

# **Interner Bericht**

**DLR-IB-FT-BS-2018-255**

## **Erweiterung einer Simulationsumgebung für Feature Tracker um Fehlermodelle**

### **Hochschulschrift**

Bayer, Julian

Deutsches Zentrum für Luft- und Raumfahrt

Institut für Flugsystemtechnik  
Braunschweig



**DLR**

**Deutsches Zentrum  
für Luft- und Raumfahrt**

Institutsbericht  
**DLR-IB-FT-BS-2018-255**

## **Erweiterung einer Simulationsumgebung für Feature Tracker um Fehlermodelle**

Bayer, Julian

Institut für Flugsystemtechnik  
Braunschweig

088	Seiten
066	Abbildungen
004	Tabellen
033	Referenzen

Deutsches Zentrum für Luft- und Raumfahrt e.V.  
Institut für Flugsystemtechnik  
Abteilung Unbemannte Luftfahrzeuge

**Stufe der Zugänglichkeit: I, Allgemein zugänglich: Der Interne Bericht wird elektronisch ohne Einschränkungen in ELIB abgelegt. Falls vorhanden, ist je ein gedrucktes Exemplar an die zuständige Standortbibliothek und an das zentrale Archiv abzugeben.**

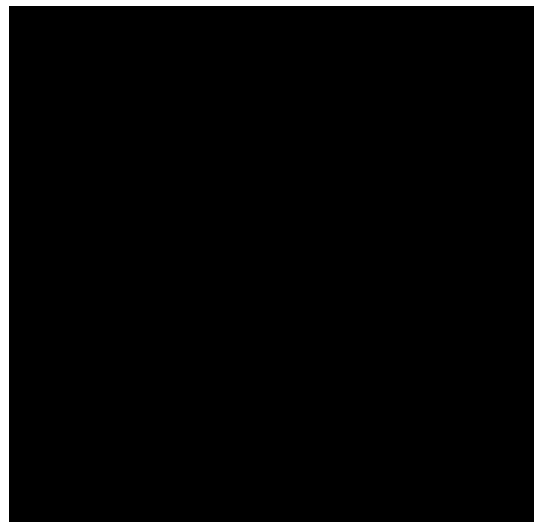
Braunschweig, den 02.02.2024

Institutsdirektor: Prof. Dr.-Ing. S. Levedag

Abteilungsleitung: Johann Dauer

Betreuer:in: Nikolaus Ammann

Verfasser:in: Julian Bayer





**Technische  
Universität  
Braunschweig**



**Deutsches Zentrum  
für Luft- und Raumfahrt e.V.**



Wissen für Morgen

# **Bachelorarbeit**

## **Erweiterung einer Simulationsumgebung für Feature Tracker um Fehlermodelle**

**Institut für Flugsystemtechnik**

28. Februar 2018

### **Julian Bayer**

Erstprüfer: Prof. Dr.-Ing. Stefan Levedag  
Zweitprüfer: Prof. Dr.-Ing. Peter Hecker  
Betreuer: Nikolaus Ammann, M.Sc.

**Deutsches Zentrum  
für Luft- und Raumfahrt e.V.**  
in der Helmholtz-Gemeinschaft



DLR Institut für Flugsystemtechnik  
Postfach 32 67, 38022 Braunschweig

Institut für  
Flugsystemtechnik  
Prof.Dr.-Ing. S. Levedag

Ihr Zeichen  
Ihr Schreiben  
Unser Zeichen

Ihr Gesprächspartner  
Telefon 0531 295-  
Telefax 0531 295-  
E-Mail

Nikolaus Ammann  
2922  
2647  
Nikolaus.ammann@dlr.de

07.11.2017

Bachelorarbeit für  
Herrn Julian Bayer  
Mat.-Nr.: 4504700

Thema:

Erweiterung einer Simulationsumgebung für Feature Tracker um Fehlermodelle

*Extension of a feature tracking simulation environment by error models*

Erläuterung:

Unbemannte Luftfahrzeuge (UAVs) nutzen zur Selbstlokalisierung vor allem globale Navigationssysteme, wie z.B. GPS, GLONASS oder Galileo. Um robuster gegen Störungen der GNSS zu sein, wird an alternativen Navigationssystemen (engl.: Alternative Positioning Navigation and Time (APNT)) geforscht. Die Nutzung von optischen Sensordaten, wie Kamerabildern, ist dabei eine große Forschungsrichtung. Basierend auf den Bilddaten, wird dabei die Eigenbewegung geschätzt. Diese Verfahren beruhen meist auf visueller Odometrie oder Simultaner Lokalisierung und Kartenerstellung (SLAM). Diese wiederum nutzen meist extrahierte Bildmerkmale eines Feature Trackers.

Das DLR hat für die Bewertung und den Test von solchen visuellen Odometrie bzw. Simultaner Lokalisierung und Kartenerstellung (SLAM) Algorithmen eine Simulationsumgebung für Feature Tracker entwickelt. Diese kann an die abteilungsinterne Software-In-The-Loop bzw. Hardware-In-The-Loop Simulation der UAVs angebunden werden. Was den Test und die Bewertung verschiedener Algorithmen zur Navigation der UAVs ohne GNSS Signal ermöglicht.

Die aktuelle Version der Feature Tracking Simulation liefert bis auf Diskretisierungsfehler nur ideale Werte. Um eine realistischere Bewertung zu ermöglichen, soll diese Simulation um Fehlermodelle erweitert werden. Diese Fehlermodelle sollen sich an der Performanz realer Feature Tracking Algorithmen orientieren, wobei diese in der Häufigkeit und Gewichtung angepasst werden können sollen.



## Aufgabenstellung:

Es soll eine vorhandene Simulationsumgebung für Feature Tracking Algorithmen um Fehlermodelle realer Feature Tracking Algorithmen erweitert werden. Dazu soll zunächst eine Literaturrecherche durchgeführt werden, bei der die verschiedenen Verfahren aufgelistet werden. Die Auflistung soll eine kurze Erklärung der Funktionsweise und die damit einhergehenden Vor- und Nachteile beinhalten. Ausgehend auf dieser Recherche sollen typische Fehlermerkmale extrahiert und in der Simulationsumgebung nachempfunden werden. Die Fehlermodelle sollen dabei in ihren Eigenschaften (Auftrittswahrscheinlichkeit, Ausmaß) anpassbar sein. Die Implementierung erfolgt dabei in C++.

### Teilaufgaben:

- ✓ - Literaturrecherche
- ✓ - Gegenüberstellung verschiedener Feature Tracker (KLT und weitere) bzgl. Funktionsweise, Fehlermerkmalen, Vor- und Nachteilen
- ✓ - Erweiterung des Feature-Emulators um entsprechende Fehlermodelle

Bearbeitungszeit: 3 Monate

Betreuer: Nikolaus Ammann, M.Sc.  
Institut für Flugsystemtechnik, DLR Braunschweig

Ausgabe: 01.12.2017

Abgabe:

---

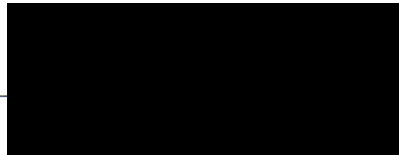
Prof. Dr.-Ing. Stefan Levedag

# Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die nachfolgende Arbeit selbständig und nur unter Zuhilfenahme der angegebenen Literatur angefertigt habe.

28.02.2018,

Datum, Unterschrift



|

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>7</b>
1.1. Motivation . . . . .	8
1.2. Ziel der Arbeit . . . . .	8
1.3. Aufbau der Arbeit . . . . .	8
<b>2. Grundlagen der Trackingverfahren</b>	<b>9</b>
2.1. Feature Tracking . . . . .	9
2.1.1. Markerbasierte Trackingverfahren . . . . .	9
2.1.2. Markerlose Trackingverfahren . . . . .	12
2.2. Detektoren . . . . .	13
2.2.1. Kantendetektoren . . . . .	13
2.2.2. Eckendetektor . . . . .	15
2.2.3. Blobdetektor . . . . .	17
2.2.4. Ridgedetektor . . . . .	18
2.3. Deskriptoren . . . . .	18
2.3.1. SIFT-Deskriptor . . . . .	19
2.3.2. SURF-Deskriptor . . . . .	21
2.3.3. BRISK-Deskriptor . . . . .	23
2.3.4. Weitere Deskriptoren . . . . .	23
2.4. Korrespondenzen . . . . .	23
2.4.1. Ähnlichkeitsmaße durch Abstandsermittlung . . . . .	23
2.4.2. Kreuzkorrelation . . . . .	24
2.5. (KLT) Kanade-Lucas-Tomasi-Algorithmus . . . . .	25
2.6. (SIFT) Scale-Invariant-Feature-Transform-Algorithmus . . . . .	29
2.7. (SURF) Speeded-Up-Robust-Features-Algorithmus . . . . .	31
2.8. (BRISK) Binary-Robust-Invariant-Scalable-Keypoints-Algorithmus . . . . .	35
<b>3. Analyse der Feature-Tracker-Algorithmen</b>	<b>38</b>
3.1. Auswertung der Trackingverfahren . . . . .	38
3.1.1. Vergleich der Trackingverfahren . . . . .	38
3.1.2. Vergleich der Detektoren . . . . .	38
3.1.3. Vergleich der deskriptiven Trackingverfahren . . . . .	39
3.1.4. Rechenzeit der Trackingverfahren . . . . .	41
3.2. Fehlermerkmale der Trackingverfahren . . . . .	42
3.2.1. Deskriptorlose Trackingverfahren . . . . .	42
3.2.2. Deskriptive Trackingverfahren . . . . .	44
3.3. Konklusion . . . . .	47

3.4. Auflistung der Fehlermerkmale . . . . .	48
3.5. Anwendung eines realen Feature Trackers . . . . .	49
<b>4. Simulationsumgebung</b>	<b>50</b>
4.1. Beschreibung der vorhandenen Simulation . . . . .	50
4.1.1. Feature Tracker Simulator . . . . .	51
4.1.2. Objekterkennung mittels des Lochkameramodells . . . . .	52
4.2. Extrahieren der Fehlermerkmale in die Simulation . . . . .	53
4.2.1. Inkrementelle ID-Generierung . . . . .	53
4.2.2. Rauschen . . . . .	55
4.2.3. Driftfehler . . . . .	57
4.2.4. Kantendriftfehler . . . . .	59
4.2.5. Mismatches . . . . .	62
4.2.6. Rechenzeit . . . . .	63
4.3. Häufigkeit der Fehlermerkmale . . . . .	64
4.3.1. Rauschen . . . . .	64
4.3.2. Driftfehler . . . . .	65
4.3.3. Kantendriftfehler . . . . .	66
4.3.4. Mismatches . . . . .	67
4.3.5. Rechenzeit . . . . .	67
4.3.6. Standardparametersatz der Fehlermodelle . . . . .	67
4.4. Weitere Ergänzungen im Feature Tracker Simulator . . . . .	69
<b>5. Ergebnis</b>	<b>70</b>
5.1. Validierung der implementierten Fehlermodelle . . . . .	70
<b>6. Zusammenfassung</b>	<b>79</b>
<b>7. Ausblick</b>	<b>80</b>
<b>Literaturverzeichnis</b>	<b>81</b>
<b>Abbildungsverzeichnis</b>	<b>83</b>
<b>Tabellenverzeichnis</b>	<b>86</b>
<b>A. Anhang</b>	<b>87</b>
A.1. Ausführliche Matrizendarstellung des Lochkameramodells . . . . .	87

# 1 Einleitung

Digitalisierung gewinnt in vielen Anwendungsbereichen immer mehr an Bedeutung, wie auch in der Luftfahrt. Mit Hilfe der bildgebenden Sensoren am Luftfahrzeug können durch die Umweltwahrnehmung von Kameras oder Laserscannern eine genaue Position des Luftfahrzeuges ermittelt werden und darüber hinaus andere Luftverkehrsteilnehmer oder Hindernisse in der Umwelt erkannt werden. Durch die Kombination Satelliten- und Inertialsensoren sowie bildgebenden Sensoren wird die Verfügbarkeit bei einem möglichen Sensorausfall erhöht und zusätzlich eine hohe Genauigkeit und Integration der verschiedenen Sensoren erzielt, die im Zusammenwirken ein zuverlässiges Gesamtsystem der Navigation ermöglichen.

