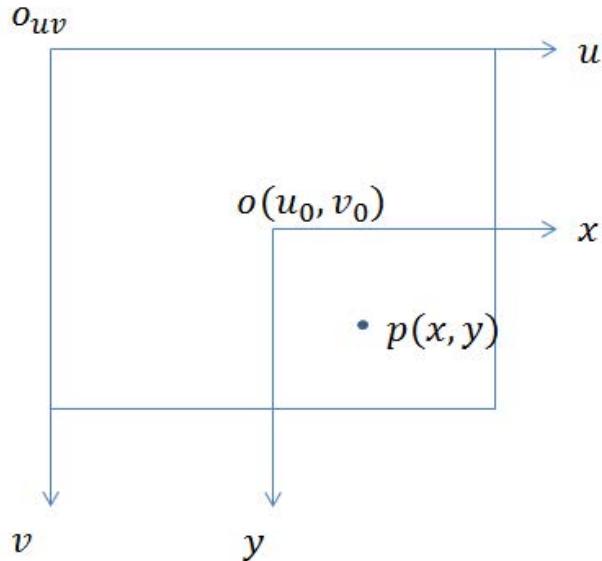


Abbildung 4.15: Konvertierung von Bildkoordinatensystem zu Pixelkoordinatensystem.

eine bestimmte Auflösung. Diese beide bestimmt die Konvertierungsbeziehung. Eine simple Beispiel, eine Größe des CCD-Sensors ist $8 \text{ mm} \times 6 \text{ mm}$, die Auflösung dafür ist 640×480 , dann die Beziehung zwischen mm und Pixel läuft 80 pixel/mm . Lassen Sie die physikalische Größe jedes Pixels des CCD-Sensors $d_x \times d_y$ sein, entspricht läuft $d_x = d_y = \frac{1}{80} \text{ mm}$.

Dann durch die Umwandlung der obigen vier Koordinatensysteme kann ein Punkt vom Weltkoordinatensystem zum Pixelkoordinatensystem erhalten.

$$\begin{aligned}
 Z_C \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} \frac{1}{d_x} & 0 & u_0 \\ 0 & \frac{1}{d_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R & t \\ \vec{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R & t \\ \vec{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}
 \end{aligned} \tag{4.20}$$

Die erste Matrix der rechten Gleichung ist die allgemein bekannt interne Referenz der Kamera. Dagegen ist die zweite Matrix die externe Referenz der Kamera. Beide Parameter der Kamera durch Zhang Zhengyou [8] Kalibrierung erhalten werden. Einige typisch Kamera Parameter von Manufaktur liegt in Tabellen 4.1.

Tabelle 4.1: Parameter des Kameras im Vergleich.

Parameter	Google Pixel	Google Pixel2	Iphone 10
Sensor Größe "	1/2.3	1/2.6	1/3
Bild Auflösung pixels	4048 × 3036	4032 × 3024	4032 × 3024
Pixel Größe μm	1.544	1.4	1.22
Brennweite mm	4.67	4.47	3.99
Formatfaktor 35mm	5.55	6.04	7.02

Aus der obigen Formel wenn die internen und externen Parameter der Kamera bekannt sind, ist nämlich die Projektionsmatrix bekannt, und zu diesem Zeitpunkt können die entsprechenden Bildkoordinaten für jeden beliebigen räumlichen Punkt erhalten werden. Dagegen wenn die Position m (u, v) in Bildkoordinate bekannt ist, und auch die Parameter innerhalb und außerhalb der Kamera bereits bekannt sind, kann die entsprechenden Punkt in Wert Koordinate nicht eindeutig bestimmt werden. Die Grund dafür ist, die Z_c Information während des Projektionsprozesses eliminiert wird.

Image Warping

In diese Arbeit wird zuerst eine vereinfachte Situation betrachten, d.h. nur mit Rotations Einfluss. Abbildung 4.15 zeigt das 3D-Rotationsbewegungsmodell der Kamera. Die Position des optischen Zentrums ändert sich während der Kamerabewegung im Drehbewegungsmodell der Kamera nicht. Unter diesem Modell ist die Abbildungsbeziehung zwischen dem Punkt X im Weltkoordinatensystem und der Bildkoordinate x im homogenen Koordinatensystem dargestellt:

$$x = KRX, X = \lambda K^{-1}x \quad (4.21)$$

Wobei: K ist der interne Parameter der Kamera, wie zuvor definiert. λ ist der unbekannte Skalierungsfaktor, d.h. unter dem Kameramodell die Quelle der Bildpunktkoordinaten einem Strahl zugeordnet ist.