Die unbemannte Luftfahrt stellt eine wichtige Ergänzung der bemannten Luftfahrt dar. Die unbemannte Luftfahrt nutzt ebenfalls wie in der bemannten Luftfahrt die globalen Satellitennavigationssysteme (GNSS) zur Positionsermittlung, die zu Abweichungen von mehreren Metern führen können. Die Ursachen der Abweichungen sind auf Positionsungenauigkeiten des Satellitenorbits, atmosphärische Einflüsse, Wettereinflüsse und Signalreflexionen zurückzuführen. Des Weiteren können Ungenauigkeiten der satellitenunterstützten Methoden durch Signalreflexionen an dicht besiedelten Wohngebieten hervorgerufen werden oder im schlimmsten Fall zu einem Abbruch der Signalverbindung führen.

Um eine verbesserte, sichere und effiziente Lösung der Ortung bei der Navigation zu ermöglichen, werden derzeit an alternativen Navigationssystemen (APNT) geforscht. Beispiele hierfür sind die Entfernungsmessung mit Hilfe des digitalen Luftfahrt-Funksystem (L-DCAS) sowie die genaue Ermittlung der Luftfahrzeugposition durch visuelle Odometrie-Verfahren oder der simultanen Lokalisierungs- und Kartenerstellungsverfahren (SLAM) aus extrahierten Bildmerkmalen.

Am Institut für Flugsystemtechnik am Deutschen Zentrum für Luft- und Raumfahrt in Braunschweig steht für die Evaluierung der Selbstlokalisierung durch visuelle Odometrie oder durch simultane Lokalisierung und Kartenerstellungsalgorithmen ein Feature Tracker Simulator zur Verfügung. Die Simulation des Feature Trackers kann an die abteilungsinterne Software-In-The-Loop bzw. Hardware-In-The-Loop der unbemannten Luftfahrzeuge angeschlossen werden.

Für eine zuverlässige Bewertung müssen möglichst realistische Simulationen entwickelt werden. Der vorhandene Feature Tracker Simulator extrahiert ideale Bildmerkmale, die für eine Evaluation der Selbstlokalisierungssysteme nur bedingt ausreichen. In der vorliegenden Arbeit sollen Fehlermerkmale im Rahmen der Simulation realistisch nachgebildet werden.

## 1.1. Motivation

Simulationen werden in vielen Anwendungsbereichen genutzt. Die hierbei genutzten Vorteile der leistungsstarken Rechenkapazitäten schaffen eine schnelle und effiziente Evaluation neuartiger Systeme. Simulationen können eine enorme Kosten- und Zeitersparnis einbringen. Für eine effiziente Evaluierung von einzelnen bis hin zu komplexen Systemkomponenten können softwaretechnische Maßnahmen Abhilfe leisten. Eine Simulation baut je nach Anwendungsfall auf einer geeigneten Programmiersprache auf und bildet anhand eines passenden Simulationsmodells die reale Umgebung ab. Für eine zuverlässige Modellierung von Hardwarekomponenten ist ein genaues Verständnis der Struktur, der Funktion und des Systemverhaltens der Simulationsmodelle erforderlich.

Um Flugversuche zu vermeiden und die Bewertung der Selbstlokalisierung zu vereinfachen, werden Bildmerkmale mit dem Feature Tracker Simulator simuliert und an das zu bewertende System verschickt.

Mit Hilfe der Hardwaresimulation, der sogenannten Hardware-In-The-Loop, lässt sich eine Verbindung mit der Hardware des unbemannten Luftfahrzeuges herstellen. Da die Trackingalgorithmen des Feature Tracker Simulators ideale Werte liefern und diese für eine Evaluation der Selbstlokalisierungssysteme nicht ausreichen, müssen zusätzliche Fehlermodelle konzipiert werden, um einen möglichst realistischen Feature Tracker Simulator zu erhalten.

## 1.2. Ziel der Arbeit

Für den vorhandenen Feature Tracker Simulator sollen für bestimmte Trackingverfahren die Fehlermerkmale simuliert werden. Um eine realitätsnahe Feature Tracker Simulation zu ermöglichen, müssen zunächst die Funktionsweisen und Unterschiede der Trackingverfahren genau untersucht werden.

Die sich aus den Untersuchungen der Trackingverfahren ergebenden Fehlermerkmale sollen anschließend durch Fehlermodelle in der Simulationsumgebung des Feature Trackers nachempfunden werden. Für eine realitätsnahe Nachempfindung der Fehlermerkmale im Feature Tracker Simulator sollen hierzu die Auftrittswahrscheinlichkeiten sowie die Ausmaße der Fehlermerkmale aufgestellt werden. Schlussendlich soll eine realitätsnahe Simulation mit den ausgewählten Trackingverfahren des Feature Trackings bestehen, die künstliche Fehlermerkmale erzeugt.

## 1.3. Aufbau der Arbeit

Im Kapitel „Grundlagen der Trackingverfahren“ wird zunächst das Feature Tracking im Allgemeinen vorgestellt. Auch werden zusätzlich die einzelnen Bestandteile des Feature Trackings näher beschrieben. Anschließend erfolgt eine genaue Untersuchung der für diese Arbeit relevanten Trackingalgorithmen. Mit der „Analyse der Feature Tracker Algorithmen“ in Kapitel 3 werden die relevanten Trackingverfahren miteinander bei verändernden Umgebungsbedingungen verglichen und auf die Ursachen der Fehlermerkmale untersucht. Das Vorgehen der Generierung von Fehlermodellen der erkannten Fehlermerkmale werden anschließend in Kapitel 4, „Simulationsumgebung“ erläutert. In Kapitel 5, „Ergebnisse“ erfolgt die Validierung der implementierten Fehlermodelle in der erweiterten Feature Tracker Simulation.

# 2 Grundlagen der Trackingverfahren

Das Kapitel Grundlagen der Trackingverfahren liefert zunächst eine kurze Beschreibung des allgemeinen Begriffs Feature Tracking. Im Anschluss folgt eine Vorstellung der markerbasierten und der markerlosen Trackingverfahren, die große Untergruppen des Feature Trackings bilden. Hierbei liegt der besondere Fokus auf den markerlosen Trackingverfahren aus denen in nachfolgenden Kapiteln Fehlermerkmale erkannt und diese in dem angelehnten markerbasierten Feature Tracker Simulator nachempfunden werden. Anhand der mathematischen Herleitungen werden die Algorithmen der markerlosen Trackingverfahren näher beschrieben.

## 2.1. Feature Tracking

Feature Tracking beschreibt im Wesentlichen das Erkennen und Verfolgen von Interessensmerkmalen. Unter dem Begriff des Trackings versteht man im Allgemeinen die Verfolgung des Interessensmerkmals.

Für die Erkennung von Objekten mit dem Feature Tracking in einer Umgebung bestehen diverse technische Ansätze. Im Allgemeinen unterscheidet man beim Tracking zwischen sensorbasierten und optischen Trackingverfahren [1].

Bei dem optischen Trackingverfahren unterscheidet man zwischen markerbasierendem und markerlosem Tracking. Für das markerbasierende Trackingverfahren werden zur Lokalisierung Muster an einem Gegenstand angebracht, die mit der rechnergestützten Bildverarbeitung erkannt werden können [1].

### 2.1.1. Markerbasierte Trackingverfahren

Bei den markerbasierten Trackingverfahren werden Muster in der Umgebung aufgestellt, die von Bildverarbeitungssystemen leicht erkannt werden [1]. In einer gewonnenen Bildaufnahme, mit Hilfe eines bildgebenden Sensors, sind mit Bildverarbeitungssystemen Positions- und Orientierungsberechnungen mit den erkannten Markern möglich [1]. Für das markerbasierte Trackingverfahren kommen reflektierende Marker und Marker mit Muster zum Einsatz [1].



### Reflektierende Marker

Viele der reflektierenden Marker bestehen an der Oberfläche aus einem retroreflektierendem Material. Bei einer Bildaufnahme können mit Hilfe von Beleuchtungsquellen die reflektierenden Marker im Bild leicht wiedererkannt werden. Für die reflektierenden Marker werden häufig Kugeln mit einer beliebigen Größe verwendet [1].

Mit mehreren bildgebenden Sensoren werden für eine dreidimensionale Positionsermittlung der reflektierenden Marker zunächst Bildaufnahmen der beleuchteten Umgebung erstellt. Die Bildaufnahmen werden im Anschluss mit einem geeigneten Detektor der Bildverarbeitung auf helle Bildbereiche untersucht, die durch die reflektierenden Marker zustande kommen. Mit den erkannten Markern im Bild ist mittels der Photogrammetrie eine genaue Berechnung der Raumposition möglich [1].

Um zur Position zusätzlich Rotationen zu ermitteln, werden Baummarker eingesetzt, die aus einer festen Struktur mit mehreren Markerkugeln bestehen. Die Abbildung 2.1 stellt einen einfachen Baummarker dar [1].

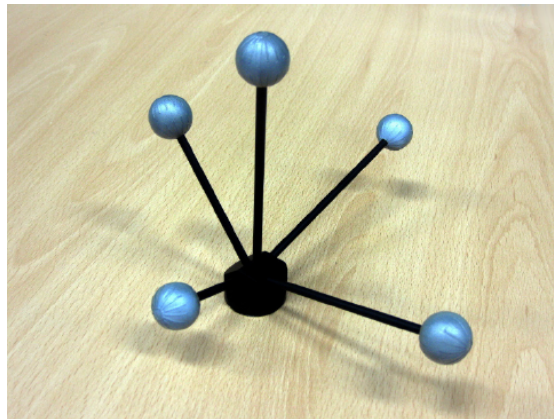


Abbildung 2.1.: Baummarker mit retroreflektiver Oberfläche

Mit den definierten Kugelabständen können Drehungen im Raum bestimmt werden. Durch die Berechnung der dreidimensionalen Position und der Rotation sind 6 Freiheitsgrade realisierbar. Im Allgemeinen wird die berechnete Position und Orientierung zusammenfassend als Pose bezeichnet. Die einfache Struktur der reflektierenden Marker ermöglicht zwar eine robuste Erkennung der Marker im Bild, kann jedoch bei symmetrischer Struktur und einer geringen Anzahl der Markerkugeln die Poseermittlung durch gegenseitige Verdeckung der Markerkugeln erschweren. Ein weiteres Problem entsteht bei der Anbringung von Markern, die auf starken Unebenheiten schlecht platziert werden können. [1]

### Marker mit Muster

Für die Erkennung der Marker mit Muster werden zweidimensionale Muster verwendet, wie beispielsweise der Flachmarker in Abbildung 2.2 zeigt. Für die Muster werden verschiedene rotations-symmetriefreie geometrische Formen eingesetzt, um kantenartige Strukturen zu erzeugen. Häufig aufzufindende geometrische Formen in Anwendungen sind somit rechteckige oder quadratische Muster. Für die Bestimmung der Pose wird zunächst die Umgebung aufgenommen. Mit einem Bild-verarbeitungssystem werden die Bildaufnahmen nach Kanten untersucht, die durch starke Farb- und Helligkeitsunterschiede gekennzeichnet sind und zwischen dem Muster und der Umgebung auftreten. Aus der Kantenlänge, den Eckpunkten der zusammenlaufenden Kanten und dem Muster im inneren Bereich, kann die Pose mit der Photogrammetrie ermittelt werden. Die ausführliche Berechnung der Pose kann in [2] nachgeschlagen werden.



Abbildung 2.2.: Flachmarker mit zweidimensionalem Muster [1]

Durch Kombination von 6 Flachmarkern lässt sich ein Markerwürfel konstruieren, der es ermöglicht, aus jeder Perspektive im Raum diesen zu erkennen. Marker mit Muster liefern ein einfaches Trackingverfahren, das mit dem Bildverarbeitungssystem diese in Bildaufnahmen wiedererkennt [1].

Zu den bekanntesten Vertretern der markerbasierten Trackingverfahren gehören das ARStudio, das ARToolKit, das ARToolKitPlus, das Studierstube-Framework und das ARTag. Hierbei verwenden das ARStudio und das ARToolKit als Flachmarker Binärbilder [Abbildung 2.3] [2]. ARToolKitPlus und ARTag nutzen  $6 \cdot 6$  große Bitfelder zur leichten Wiedererkennung der Muster und das Studierstube-Framework Farbmuster, die Kanten durch Helligkeitsunterschiede der Farben auffinden [Abbildung 2.4] [2].

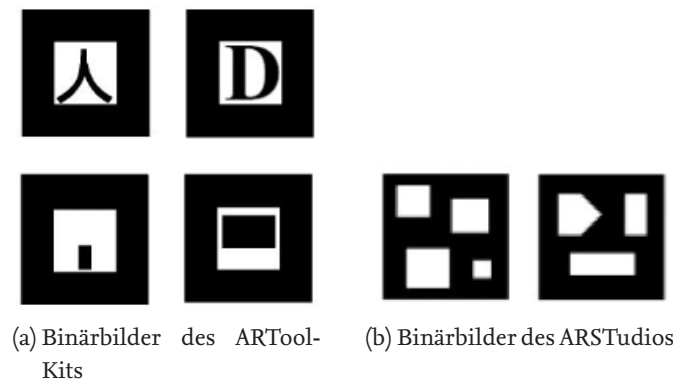


Abbildung 2.3.: Binärbilder als Flachmarker der markerbasierten Trackingverfahren  
[2]

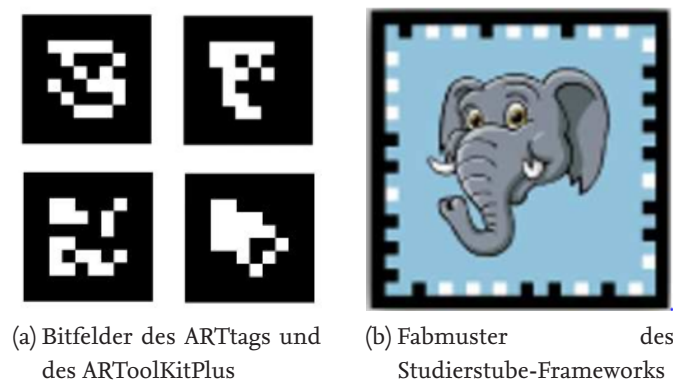


Abbildung 2.4.: Bitfelder und Farbmuster als Flachmarker der markerbasierten Trackingverfahren  
[2]

### 2.1.2. Markerlose Trackingverfahren

Für das markerlose Tracking bestehen vielfältige Trackingverfahren. Die Trackingalgorithmen unterscheiden sich in ihren Berechnungsmethoden, die im Nachfolgenden näher beschrieben werden. Im Allgemeinen suchen die Trackingalgorithmen nach natürlichen Interessensmerkmalen im Bild [1].

Beim markerlosen Tracking unterscheidet man zwischen modellbasierten und merkmalsbasierten Bildverarbeitungsverfahren. Im modellbasierten Bildverarbeitungsverfahren werden Bildsynthesen aus beispielsweise gerenderten 3D-Modellen erzeugt, die als Referenzmodell für die Erkennung der Interessensmerkmale im Bild dienen. Mit dem merkmalsbasierten Bildverarbeitungsverfahren werden ausschließlich die Interessensmerkmale selbst für die Verfolgung in Bildsequenzen verwendet [1].

Viele der merkmalsbasierten Bildverarbeitungsverfahren bestehen aus folgenden wesentlichen Bestandteilen, den Detektoren, den Deskriptoren und der Korrespondenzsuche. Die Detektoren identifizieren zunächst Interessensmerkmale in einer der Bildsequenzen. Um die gefundenen Interessensmerkmale voneinander zu unterscheiden, werden Deskriptoren zur Beschreibung der Interessensmerkmale ihrer Eigenschaften verwendet. Die Wiederfindung der Interessensmerkmale in nachfolgenden Bildern wird mit der Korrespondenzsuche ermöglicht.

Im Folgenden werden zunächst die einzelnen Bestandteile der Trackingalgorithmen und im Anschluss die für die Feature Tracking Simulation relevanten Trackingalgorithmen vorgestellt.

## 2.2. Detektoren

Strukturelle Formen im Bild, wie beispielweise Ecken, Kanten oder besondere Konturen sind durch starke Helligkeitsunterschiede geprägt und haben für die Erkennung von robusten Interessensmerkmalen eine besondere Bedeutung. Es besteht eine Vielzahl an Detektoren, die sich in 4 Bereiche kategorisieren lassen und im Folgenden vorgestellt werden.

### 2.2.1. Kantendetektoren

Kanten beschreiben auf kleinem Raum und entlang einer ausgeprägten Richtung Intensitätsunterschiede im Bild. Eine robuste Kante zeichnet sich durch große Helligkeitsunterschiede aus, die sich mathematisch durch Grauwertfunktionen der Bildebene ausdrücken lassen. Die Intensitätsstärke, die ein Maß der Kantenstärke ist, lässt sich durch Ableiten der Grauwertfunktionen  $\frac{\delta I}{\delta x}(u, v)$  und  $\frac{\delta I}{\delta y}(u, v)$  der Bilder nach den jeweiligen Koordinatenrichtungen ermitteln. Ermittelte Extremwerte der ersten Ableitung sind charakteristisch für Kanten, die auf große Helligkeitsänderungen im Bild zurückzuführen sind [3].

Für eine mathematische Beschreibung eines Bildes werden die konstanten Bildgrauwerte aus der Pixelmatrix, die das Bild aufbauen, verwendet. Aus den Bildwerten lässt sich anschließend eine diskrete Grauwertfunktion bestimmen [3]. Die erste Ableitung der Grauwertfunktion ist eine Approximation der diskreten Differenzen und bestimmt die Steigung eines Pixels. Die Betrachtung der benachbarten Bildpunkte ergibt die resultierende Steigung der Bildfunktion an dem ausgewählten Bildpunkt. Aus den Steigungen  $\frac{df}{dx}(u)$  der Bildreihe und der Bildspalte aus Gleichung 2.1 lässt sich darauffolgend der Gradientenvektor  $\nabla I(u, v)$  bestimmen, der in Gleichung 2.2 aufgeführt und in Abbildung 2.5 dargestellt ist [3]. Für das Aufstellen von mehrdimensionalen Grauwertfunktionen in Gleichung 2.2, werden partielle Ableitungen gebildet.

$$\frac{df}{dx}(u) \approx \frac{f(u+1) - f(u-1)}{(u+1) - (u-1)} \quad (2.1)$$

$$\nabla I(u, v) = \begin{pmatrix} I_x(u, v) \\ I_y(u, v) \end{pmatrix} = \begin{pmatrix} \frac{\delta I}{\delta x}(u, v) \\ \frac{\delta I}{\delta y}(u, v) \end{pmatrix} \quad (2.2)$$

$$|\nabla I(u, v)| = \sqrt{I_x^2(u, v) + I_y^2(u, v)} \quad (2.3)$$

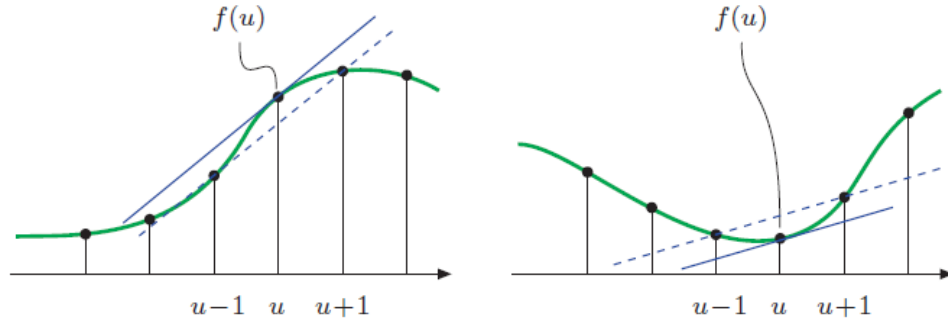


Abbildung 2.5.: Steigung einer Bildfunktion [3]