Herleiten nun die Beziehung zwischen Bildpunkten in einem Framepaar für zwei verschiedene Kameraausrichtungen (siehe Abbildung 4.15). Für einen Weltkoordinatepunkt X sind die projizierten Punkte x_i und x_j in der Bildebene von zwei Bildern i und j gegeben durch

$$x_i = KR_i X, x_j = KR_j X \quad (4.22)$$

Anordnen diese Gleichungen weiter und ersetzen X , wird eine Beziehung aller Punkte im Bildrahmen i auf alle Punkte im Rahmen j erhalten:

$$x_j = KR_j R_i^T K^{-1} x_i \quad (4.23)$$

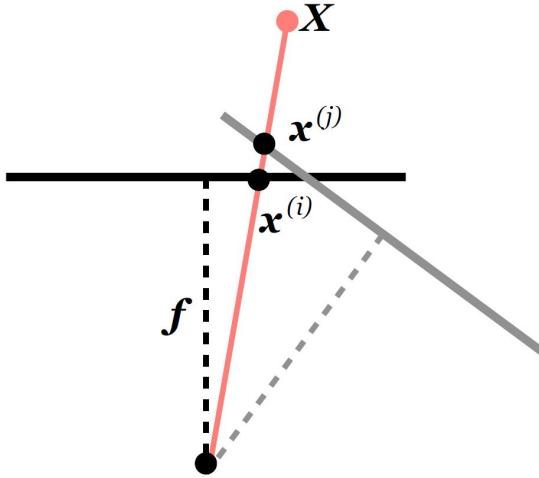


Abbildung 4.16: Rotationsbewegungsmodell.

Bisher wird nur die Beziehung zwischen zwei Bildern desselben Videos betrachtet. Lockern diese Einschränkung, indem Frames von einer Kamera, die sich gemäß R dreht, zu einer anderen Kamera, die sich gemäß R' dreht, abbilden. Es gibt eine Hypothese, dass beide Kamerazentren sich im Ursprung befinden. Dann die Warping-Matrix, die Punkte von einer Kamera auf die andere abbildet, definiert werden können:

$$W = KR'R^T K^{-1} \quad (4.24)$$

Es wird hier angenommen, dass das erste Bild als Referenz genommen wird und der Rotationswinkel 0 ist. Dann die Gleichung kann vereinfacht werden:

$$W = KRK^{-1} \quad (4.25)$$

Kombiniert mit Formel 4.23 können es ausgedrückt als:

$$x_j = Wx_i \quad (4.26)$$

Kalibrierung

In diesem Abschnitt wird die Parameter in der Warping-Matrix berechnet. Überprüfen die letzten beiden Abschnitte, mit Verwendung des SURF [6] die übereinstimmende Punkte in aufeinanderfolgenden Bilderrahmen entdeckt werden dann durch RANSAC [7], Ausreißer verworfen werden. Das Ergebnis ist eine Menge von Punktkorrespondenzen x_i und x_j für alle benachbarten Bilder. Angesichts dieser Grundwahrheit kann man die Kalibrierung als ein Optimie-

rungsproblem formulieren, wobei wir den Fehler bei der Mittelwertbildung im Quadrat aller Punktkorrespondenzen minimieren wollen:

$$J = \sum_{(i,j)} \|x_j - Wx_i\|^2 \quad (4.27)$$

Beachten, dass dies ein nichtlineares Optimierungsproblem ist. Einige nichtlinearer Optimierer könnte verwendet werden, um diese Zielfunktion zu minimieren. Jedoch ist es gefunden, dass Koordinatenabstieg durch direkte objektive Funktionsbewertung schnell konvergiert. Jedes Mal, wenn einen Schritt gemacht wird, bei dem die Zielfunktion J nicht abnimmt, kehren die Schrittrichtung um und verringern die Schrittweite des entsprechenden Parameter. Der Algorithmus endet, sobald die Schrittgröße für alle Parameter unter einen gewünschten Schwellenwert fällt (d.h. Wenn eine Zielgenauigkeit erreicht haben). Das Flussdiagramm des Algorithmus ist wie folgt:

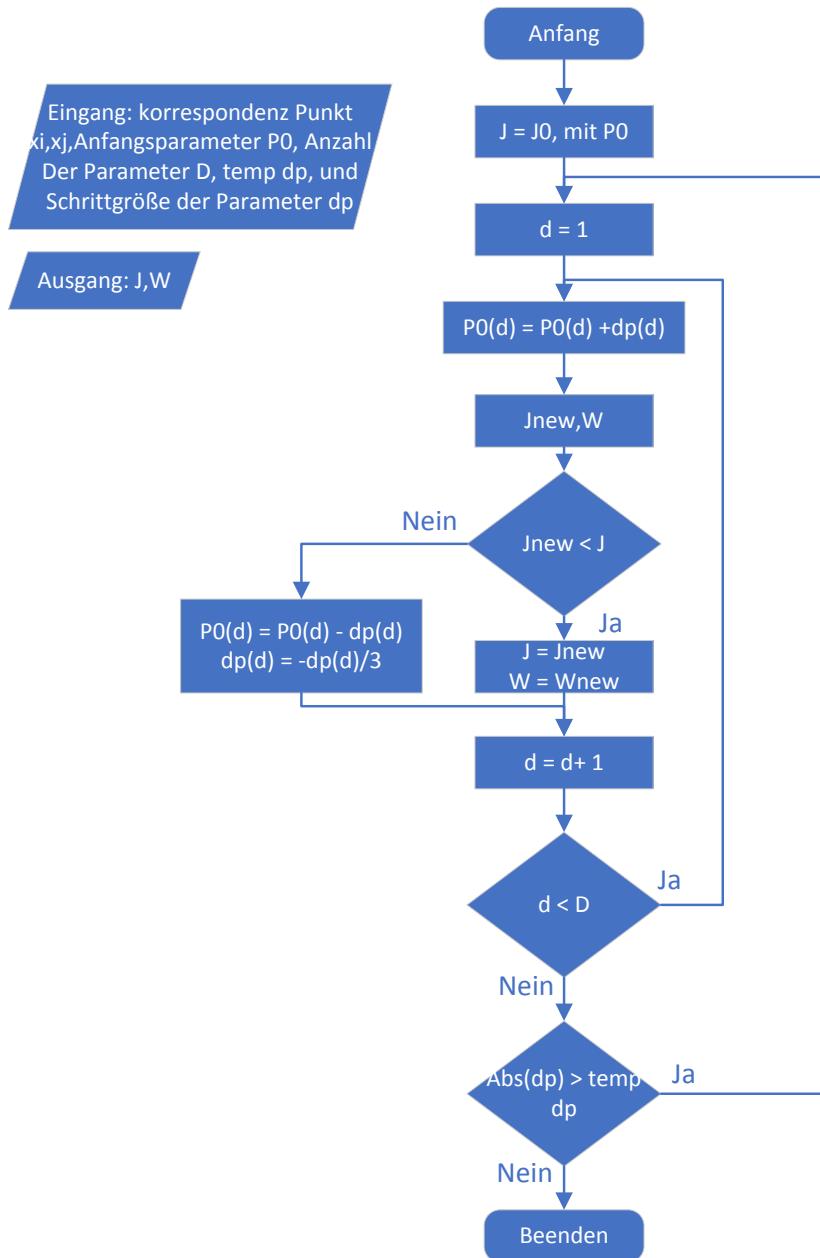


Abbildung 4.17: Flussdiagramm für Algorithmus.

Weiter in einem allgemeineren Fall, d.h. derzeit nicht nur mit Rotations Einfluss, sondern auch Translations Einfluss nehmen. Das Verlauf ist im Allgemeinen gleich. Der Hauptunterschied ist Anzahl der Parameter, die von original nur 3 Rotationsparameter zur jetzt 6 Parameter einschließlich 3 Rotationsparameter und 3 Translationsparameter. Das neu Warping-Matrix

darstellt wie folgen:

$$W = \begin{bmatrix} f & 0 & \frac{w}{2} & 0 \\ 0 & f & \frac{h}{2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R_{11} & R_{21} & R_{31} & 0 \\ R_{12} & R_{22} & R_{32} & 0 \\ R_{13} & R_{23} & R_{33} & 0 \\ t1 & t2 & t3 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{f} & 0 & -\frac{w}{2f} & 0 \\ 0 & \frac{1}{f} & -\frac{h}{2f} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.28)$$

Dann durch Transformationsmatrix können die Koordinaten des zweiten Bildes in die Koordinaten des ersten Bildes umgewandelt werden. Abbildung 4.17 zeigt diese Verlauf. Die Ergebnisse der beiden experimentellen Fälle werden im nächsten Kapitel detailliert beschrieben.