Mit dem Betrag des Gradientenvektors in Gleichung 2.3 ist die Anstiegsstärke der Funktion berechenbar, die eine Aussage über die Kantenstärke trifft. Durch die Betragsbildung des Gradientenvektors entsteht eine Invarianz gegenüber einer Bilddrehung, die infolgedessen zur Berechnung der Kantenstärke keinen Einfluss durch Rotationen hat. Dies ist eine wichtige Bedingung der isotropen Kantendetektion [4]. Aus dem Anstieg wird anschließend in Gleichung 2.4 der Richtungswinkel des Gradientenvektors bestimmt.

$$\phi(u, v) = \arctan \left( \frac{I_y(u, v)}{I_x(u, v)} \right) \quad (2.4)$$

Die approximierten ersten Ableitungen der Grauwertfunktionen, liefern Schätzungen der Gradientenvektoren in einem beliebigen Bildpunkt, die sich im Anschluss durch lineare Filteroperatoren in eine Matrix übertragen lassen. Aus den erstellten Matrizen lässt sich durch partielle Ableitungen die Gleichung 2.5 aufstellen [5]. Hierbei wird mit den Bildmatrizen  $\underline{\underline{I_x}}$  und  $\underline{\underline{I_y}}$  in Gleichung 2.5, die aus einer Nachbarschaft von Bildpunkten bestehen mit den reellwertigen Matriceinträgen  $\underline{\underline{H_x}}$  und  $\underline{\underline{H_y}}$  gewichtet. Durch die anschließende Betragsbildung der gewichteten Bildmatrizen  $\underline{\underline{\tilde{I_x}}}$  und  $\underline{\underline{\tilde{I_y}}}$  können Kantenbilder bestimmt werden [5].

$$\underline{\underline{\tilde{I_x}}} = \underline{\underline{I_x}} \cdot \underline{\underline{H_x}} \quad ; \quad \underline{\underline{H_x}} = \begin{pmatrix} 0 & 0 & 0 \\ -0,5 & 0 & 0,5 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{und} \quad \underline{\underline{\tilde{I_y}}} = \underline{\underline{I_y}} \cdot \underline{\underline{H_y}} \quad ; \quad \underline{\underline{H_y}} = \begin{pmatrix} 0 & 0,5 & 0 \\ 0 & 0 & 0 \\ 0 & 0,5 & 0 \end{pmatrix} \quad (2.5)$$

Zu den bekanntesten Filteroperatoren der Kantendetektoren gehören die Sobel und Prewitt-Operatoren. Generell unterscheiden sich die Filteroperatoren in ihren Gewichtungen der Matriceinträge, die zu einer verändernden Isotropie herbeiführen. Die Isotropie der Kantendetektion beschreibt hierbei die Empfindlichkeit der Kantenintensität durch Bildrotationen [6]. Zur Bestimmung von lokalen Extrempunkten der Grauwertfunktionen werden die zweiten Ableitungen mit Laplacefiltern bestimmt, die anstatt der Graubildstufen [Abbildung 2.6] punktförmige Bildstrukturen hervorrufen [Abbildung 2.7][6].

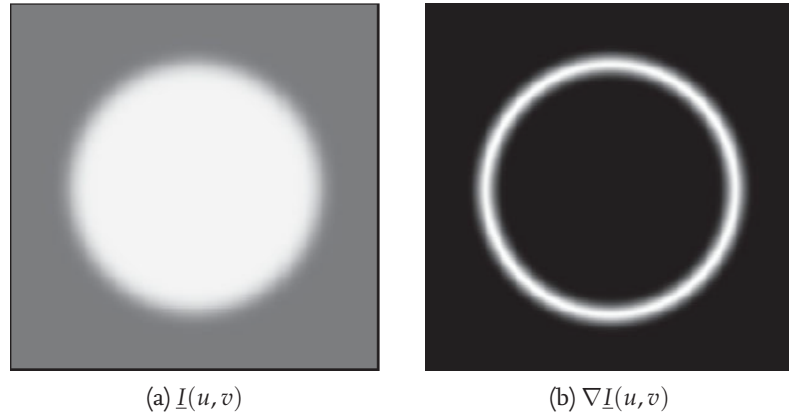


Abbildung 2.6.: Detektierte Kante eines Graubildes mit erster partieller Ableitung [5]

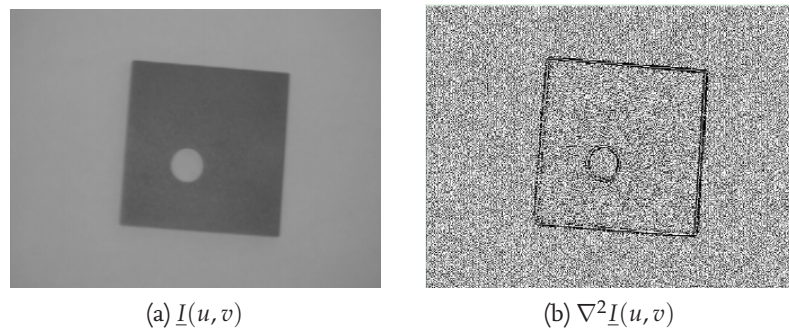


Abbildung 2.7.: Detektierte Kante eines Graubildes mit zweiter partieller Ableitung [6]

### 2.2.2. Eckendetektor

Aus der bereits ausführlich beschriebenen Kantendetektion, nutzen auch Eckendetektoren Grauwertfunktionen und deren approximierte Ableitungen zur Ermittlung von Ecken.

Der wesentliche Unterschied zur Kantendetektion besteht bei der Richtung der Intensitätsunterschiede. Ausgeprägte Ecken weisen nicht nur starke Intensitätsunterschiede in einer Richtung auf, wie dies bei Kanten der Fall ist. Für die Eckendetektion ist somit der Intensitätsunterschied bzw. sind die Gradientenwerte in mehreren Richtungen festzustellen. Für die Eckenbestimmung besteht eine Vielzahl an Filteroperatoren. Ein repräsentatives und häufig aufzufindendes Beispiel ist der sogenannte Harris-Eckendetektor [7].

Der Harris-Eckendetektor findet Interessensmerkmale mit Grauwertecken an komplexen Grauwertübergängen.

Für jeden Bildpunkt wird die Strukturmatrix in Gleichung 2.6 mit den partiellen Ableitungen der lokalen Grauwertfunktionen beschrieben. Eine Ecke kennzeichnet zwei Kanten, die im Bild zusammenlaufen [7].

$$S = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (2.6)$$

Zur Erkennung von robusten Ecken, müssen Ecken, die durch Störeinflüsse wie Bildrauschen beeinflusst werden, unterdrückt werden. Die Ecken werden durch die Eigenwerte der Hesse-Matrix mit Gleichung 2.7 bestimmt und anschließend miteinander verglichen. Potenzielle Ecken besitzen positive Eigenwerte. Mit positiven Eigenwerten ist davon auszugehen, dass die Spur der Strukturmatrix ebenfalls positiv ist und somit zur Bedingung  $\lambda_1 \geq \lambda_2$  beiträgt [7].

$$\lambda_{1,2} = \frac{\text{spur}(\underline{\underline{S}})}{2} \pm \sqrt{\left(\frac{\text{spur}(\underline{\underline{S}})}{2}\right)^2 - \det(\underline{\underline{S}})} \quad (2.7)$$

Für die anschließende Differenzbildung der Eigenwerte in Gleichung 2.8, ist ein möglichst kleiner Differenzwert erstrebenswert, um robuste Ecken erkennen zu können. Aus Abbildung 2.8 wird ersichtlich, dass für eine robuste Eckendetektion etwa gleich große Eigenwerte erforderlich sind, um minimale Differenzen zu erhalten [7].

$$\lambda_1 - \lambda_2 = 2 \cdot \sqrt{\left(\frac{\text{spur}(\underline{\underline{S}})}{2}\right)^2 - \det(\underline{\underline{S}})} \quad (2.8)$$

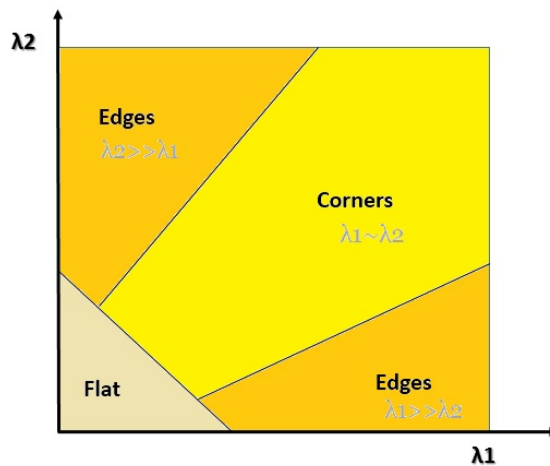


Abbildung 2.8.: Eckendetektion durch Vergleich der Eigenwerte



Mit dem Funktionswert aus der Gleichung 2.9 wird die Eckenstärke bestimmt, die einen vordefinierten Schwellwert überschreiten muss. Die Variable  $k$  stellt hierbei die Empfindlichkeit des Eckendetektors ein [7].

$$C(x, y) = \det \left( \underline{\underline{S}} \right) - k \cdot \text{spur} \left( \underline{\underline{S}} \right)^2 \quad (2.9)$$

### 2.2.3. Blobdetektor

Interessensmerkmale können gefunden werden, wenn in einem kleinen Bildbereich sogenannte Blobs zu finden sind. Blobs besitzen, wie in Abbildung 2.9 zu erkennen ist, in kleinen Bildbereichen ähnliche Bildwerte und unterscheiden sich mit der Bildumgebung durch starke Helligkeitsunterschiede.

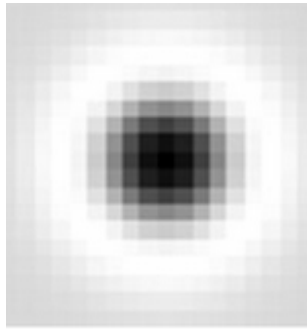


Abbildung 2.9.: Blob als Interessensmerkmal

Zur Ermittlung der markanten Bildpunkte mit dem Blobdetektor wird in ähnlicher Weise wie der Harris-Eckendetektor die Hessematrix genutzt [8].

Die Elemente der Hessematrix beinhalten hierbei die geglätteten Grauwertfunktionen. Die nachfolgende Determinante der Hessematrix detektiert Blobs als lokale Maxima und Minima [8].

Ein weiteres häufig verwendetes Verfahren ist die Suche nach lokalen Maxima mit Hilfe der rechenintensiven Laplacian-of-Gaussian (LoG). Für die Reduktion des Rechenaufwandes werden häufig die LoG-Filter durch andere Filter approximiert [8].

### 2.2.4. Ridgedetektor

Dieses Verfahren ermittelt Bildlinien, die verschiedene Formen annehmen können. Die Bildlinien bestehen aus einer Anzahl aneinandergereihter Bildpunkten mit lokalen Maxima. Lokale Minima werden bei den Ridgedetektoren als Valleys (Täler) bezeichnet [9]. Für das Verfahren zur Ermittlung von Bildlinien wird ähnlich wie beim Kantendetektor vorgegangen. Hierbei beschränkt sich die Suche lediglich auf große Haupteigenwerte im Bild, die wie aus Abbildung 2.10 hervorgeht Bildlinien erzeugen [9].

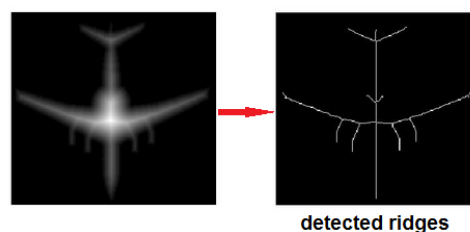


Abbildung 2.10.: Detektion von Bildlinien eines Flugzeuges [10]

## 2.3. Deskriptoren

Deskriptiv arbeitende Trackingverfahren beschreiben die detektierten Interessensmerkmale nach ihren Bildeigenschaften. Um optimale Bildeigenschaften zu erhalten sollten Deskriptoren möglichst robust gegenüber Bildrauschen, perspektivischen Veränderungen und invariant gegenüber Skalierung, Rotation und Helligkeitsänderungen sein [11].

Deskriptoren lassen sich in verteilungsbasierte, raumfrequenztechnische und differentialbasierte Deskriptoren kategorisieren. Die verteilungsbasierten Deskriptoren nutzen zur Charakteristik der Interessensmerkmale Histogramme, die eine Verteilung der Grauwerte der Bildpunkte darstellt. Häufig wird ein Histogramm auch für die Anordnung und Beziehung zwischen den Grauwerten der Bildpunkte genutzt [12].

Raumfrequenztechnische Deskriptoren nutzen gefensterte Fourier Transformationen, die Frequenzen- und Orientierungsänderungen zwischen benachbarten Bildpunkten bestimmen.

Die differentialbasierten Deskriptoren basieren auf lokalen Ableitungen der Grauwertfunktionen, die alle räumlichen und partiellen Ableitungen in einem Interessensmerkmal betrachten. Ein Beispiel ist der steuerbare Filter, der die Ableitungen in eine bestimmte Richtung lenkt [12].

### 2.3.1. SIFT-Deskriptor

Zur Beschreibung der Interessensmerkmale verwendet der SIFT-Deskriptor Differenzbilder, die sogenannten „Difference of Gaussian“, die sich aus der Differenz der benachbarten Gauß-Ebenen in der Gauß-Pyramide ergeben [Abbildung 2.11].

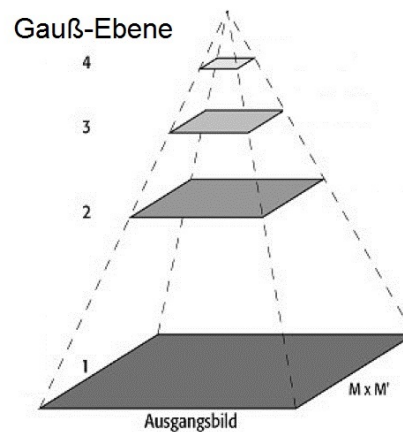


Abbildung 2.11.: Gauß-Pyramide einer jeweiligen Multiskalen-Oktave

Aus einem gegebenen Eingangsbild, das als Grauwertfunktion beschrieben wird, werden zunächst mehrere sogenannte Multiskalen-Oktaven gebildet. Eine jeweilige Multiskalen-Oktave besteht aus einer Gauß-Pyramide, die ihre pyramidenartige Struktur durch die Skalierung des aufgelösten Grauwertbildes mit der Variable  $k$  in Gleichung 2.10 erhält. Die Gauß-Ebenen in der Gauß-Pyramide werden von der Bildgröße des Skalierungsfaktors  $k$  abhängig gemacht. Als Beispiel ist in Abbildung 2.12 die Gauß-Pyramide dargestellt [7].

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) \cdot I(x, y) \quad (2.10)$$

mit:  $G(x, y, \sigma) = \frac{1}{2\pi\sigma} \cdot e^{-\frac{x^2 + y^2}{2\sigma^2}}$

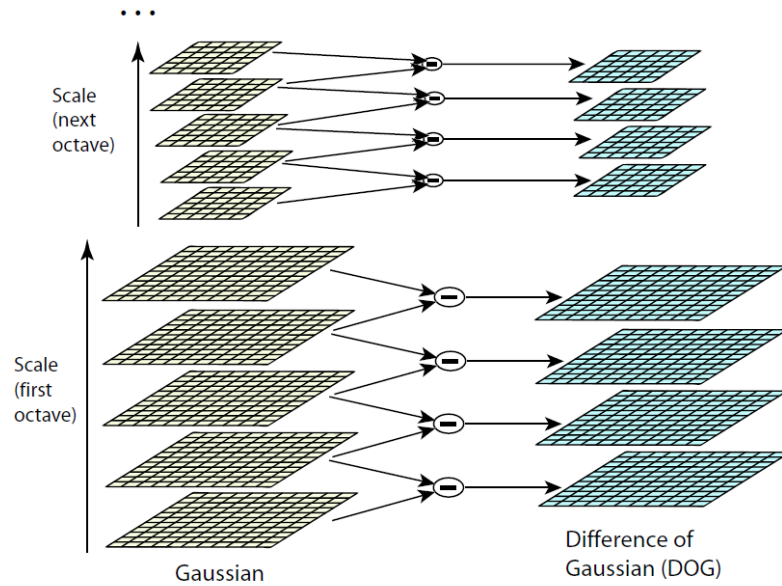


Abbildung 2.12.: Erstellen des Multiskalenraums [7]

Für die Orientierung des zu beschreibenden Interessensmerkmals im jeweiligen Differenzbild, werden Orientierungshistogramme verwendet. Diesbezüglich wird um das Interessensmerkmal eine Umgebung der Größe  $1,5 \cdot \text{Sigma}$  betrachtet und nach dominierenden Gradientenrichtungen, die durch Grauwertänderungen zustande kommen, bestimmt. Die Betrachtung der Grauwertänderung erzielt hierbei eine Invarianz der Orientierung [7].

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2.11)$$

mit:  $L(x, y) = G(x, y, \sigma) * I(x, y)$

$$\phi = \arctan \left[ \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right] \quad (2.12)$$

Die Gleichung 2.11 bestimmt den Gradientenbetrag  $m(x, y)$  des Grauwertes und die Gleichung 2.12 die Gradientenrichtung  $\phi$ , die anschließend im Histogramm dargelegt werden. Ein Beispiel zeigt die Abbildung 2.13, die mit einer Unterteilungsschrittweite von 10 Grad das Orientierungshistogramm mit insgesamt 8 Klassen darstellt. Die jeweiligen Klassen bilden die Summe der Gradientenbeträge der Umgebungsmerkmale, die bei Überschreiten einer vordefinierten Häufigkeitsdichte die dominierende Orientierung des Extremwertes angibt [7].

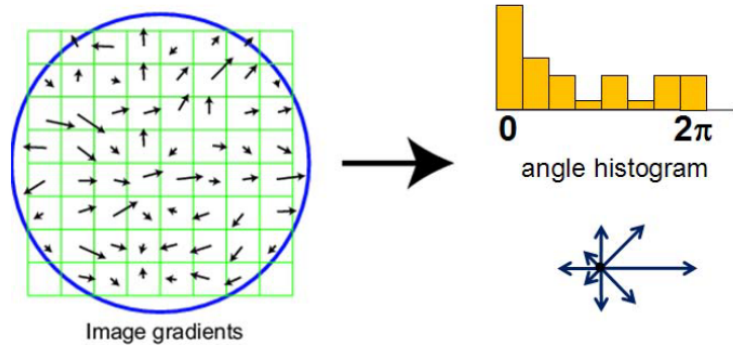


Abbildung 2.13.: Bestimmen der Gradientenrichtung [7]

### 2.3.2. SURF-Deskriptor

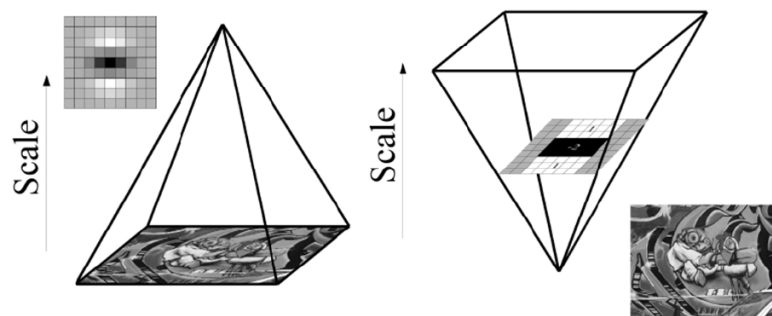


Abbildung 2.14.: Berechnung eines Integralbildes [13]

Mit dem SURF-Deskriptor wird, wie aus Abbildung 2.14 hervorgeht, im Gegensatz zum SIFT-Deskriptor das Eingangsbild durch entsprechende Filtermasken vergrößert. Die Integralbilder aus der Vergrößerung des Eingangsbildes bilden die Skalierungsebenen einer jeweiligen Oktave ab. Analog zum SIFT-Deskriptor können hierbei mehrere Oktaven bestehen, aus denen subpixelgenaue Positionen der Interessensmerkmale im Eingangsbild bestimmt werden können. Zur Detektion von Interessensmerkmalen werden die verschiedenen Skalierungsebenen nach besonderen Bildbereichen, den sogenannten „Blobs“ aus Abschnitt 2.2.3 untersucht [14]. Die Berechnung der Integralbilder ist in Gleichung 2.13 vermerkt.

$$I_{Integral}(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j) \quad (2.13)$$

Durch Aufsummieren der Grauwerte zwischen dem Bildpunkt des Bildausschnitts und des Eingangsbildes entstehen Integralbilder. Ein einfaches Beispiel illustriert die Abbildung 2.15 für die in Gleichung 2.14 demonstrativ ein Integralbild berechnet wird [15].

$$I_{Integral}(x, y) = \sum_{i=A_x}^{i \leq D_x} \sum_{j=A_y}^{j \leq D_y} I(i, j) = I_{Integral}(D) - I_{Integral}(B) - I_{Integral}(C) + I_{Integral}(A) \quad (2.14)$$

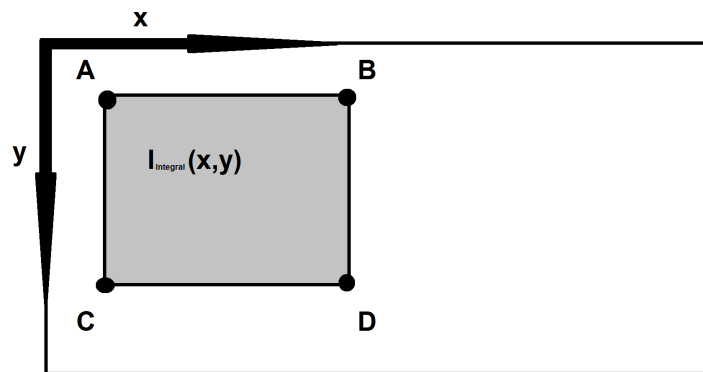


Abbildung 2.15.: Berechnung eines Integralbildes

Zur Orientierungsbeschreibung der Interessensmerkmale werden Haar-Wavlet-Filter auf Integralbilder angewendet, die einen Vergleich der Grauwertintensitäten der benachbarten Bildpunkte erlauben. Der einfache Haar-Wavlet-Filter in Abbildung 2.16 untersucht Bildgradienten, die ausschließlich in den Koordinatenrichtungen verlaufen. Dies ermöglicht eine einfache Berechnung der Grauwertgradienten.