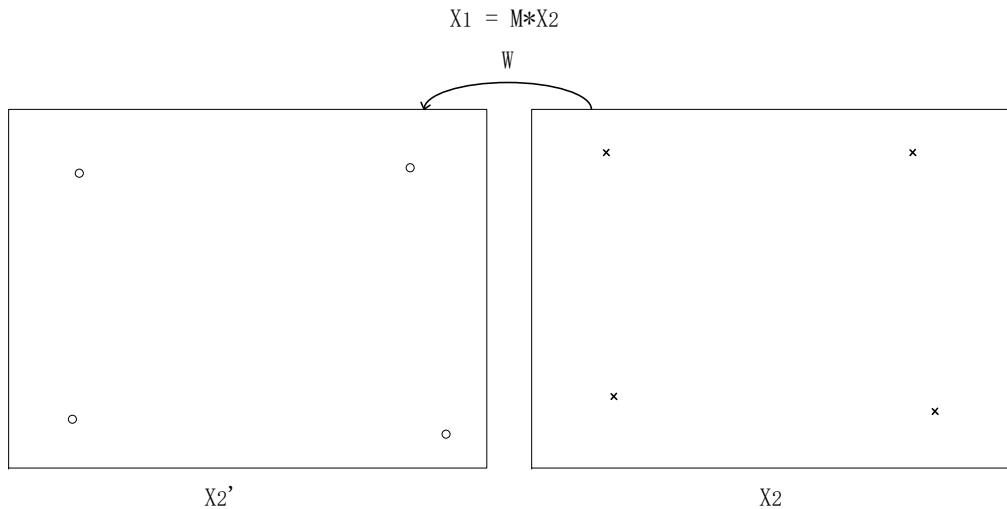


Abbildung 4.18: Transformation in eine Koordinate.

4.3 Differenzbild Optimierung

Durch Bildregistration erhalten eine Reihe Bildern von der Kamera, deren Koordinaten in das-selbe Koordinatensystem umgewandelt wurden. Zwei beliebige Bilder subtrahiere und einen Differenzbild enthalten. Eine Beispiel Differenzbild wird in Abbildung 4.18 gezeigt. An jeder Ecke der Modulationsbereich existiert ein QR-Pattern. Die Ziel in diesem Abschnitt ist eine Bilder herstellen, der die QR Muster Detektion vereinfachen werden können.

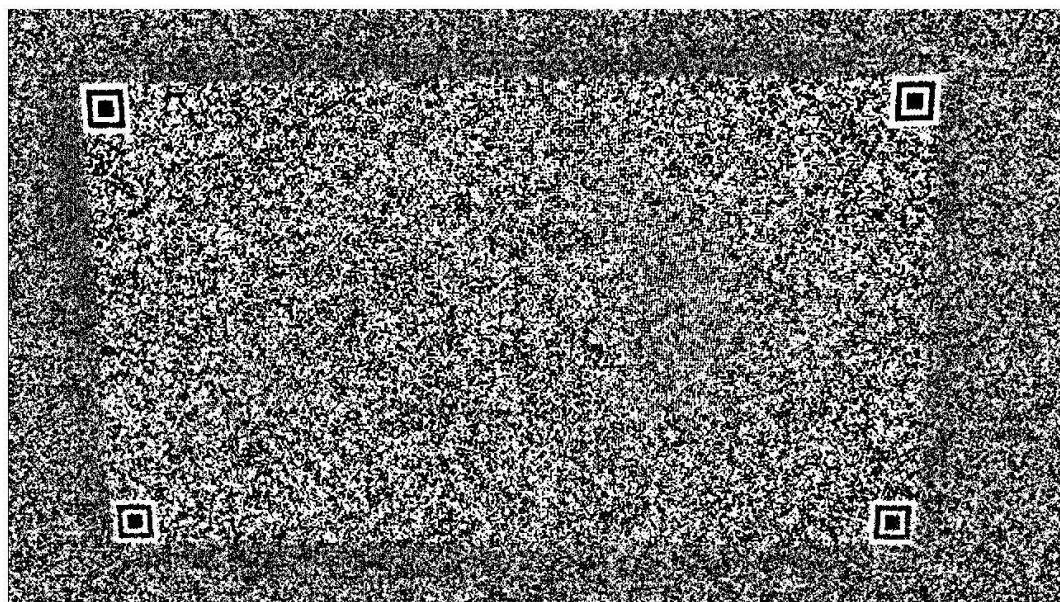


Abbildung 4.19: Differenzbild Beispiel.