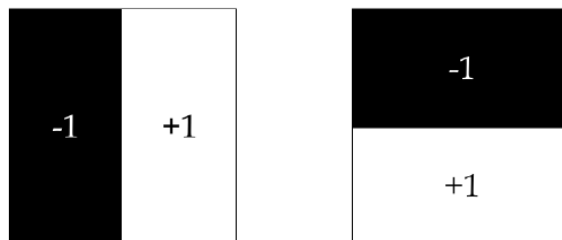


Abbildung 2.16.: Haar-Wavlet-Filter [15]

### 2.3.3. BRISK-Deskriptor

Auch das BRISK-Trackingverfahren verwendet für eine zuverlässige Erkennung der Interessensmerkmale skalierte Bildebenen, die für Skalen- und Rotationsinvarianz sorgen und eine subpixelgenaue Positionsermittlung der Interessensmerkmale ermöglichen.

Um skalierungsinvariante Interessensmerkmale zu erhalten, werden die skalierten Bildebenen verwendet, die aus Oktaven und Suboktaven bestehen. Mit steigender Oktavenzahl wird das Eingangsbild wie beim SIFT-Algorithmus verkleinert [16]. Eine detaillierte Beschreibung der subpixelgenauen Positionsbestimmung im Skalenraum ist in Abschnitt 2.8. zu finden.

### 2.3.4. Weitere Deskriptoren

Einige weiterentwickelte Deskriptoren orientieren sich nach dem SIFT-Deskriptor, wie zum Beispiel der GLOH-Deskriptor (Gradient Location and Orientation Histogram), der eine größere Umgebung für die Orientierungshistogramme betrachtet und zu einer Verbesserung der Eigenschaftsbeschreibung der Interessensmerkmale beiträgt.

Des Weiteren bestehen Kombinationen aus verschiedenen Deskriptoren, die die Vorteile der jeweiligen Deskriptoren nutzen.

## 2.4. Korrespondenzen

Das Wiederfinden von Interessensmerkmalen in nachfolgenden Bildern wird als Korrespondenz verstanden. Mit einem Ähnlichkeits- oder Korrelationsmaß können Interessensmerkmale miteinander verglichen werden, die bei einer Übereinstimmung von zwei Interessensmerkmalen zu einer Korrespondenz führen.

Grundsätzlich wird ein Bildausschnitt aus einem Referenzbild  $R(i, j)$  im nachfolgenden Zielbild  $I(r + i, s + j)$  schrittweise verschoben, um korrespondierende Interessensmerkmale zu identifizieren.

### 2.4.1. Ähnlichkeitsmaße durch Abstandsermittlung

Große Ähnlichkeiten zwischen Interessensmerkmalen sind durch Vergleiche der Charaktereigenschaften zu erreichen.

Das einfachste Verfahren für die Korrespondenzsuche ist die Ermittlung des Abstands zwischen zwei Interessensmerkmalen, die bei kleinem Abstand große Ähnlichkeiten aufweist [3].

Für die Abstandsermittlung der Interessensmerkmale zwischen einem aktuellen  $R(i, j)$  und einem nachfolgenden Bild  $I(r + i, s + j)$ , sind verschiedene Ansätze vorhanden. Die Gleichung 2.15 stellt die euklidische Distanz dar, die im N-dimensionalen Raum die Summe der quadratischen Abstände zwischen zwei Vektoren mit einem Bildwert der Länge N bestimmt. Gleichung 2.16 repräsentiert die Mannheimer-Norm, in der die Summe der Beträge bestimmt wird und Gleichung 2.17 die Maximumnorm, die den maximalen Differenzbetrag berechnet. [3]

Übliche Abstandsbestimmungen bilden die Gleichungen 2.15, 2.16 und 2.17.



$$d_E(r, s) = \sqrt{\sum_{(i,j) \in R} (I(r + i, s + j) - R(i, j))^2} \quad (2.15)$$

$$d_M(r, s) = \sum_{(i,j) \in R} |I(r + i, s + j) - R(i, j)| \quad (2.16)$$

$$d_{Max}(r, s) = \max_{(i,j) \in R} |I(r + i, s + j) - R(i, j)| \quad (2.17)$$

### 2.4.2. Kreuzkorrelation

Die Kreuzkorrelation sucht nach linearen Zusammenhängen zwischen dem Referenz-  $R(i, j)$  und Zielbild  $I(r + i, s + j)$ . Die zu vergleichenden Bildpunkte werden nach ihren Bildgrauwerten untersucht, die abhängig oder unabhängig voneinander sind. Der lineare Zusammenhang ist mit der Summe über die Produkte der Bildpunkte zu bestimmen [17].

Um einen besseren Vergleich zu erhalten, werden die Intensitäten durch die Normierung skaliert. Mit der normalisierten Kreuzkorrelation in Gleichung 2.18, sind positive Werte im Intervall zwischen 0 und 1 zu erreichen.

$$C_N(r, s) = \frac{C(r, s)}{\sqrt{A(r, s) \cdot B}} = \frac{\sum_{(i,j) \in R} I(r + i, s + j) \cdot R(i, j)}{\sqrt{\left[ \sum_{(i,j) \in R} I^2(r + i, s + j) \right] \cdot \left[ \sum_{(i,j) \in R} R^2(i, j) \right]}} \quad (2.18)$$

Eine Übereinstimmung der Bildpunkte schließt auf einen maximalen Wert zurück. Die im Nenner verwendete Variable  $A(r, s)$  liefert die Summe der quadratischen Werte eines Bildausschnittes des Zielbildes  $I(r + i, s + j)$ . Die Variable  $B$  liefert die Summe der quadratischen Werte des Referenzbildes  $R(i, j)$ . Die Summe der Produkte des Referenz-  $R(i, j)$  und Zielbildes  $I(r + i, s + j)$  bestimmen den Korrelationskoeffizienten  $C(r, s)$  [3].

Da jedoch die normalisierte Kreuzkorrelation stark anfällig gegenüber Helligkeitsänderungen im Bild ist, weil durch den Vergleich der ursprünglichen Werte der Bildfunktionen auch der Wert des Korrelationskoeffizienten stark ansteigt, werden für die Berücksichtigung der Helligkeitsänderungen nicht die ursprünglichen Werte betrachtet. Es sind hierbei die Differenzen der arithmetischen Mittelwerte des Referenzbildes  $R(i, j)$  und des Zielbildes  $I(r + i, s + j)$  zu bestimmen, die anschließend vom ursprünglichen Referenzbild  $R(i, j)$  und Zielbild  $I(r + i, s + j)$  in Gleichung 2.18 subtrahiert werden [3].

## 2.5. (KLT) Kanade-Lucas-Tomasi-Algorithmus

Grundsätzlich verfolgt der Kanade-Lucas-Tomasi-Algorithmus Interessensmerkmale in einer Bildfolge durch Transformation des Referenzbildes, die durch Bildbewegungen, dem optischen Fluss, bestimmt wird.

Die Transformation bestimmt die Verschiebung des Interessensmerkmals im Zielbild, die durch Bildänderungen und Bildbewegungen zustande kommen.

Für die Detektion der Interessensmerkmale in der Bildfolge, wird der Shi-Tomasi Detektor verwendet, der eine Erweiterung zum Harris-Eckendetektor in Abschnitt 2.2.2 bildet und für robustere Ecken sorgt. Anders als beim Harris-Eckendetektor entnimmt der Shi-Tomasi Detektor den kleinsten Eigenwert aus der berechneten Hesse-Matrix aus Gleichung 2.6 und vergleicht diesen Eigenwert mit einem vordefinierten Schwellwert [3].

Die detektierten Interessensmerkmale in einem Referenzbild, werden anschließend durch eine Schätzung der Transformationsparameter, die auf Bildbewegung basiert, im Zielbild verschoben. Die Verschiebung ergibt anschließend die neue Position des Interessensmerkmals. Eine vereinfachte Darstellung der Verschiebung ist als eindimensionaler Fall in Abbildung 2.17 dargestellt [18].

Aus den zwei stetigen Funktionen  $f(x)$  und  $g(x)$ , für die ein gleicher Funktionsablauf angenommen wird, lässt sich die Verschiebung  $\bar{t}$  in dem Intervall zwischen  $x_a$  und  $x_b$  auf der Abszisse bestimmen [3].

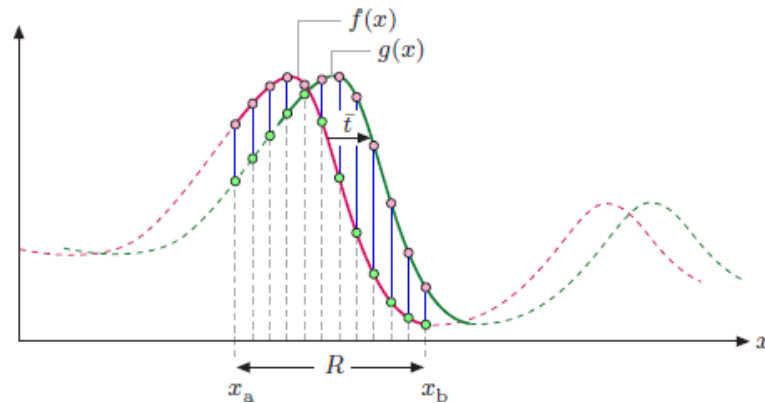


Abbildung 2.17.: Verschiebung im eindimensionalen Fall [3]

Für eine lineare Funktion an einem Bildpunkt auf der Abszisse, kann hierbei mit bekannter Steigung  $f'(x)$  folgende Gleichung 2.19 aufgestellt werden [3].

$$g(x) \approx f(x) - t \cdot f'(x) \quad (2.19)$$

Die geschätzte Verschiebung  $t$  ergibt sich durch Umstellen der Gleichung 2.19, die der Taylor-Entwicklung erster Ordnung in Gleichung 2.20 entspricht [3].