Hier in dieser Arbeit wird eine einfach Differenzbild Algorithmus vorstellen, wie folgen darstellt:

1. Wähle zwei beliebige Bilder und subtrahiere, um Differenzbild zu enthalten.
2. Berechnen die Energie jedes Differenzbild. Um den Berechnungsaufwand zu reduzieren, kann nur der zentrale Bereich berechnet werden.
3. Sortiere und nehmen 3 Differenzbildern, die maximaler Energie verfügen.
4. Addieren diese 3 Differenzbildern und nächste Bildverarbeitung.

Das Skript in Matlab ist wie folgt:

Quellcode 4.1: MATLAB Code Binarisierung.

```
1 %% Differenzbild
2 diff = functions.createDifferenzbild(U2new);
3 Mal_num = 3;
4 diffplus = functions.sum_of_diff(abs(diff),Mal_num);
5
```

4.4 Bildverarbeitung

Der detaillierte Inhalt der Bildverarbeitung wurde in der ersten Methode eingeführt, hier ist nur eine einfache Beschreibung der verwendeten Funktionen in Matlab. Durch Bildverarbeitung können kleine Punkte und Lücken, die durch Rauschen und Fehler verursacht werden, entfernt werden, um die nachfolgende Detektion zu erleichtern.

Bild Binarisierung

Für die Schwellenwertbildung und das Erstellen eines Binärbildes wurde eine Funktion namens "imbinarize" verwendet. Diese Funktion erhält das Bild und verwendet einen anpassungsfähigen Schwellwert, um das Schwarz-Weiß-Bild zurückzugeben. Das ist genug für QR-Pattern Detektion, weil es nur dunkle und helle Module enthält, die binär 1 bzw. 0 sind. Das Skript ist wie folgt:

Quellcode 4.2: MATLAB Code Binarisierung.

```
1 % Threshold image - adaptive threshold
2 BW = imbinarize(X, 'adaptive', 'Sensitivity', 0.500000, ...
3   'ForegroundPolarity', 'bright');
4
```

Medianfilter

Der Grund für das Median-Filtern ist, dass manchmal beim Prozess Binarisierung aus einem Bild die Muster wie Salz- und Pfeffergeräusche erzeugt werden können. Um diesen Fehler zu vermeiden, ist Median Filterung eine leistungsfähige Methode. Das Skript ist so einfach wie `medfilt2(img)`. Zur Verbesserung der Ergebnisse kann natürlich die verschiedenen Fenstergrößen für die Matlab-Funktion verwenden.

Morphologie

Mit öffnenden und schließenden Filtern können die Lücken zwischen Blöcken und die kleinen Punkte von Rausch stark reduziert werden, was das resultierende binäre Bild zu einer guten Schätzung des QR-Pattern macht. Das Skript ist wie folgt:

Quellcode 4.3: MATLAB Code Beispiel.

```
1 % Close mask with square
2 width = 5;
3 se = strel('square', width);
4 BW = imclose(BW, se);
5
```

```

6 % Open mask with square
7 width = 5;
8 se = strel('square', width);
9 BW = imopen(BW, se);
10

```

4.5 QR Musters Detektion

Nach Bildverarbeitung wird die QR Musters Detektion ausgeführt, d.h. lokalisieren die QR Musters im Bild rekonstruiere die Bild. Abbildung 4.19 zeigt eine simple Struktur des QR Musters. Es gibt insgesamt vier QR Muster, jede an der Ecke der Modularisationsbereiche.