$$t \approx \frac{f(x) - g(x)}{f'(x)} \quad (2.20)$$

Da lineare Funktionen in der Umgebung nicht immer gegeben sind, müssen mehrere Funktionswerte für eine zuverlässige Schätzung der Verschiebung im Bereich  $R$  aus Abbildung 2.17 bestimmt werden. Aus der Ableitung der Verschiebungsfunktion zwischen den Funktionen im Intervall  $R$  lässt sich die minimalste Verschiebung  $t$  bestimmen. Durch Nullsetzen der abgeleiteten Verschiebungsfunktion kann eine optimale Verschiebung in Gleichung 2.21 bestimmt werden [3].

$$t_{optimal} = \left[ \sum_{x \in R} [f'(x)]^2 \right]^{-1} \cdot \sum_{x \in R} f'(x) \cdot [f(x) - g(x)] \quad (2.21)$$

Für die Betrachtung von mehrdimensionalen Fällen, nehmen Positions- und Verschiebungsvektoren  $n$ -dimensionale Größen an. In Gleichung 2.22 ist die optimale Verschiebung für den mehrdimensionalen Fall gegeben [3].

$$t_{optimal} = \left[ \sum_{x \in R} [(\nabla F(x))^T \cdot \nabla F(x)] \right]^{-1} \cdot \left[ \sum_{x \in R} (\nabla F(x))^T \cdot [F(x) - G(x)] \right] \quad (2.22)$$

mit  $\nabla F(x) = \left( \frac{\delta F}{\delta x_1(x)}, \dots, \frac{\delta F}{\delta x_m(x)} \right)$

Der KLT-Algorithmus, der auf dem vorherigen vorgestellten Konzept basiert, berechnet nicht nur optimale Translationen. Durch den  $p$ -dimensionalen Vektor aus Gleichung 2.23 lassen sich verschiedene geometrische Transformationen parametrisieren, die zu einer Positionsänderung und Verzerrung des Bildausschnittes führen. Die Abbildung 2.18 stellt hierbei mögliche zweidimensionale Transformationen für den KLT-Algorithmus vor [3]. Hierbei bestimmt die translatorische Transformation eine einfache Bildverschiebung. Die Euklidische-Transformation ermöglicht Bild-drehungen und Translationen. Zusätzlich kann die Ähnlichkeits-Transformation Bildskalierungen bestimmen. Mit affinen und projektiven Transformationen können kollineare Bilder erzeugt werden [19].

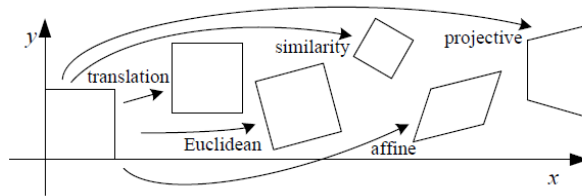


Abbildung 2.18.: Mögliche Transformationen des KLT-Algorithmus [19]

Das Ziel beim KLT-Algorithmus zur Wiederfindung von Interessensmerkmalen ist die Suche nach der minimalsten Fehlergröße  $E_{\text{Fehlergröße}}$  in Gleichung 2.23, die durch Differenzieren des Ziel- $I$  und des Referenzbildes  $R$  ermöglicht wird [3].

$$\begin{aligned}
E_{\text{Fehlergröße}} &= \sum_{x,y \in R} \left[ \underline{I}(\underline{T}_{p+q}(x,y)) - \underline{R}(x,y) \right]^2 \\
\underline{I}(\underline{T}_{p+q}(x,y)) &\approx \nabla \underline{I}(\underline{T}_p(x,y)) + \nabla \underline{I}(\underline{T}_p(x,y)) \cdot \underline{J}_{\underline{T}_p}(x,y) \cdot \underline{q} \\
\underline{J}_{\underline{T}_p}(x,y) &= \begin{bmatrix} \frac{\delta \underline{T}_{x,p}}{\delta p_1}(x,y) & \dots & \frac{\delta \underline{T}_{x,p}}{\delta p_n}(x,y) \\ \frac{\delta \underline{T}_{y,p}}{\delta p_1}(x,y) & \dots & \frac{\delta \underline{T}_{y,p}}{\delta p_n}(x,y) \end{bmatrix} \text{ mit } \underline{p} = (p_1, \dots, p_n)^T
\end{aligned} \tag{2.23}$$

Dabei sind geeignete Größen der Transformationsparameter  $\underline{T}_{p+q}(x,y)$  zu finden, die zu einer minimalen Fehlergröße in Gleichung 2.18 führen und die Übereinstimmung zwischen dem transformierten Referenzbild  $\underline{\tilde{R}}$  und dem Zielbild  $\underline{I}$  im Intervall  $R$  maximiert. Für den Anfang wird ein Startwert für die Transformationsparameter  $\underline{T}_p(x,y)$  festgelegt, der durch iterative Berechnung der Fehlergröße  $E_{\text{Fehlergröße}}$  optimiert wird. Bei Erreichen eines festgelegten Schwellwertes wird der Optimierungsvorgang abgebrochen. Die Abbildung 2.19 stellt die Koordinatentransformation des Referenzbildes  $\underline{R}$  im Zielbild  $\underline{I}$  grafisch dar [3].

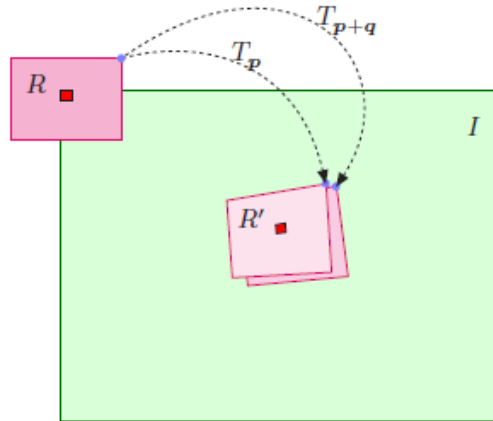


Abbildung 2.19.: Verschieben des Referenzbildes im Suchbild durch Koordinatentransformation [3]

Die ausführliche Fehlergröße des KLT-Algorithmus lässt sich mit den Gradienten der Grauwertfunktionen aus Gleichung 2.2 an einem Bildpunkt aus der Taylor-Entwicklung des Zielbildes und der Jacobimatrix mit der Transformationsfunktion, die mit  $\underline{p}$  Parametern beschrieben wird, bestimmen. Dies wird in Gleichung 2.17 dargestellt [3].

Durch Nullsetzen und Umstellen der Gleichung 2.24 sind schlussendlich optimale Transformationsparameter berechenbar. Die Hessematrix  $\underline{\underline{H}}$  ist für die Schätzung der Transformationsparameter zuständig. Der Zeilenvektor  $\underline{S}(x, y)$  zu dessen Elementen jeweils ein Parameter zugewiesen wird, entspricht der Änderung des Transformationsparameters für jede Position im vordefinierten Intervall. Die Funktion  $\underline{D}(x)$  bestimmt das resultierende Fehlerbild für jede Position. Der resultierende Parameter  $\underline{q}_{optimal}$  in Gleichung 2.25 ist durch die Iteration zu aktualisieren, bis der optimale Parameter ein Konvergenzlimit unterschreitet [3].

$$\underline{q}_{optimal} = \underline{\underline{H}}^{-1} \cdot \left[ \sum_{x,y \in R} \underline{S}^T(x, y) \cdot \underline{D}(x, y) \right] = \left[ \underline{S}^T(x, y) * \underline{S}(x, y) \right]^{-1} \cdot \left[ \sum_{x,y \in R} \underline{S}^T(x, y) \cdot \underline{D}(x, y) \right]$$

$$\text{mit: } \underline{S}(x, y) = \nabla \underline{I}(x + i, y + j) \cdot \underline{J}_{\underline{T}_p}(x, y) \quad ; \quad \underline{D}(x, y) = \underline{R}(x, y) - \underline{I}(x + i, y + i) \quad (2.24)$$

$$\underline{p}_n = \underline{p}_{n-1} + \underline{q}_{optimal} \quad : \quad \underline{q}_{optimal} \Rightarrow ||\underline{q}_{optimal}||_2 < \tau \quad (2.25)$$

## 2.6. (SIFT) Scale-Invariant-Feature-Transform-Algorithmus

Für das Identifizieren von besonderen Interessensmerkmalen sucht der SIFT-Algorithmus zunächst nach markanten Bildbereichen, den sogenannten Blobs aus Abschnitt 2.2.3. in Abbildung 2.9. Zur Beschaffenheit der besonderen Interessensmerkmale werden die markanten Bildbereiche auf mehrere Bildebenen skaliert und die Grauwertrichtungen berechnet. Das Ziel des SIFT-Algorithmus besteht hierbei auf einer robusten Wiedererkennung der Interessensmerkmale anhand ihrer mitgelieferten Eigenschaften in einer Bildfolge. Besonders gegenüber Rotationen, Translationen, Skalierungen und überwiegenden Helligkeitsänderungen ist der SIFT-Algorithmus invariant [7].

Für die Detektion von Interessensmerkmalen wird das Verfahren „Difference of Gaussian“ aus Abschnitt 2.3.1. eingesetzt, um Extremwerte im Multiskalenraum zu erkennen. Der verwendete Multiskalenraum stellt eine Grundlage für die Detektion der besonderen Interessensmerkmale dar. Darauf folgend wird zu jeder Oktave die „Difference of Gaussian“ aus Gleichung 2.10 angewandt, die aus den benachbarten Gauß-Schichten Differenzbilder erzeugt. Aus den Differenzbildern werden anschließend die Extremwerte der Bildpunkte bestimmt, die im Multiskalenraum Minima oder Maxima annehmen können.

Um die Extremwerte zu ermitteln, werden alle benachbarten Werte der Bildpunkte im Multiskalenraum miteinander verglichen. In Abbildung 2.20 sind 3 Differenzbilder dargestellt. Der Wert im mittleren Differenzbild wird insgesamt mit seinen 26 Nachbarwerten aus dem obersten und untersten Differenzbild verglichen. Extremwerte aus dem obersten und untersten Differenzbild reichen für einen robusten Vergleich mit einem Differenzbild nicht aus, da zum Vergleichen nur eine geringe Anzahl an Werten vorhanden ist. Der Vergleich der Extremwerte mit seinen Nachbarwerten findet daher in jeder Oktave statt [7].