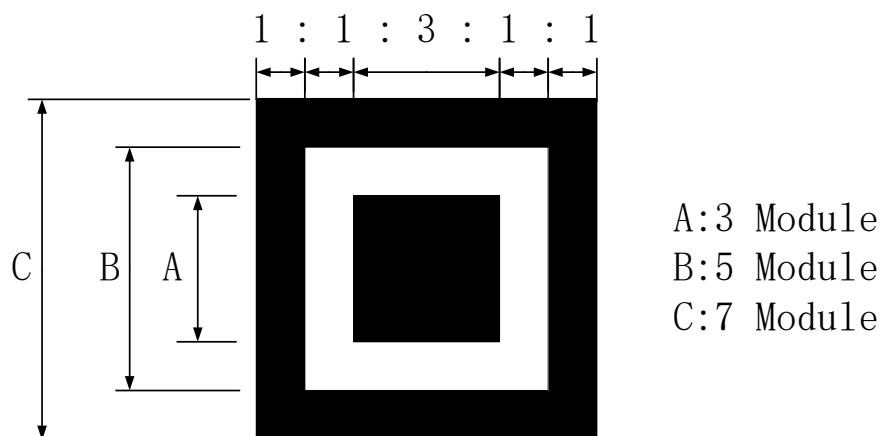


Abbildung 4.20: QR Pattern.

Jedes Muster kann als drei konzentrische Quadrate betrachtet werden und besteht aus einer schwarzen (dunklen) 7×7 Modulen, einer weißen (hellen) 5×5 Modulen und schließlich einer dunklen 3×3 Modulen. Das Breiteverhältnis der alternativen Schwarz- und Weißmodul in jedem Muster beträgt etwa 1 : 1 : 3 : 1 : 1 in jeder Richtung, wie zeigt in Abbildung 4.20. Diese wichtige Eigenschaft hilft uns, die Lokalität der QR Muster zu finden. Es gibt noch ein Trennmuster, das ein Funktionsmuster mit aller weißen (hellen) Module, Breit ein Modul. Es spielt eine Rolle als einer Grenze zwischen den Mustern und dem Datenbereich.

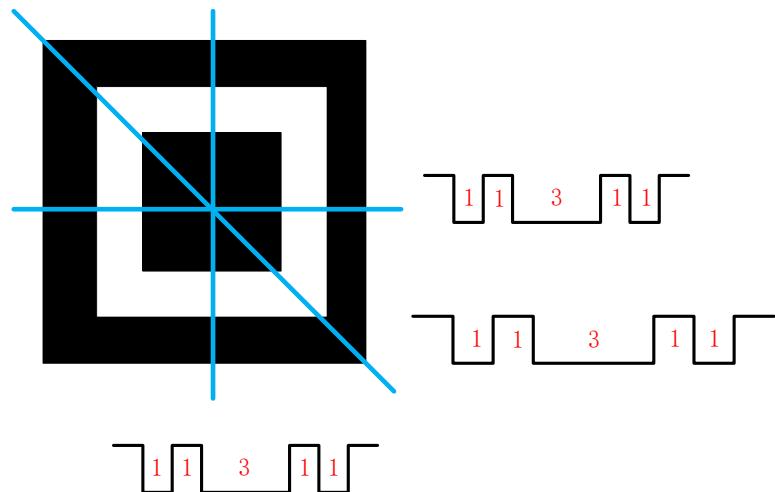


Abbildung 4.21: QR Pattern Ratio.

Die Unterfunktion "detectFIP" verwendet das folgende Verfahren zum Finden eines QR Musters:

Schritt 1:

Um die Aufwand für Berechnung zu reduzieren, brauchen es nicht die Analysen für ganzen Pixel im Bild. d.h. nur das Pixel am der Ecke werden analysiert.

Quellcode 4.4: Detektion Bereich.

```

1 % take the size and the dimension of the image
2 h = size(img, 1); % height
3 w = size(img, 2); % weight
4
5 s.offset = 0;
6 s.sizeH = ceil(h/4); %1/4 height of the image
7 s.sizeW = ceil(w/4); %1/4 weight of the image
8
9 PatternAreaBW(:,:,1) = img(s.offset+1:s.sizeH+s.offset, ...
10    s.offset+1:s.sizeW+s.offset);
11 PatternAreaBW(:,:,2) = img(s.offset+1:s.sizeH+s.offset, ...
12    end-s.sizeW-s.offset+1:end-s.offset);
13 PatternAreaBW(:,:,3) = img(end-s.sizeH-s.offset+1:end-s.offset, ...
14    s.offset+1:s.sizeW+s.offset);
15 PatternAreaBW(:,:,4) = img(end-s.sizeH-s.offset+1:end-s.offset, ...
16    end-s.sizeW-s.offset+1:end-s.offset);
17

```

Schritt 2:

Scannen jede Zeile des Bereich. Speichern die Länge der Schwarz und Weiß Module in eine fünf Element Vektor. Die Länge heißt die Anzahl aufeinanderfolgende Pixel in einer Zeile mit die gleiche Farbe. Speicherreihenfolge ist Schwarz-Weiß-Schwarz-Weiß-Schwarz. Unterfunktion "FinderPatternFinder" implementieren diese Aufgabe.

Quellcode 4.5: MATLAB Code Beispiel.

```
1
2 function patternInfo = FinderPatternFinder(bw, tryHarder)
3
```

Schritt 3:

Immer wenn der Vektor voll ist, nutzen die Unterfunktion "foundPatternCross". Solange die Anteile der Zählungen nahe genug an den $1 : 1 : 3 : 1 : 1$ Verhältnissen ist, als eine Übereinstimmung betrachtet werden und eine Bestimme zurückgeben.

Quellcode 4.6: MATLAB Code Beispiel.

```
1
2 function result = foundPatternCross(stateCount)
3
```

Schritt 4:

Wenn ein horizontaler Abtasten ein mögliches Muster findet, dann läuft eine Kreuzprüfung mit vertikalen Abtasten. Falls erfolgreich, machen ein Kreuz-Kreuz-Check mit einem anderen horizontalen Abtasten. Dies wird hauptsächlich benötigt, um die reale horizontale Mitte des Musters in extremer Schräglage Fällen zu lokalisieren. Die Unterfunktion "handlePossibleCenter" wurde definiert, um das Überprüfung zu implementieren.

Quellcode 4.7: MATLAB Code Beispiel.

```
1
2 [confirmed, possibleCenters] = ...
3     handlePossibleCenter(bw, possibleCenters, ...
4     stateCount, i, j);
5
```

Schritt 5:

Je nach einer potenzielle Muster Zentrum gefunden ist, Löschen den Vektor, um wieder mit der Suche zu beginnen, sonst verschieben die Element des Vektors um zwei zurück.

Quellcode 4.8: MATLAB Code Beispiel.

```

1 % Clear state to start looking again
2         currentState = 0;
3         stateCount = zeros(1, 5);
4
5             % shift counts back by two
6             stateCount(1) = stateCount(3);
7             stateCount(2) = stateCount(4);
8             stateCount(3) = stateCount(5);
9             stateCount(4) = 1;
10            stateCount(5) = 0;
11            currentState = 3;
12

```

Schritt 6:

Schließlich falls es nicht nur eine potentiell Muster Zentrum gefunden, nutzen diese Funktion `SelectBestPattern` die beste zu auswählen.

Quellcode 4.9: MATLAB Code Beispiel.

```

1
2 patternInfo = selectBestPattern(possibleCenters);
3

```

Es sollte angemerkt werden, dass wenn die mögliche Musterung nach dem Ende der Erkundung nicht gefunden wird, ein Signal "Nofind" übergeben wird und die System zu Differenzbild Optimierung zurückkehren wird. Die Operation besteht darin, der ursprünglichen Bild, die aus drei Differenzbild besteht, ein mehrere Differenzbild hinzuzufügen.

Quellcode 4.10: MATLAB Code Beispiel.

```

1 while true
2 diffplus = functions.sum_of_diff(abs(diff), Mal_num);
3
4 [BW, ~] = functions.imageSegmenter(diffplus);
5
6 [FiPx, FiPy, ux, vx, cr, Nofind] = detectFIP(BW);
7
8 if Nofind
9     Mal_num = Mal_num+1;
10 else
11     break;
12 end
13 end

```

Die folgenden Bilder zeigen einige Beispiele für das Lokalisierung von QR Muster.

5 Auswertung

Text hier zwischen. Referenz auf ein Bild mit cleverref (siehe ??). Und dann noch ein paar Zitierungen [9]–[11].

5.1 Section

Jetzt nur noch schreiben! :)

6 Zusammenfassung

Nicht vergessen, dass Überschriften nicht aufeinander folgen dürfen...

6.1 Section

Anhang A

Erster Anhang

A.1 Section

Jetzt nur noch schreiben! :)

Abbildungsverzeichnis

2.1 Eine beispielhafte Implementierung des DaViD-Systems	3
2.2 Schematische Darstellung von DaViD System	4
2.3 Blockschaltbild der Signalverarbeitung in zeitlicher differentieller Modulation	5
2.4 Ein Beispiel einer modulierten Bildfolge	7
3.1 TeXLipse Spell Checker preferences	11
3.2 PdfLatex Builder Settings	11
3.3 TeXLipse Viewer Settings	12
3.4 SumatraPDF Options	12
4.1 Strukturdiagramm	15
4.2 Flussdiagramm der Bildregistrierung	15
4.3 Scale space	17
4.4 Extreme Wert Erkennung	18
4.5 Dominante Orientierung Feststellen	19
4.6 Merkmalspunkt Deskriptor	19
4.7 SURF Merkmal	20
4.8 Linien Detektion	21
4.9 OhneRANSAC	23
4.10 MitRANSAC	23
4.11 Kamera Model	24
4.12 Transformation vom Weltkoordinatensystem zum Kamerakoordinatensystem	25
4.13 Rotation um Z-Achse	25
4.14 Transformation vom Kamerakoordinatensystem zum Bild Koordinatensystem	27
4.15 Konvertierung von Bildkoordinatensystem zu Pixelkoordinatensystem	28
4.16 Rotationsbewegungsmodell	30
4.17 Flussdiagramm für Algorithmus	32
4.18 Transformation in eine Koordinate	33
4.19 Differenzbild Beispiel	34
4.20 QR Pattern	36
4.21 QR Pattern Ratio	37

Tabellenverzeichnis

4.1 Parameter des Kameras im Vergleich	29
--	----

Quellcodeverzeichnis

4.1	MATLAB Code Binarisierung	34
4.2	MATLAB Code Binarisierung	35
4.3	MATLAB Code Beispiel	35
4.4	Detektion Bereich	37
4.5	MATLAB Code Beispiel	38
4.6	MATLAB Code Beispiel	38
4.7	MATLAB Code Beispiel	38
4.8	MATLAB Code Beispiel	39
4.9	MATLAB Code Beispiel	39
4.10	MATLAB Code Beispiel	39

Literatur

- [1] R. Kays, C. Brauers und J. Klein, „Modulation concepts for high-rate display-camera data transmission“, in *2017 IEEE International Conference on Communications (ICC)*, Mai 2017, S. 1–6.
- [2] R. Kays, C. Brauers und J. Klein, „DaViD : Data Transmission Using Video Devices-An Innovative System for Smart Media Applications“, 2016.
- [3] R. Kays, „Modulation concepts for visible light communication using video displays“, in *2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Sep. 2015, S. 388–392.
- [4] R. Kays, „Visible light communication using TV displays and video cameras“, in *2015 IEEE International Conference on Consumer Electronics (ICCE)*, Jan. 2015, S. 554–555.
- [5] S. Lin und B. W. Kernighan, „An Effective Heuristic Algorithm for the Travelling-Salesman Problem“, *Operations Research*, Jg. 21, S. 498–516, 1973.
- [6] H. Bay, A. Ess, T. Tuytelaars und L. V. Gool, „SURF:Speeded Up Robust Features“, Bd. 110, 2008, S. 346–359.
- [7] M. A. Fischler und R. C. Bolles, „Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography“, *Commun. ACM*, Jg. 24, Nr. 6, S. 381–395, Juni 1981. Adresse: <http://doi.acm.org/10.1145/358669.358692>.
- [8] Z. Zhengyou, „A Flexible New Technique for Camera Calibration“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 22, S. 1330–1334, Dez. 2000.
- [9] R. Reinhold und R. Kays, „Improvement of IEEE 802.15.4a IR-UWB for time-critical industrial wireless sensor networks“, in *2013 IFIP Wireless Days (WD)*, Nov. 2013, S. 1–4.
- [10] T. K. Moon, *Error Correction Coding, Mathematical Methods and Algorithms*, 1. Aufl. Hoboken, NJ: Wiley-Interscience, Juni 2005, 800 S.
- [11] *Ieee standard for local and metropolitan area networks, part 15.4: Low-rate wireless personal area networks*, IEEE Std 802.15.4-2011, 2011.

Eidesstattliche Versicherung

Blaubär, Käpt'n Kevin
Name, Vorname

123456
Matr.-Nr.

Ich versichere hiermit an Eides statt, dass ich die vorliegende Bachelorarbeit mit dem Titel
Tasty Kanalmodell für die drahtlose Kommunikation zwischen Gebäuden und Außeninstallationen
selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäß Zitate kenntlich gemacht.
Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Dortmund, 16. August 2018
Ort, Datum

Unterschrift

Belehrung:

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -)

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird ggf. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Dortmund, 16. August 2018
Ort, Datum

Unterschrift