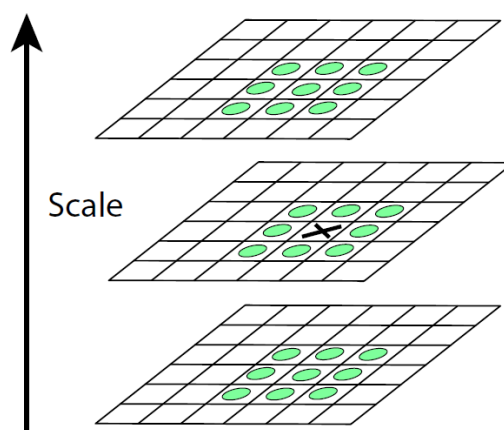


Abbildung 2.20.: Vergleich der Nachbarwerte in Differenzbildern [7]

Für eine subpixelgenaue Positionsermittlung des Bildpunktes, werden mit der Taylorreihenentwicklung der Differenzbilder der benachbarten Gauß-Schichten in Gleichung 2.26 Extremwerte aus dem Multiskalenraum berechnet [7].

$$D(\underline{x}) = D + \left( \frac{\delta D}{\delta \underline{x}} \right)^T \underline{x} + \frac{1}{2} \underline{x}^T \left( \frac{\delta^2 D}{\delta \underline{x}^2} \right)^T \underline{x} \quad ; \quad \underline{x} = [x, y, \sigma]^T \quad (2.26)$$

Das  $\sigma$  bestimmt hierbei die Stärke des Tiefpassfilters. Der Vektor  $\underline{x}$  in Gleichung 2.27 bestimmt die Verschiebung, die für eine subpixelgenaue Ermittlung der Position im Eingangsbild notwendig ist [7].

$$\underline{x} = \left( \frac{\delta^2 D}{\delta \underline{x}^2} \right)^{-1} \frac{\delta D}{\delta \underline{x}} \quad (2.27)$$

Um mögliche Helligkeitsänderungen zu berücksichtigen, werden die Grauwerte zunächst normiert und mit einem vordefinierten Konvergenzlimit verglichen. In [7] ist zur Unterdrückung von geringen Kontrasten im lokalen Bereich ein Konvergenzlimit von  $|D(\underline{x})| < 0.03$  festgelegt.

Die detektierten Interessensmerkmale in den Differenzbildern aus Abbildung 2.12 reagieren jedoch sehr stark auf Kanten, die im nächsten Schritt auf Grauwertecken mit dem Harris-Eckendetektor aus Abschnitt 2.2.2. überprüft werden. Potenzielle Interessensmerkmale bestehen aus Grauwertecken, die in etwa gleich große Eigenwerte der Hesse-Matrix besitzen [7].

Um die orientierten Interessensmerkmale mit dem Orientierungshistogramm aus Abschnitt 2.3.1. eindeutig identifizieren zu können, folgt im nächsten Schritt eine Untersuchung der Nachbarschaft um das Interessensmerkmal. Die Umgebung um das zu beschreibende Interessensmerkmal, ist in ein  $16 \cdot 16$  großes Fenster auf der Schicht in der Gauß-Pyramide eingeteilt, das erneut in ein  $4 \cdot 4$  großes Subfenster mit jeweils  $4 \cdot 4$  große Pixel zerlegt ist. Für die Pixel im Subfenster werden die Beträge und Richtungen des Grauwertgradienten bestimmt. Hierbei werden die Gradientenrichtungen in Richtung der zuvor ermittelten Orientierung ausgerichtet, um eine Rotationsinvarianz der Gradientenrichtungen zu erreichen [7].

Die ausgerichteten Gradientenrichtungen in einem Subfenster werden in einem weiteren Orientierungshistogramm eingetragen. Die Häufigkeitsdichte des Orientierungshistogramms ist durch die Gewichtung der Gradientenbeträge bestimmt, die vom Abstand des Interessens- und Bildpunktes in der Umgebung abhängt. Bildpunkte, die nahe dem Interessensmerkmal liegen, beeinflussen am stärksten die Häufigkeitsdichte. Aus dem ursprünglich  $16 \cdot 16$  großen Fenster sind  $4 \cdot 4$  große Orientierungsfenster zu gewinnen [Abbildung 2.21]. Die berechneten Werte aus den Orientierungshistogrammen werden in einem mehrdimensionalen Vektor beschrieben. Insgesamt besteht für das gesamte Fenster der mehrdimensionale Vektor aus 128 Einträgen [7].



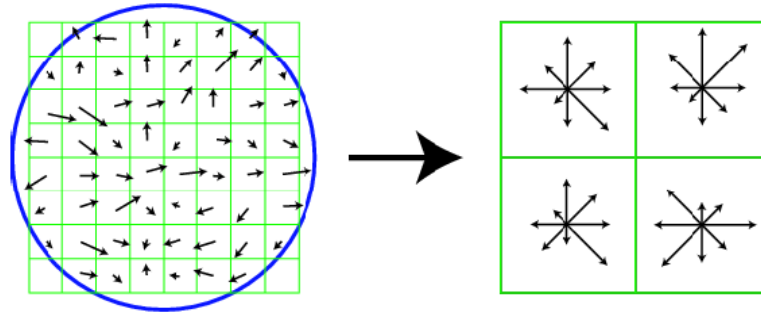


Abbildung 2.21.: Aus Orientierungshistogrammen entstandene Orientierungsfenster [7]

## 2.7. (SURF) Speeded-Up-Robust-Features-Algorithmus

Das Rechenverfahren des SURF-Algorithmus ist an den vorherigen SIFT-Algorithmus angelehnt. Ein wesentliches Merkmal zur Erkennung von robusten Interessensmerkmalen beim SURF-Algorithmus ist die Verwendung von Integralbildern aus Abschnitt 2.3.2.. Für eine effiziente Wiederfindung der Interessensmerkmale in nachfolgenden Bildern werden diese durch Nutzen von SURF-Deskriptoren näher beschrieben. Ähnlich wie beim SIFT-Algorithmus ist der SURF-Algorithmus invariant gegenüber Rotationen, Skalierungen, Kontraständerungen, Bildrauschen und weitgehend gegenüber perspektivischen Verzerrungen.

Die generierten Integralbilder werden mit dem Fast-Hessian-Detektor auf Interessensmerkmale untersucht. Der Fast-Hessian-Detektor in Gleichung 2.28 berechnet die Determinante der Hessematrix, die bei potenziellen Interessensmerkmalen lokale Maxima erreicht. Die Elemente der Hessematrix bestehen aus dem gefalteten Bildausschnitt, die aus der Faltung der zweiten partiellen Ableitungen der „Laplacian of Gaussians“ gebildet werden [14].

$$\underline{\underline{H}} = \begin{bmatrix} G_{xx}(x, y, \sigma) * I(x, y) & G_{xy}(x, y, \sigma) * I(x, y) \\ G_{xy}(x, y, \sigma) * I(x, y) & G_{yy}(x, y, \sigma) * I(x, y) \end{bmatrix} \quad (2.28)$$

Der „Laplacian of Gaussian“ verwendet Laplace-Operatoren, die auf die Gaußfunktion angewendet werden und als Ergebnis die zweite partielle Ableitung liefern. Mit der Varianz der Gaußfunktion lässt sich das Bild glätten, mit der die Determinante auf verschiedenen Skalierungsebenen ermittelt wird. Für die Faltung des Bildausschnittes werden Filtermasken zur Diskretisierung eingesetzt [14].

Für die Faltung des Bildausschnittes beim SURF-Algorithmus werden Mittelwertfilter, sogenannte Boxfilter verwendet, die eine erweiterte Approximation der diskretisierten Filtermasken darstellen. Die beiden Faltungsoperationen sind in Abbildung 2.22 veranschaulicht [14].

Die Determinante der Hessematrix ist in Gleichung 2.29 definiert. Hierbei definiert die Variable  $w$  die Gewichtung des gefilterten Bildausschnittes. Zur Veranschaulichung der Filterungsoperation dient die Abbildung 2.22 in der durch die Gewichtung die entstandene Filterfläche entsteht. Der Mittelwertfilter liefert bei einer Gewichtung von  $w = 0,9$  optimale Grauwerte für jeden Bildpunkt der gefilterten Integralbilder [14].

$$\det(\underline{\underline{H}}_{approx}) = D_{xx}(x, y, \sigma)D_{yy}(x, y, \sigma) - (wD_{xy}(x, y, \sigma))^2 \quad (2.29)$$

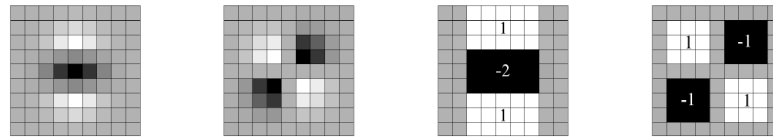


Abbildung 2.22.: Erweiterte Diskretisierung der LoG-Filter durch Mittelwertfilter [14]

Zur Erkennung von Interessensmerkmalen werden analog zum Detektor des SIFT-Algorithmus lokale Extremwerte durch Vergleiche der 26 Werte aus den benachbarten Oktaven, auf jeder Oktave ermittelt. Um subpixelgenaue Positionen der Interessensmerkmale im Originalbild zu erhalten, werden die exakten Positionen der Extremwerte aus den Skalierungsebenen mit dem Interpolationsverfahren bestimmt [14].

Hierzu wird mit Hilfe der Taylor-Entwicklung die Hessematrix, als eine Funktion über den Skalenraum erstellt. Durch Nullsetzen und Differenzieren der Funktion nach dem Abstand  $\underline{x}$  ergibt sich eine lineare Funktion aus Gleichung 2.30 aus der die subpixelgenaue Position interpoliert wird [14].

$$\underline{x} = \left( \frac{\delta^2 \underline{\underline{H}}(\underline{x}_e)}{\delta \underline{x}^2} \right) \frac{\delta \underline{\underline{H}}(\underline{x}_e)}{\delta \underline{x}} \quad \text{mit: } \underline{x}_e = (x, y, \sigma)^T \quad (2.30)$$

Ausgehend vom detektierten Interessensmerkmal werden in einem Radius der 6-fachen Größe der Bildskalierung die Haar-Wavlets aus Abschnitt 2.3.2. lokalisiert. Diese werden anschließend durch die im Zentrum des Interessensmerkmals liegende Gaußfunktion gewichtet. Für die Orientierung der Interessensmerkmale wird die gesamte Kreisfläche mit einem schrittweise rotierenden Kreisausschnitt der Größe  $\frac{\pi}{3}$  um das Interessensmerkmal untersucht. Die im Kreisausschnitt befindlichen Grauwertintensitäten der Haar-Wavlets werden aufsummiert und bestimmen die Gradientenrichtung. Zur Orientierung wird die dominierende Gradientenrichtung aus den Kreisausschnitten ausgewählt [Abbildung 2.23] [14].

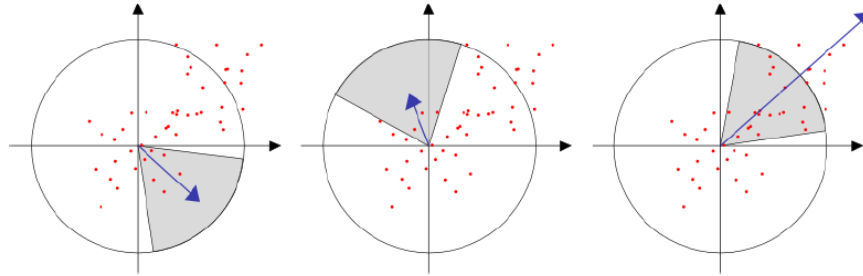


Abbildung 2.23.: Orientierung der benachbarten Interessensmerkmale [20]

Am Interessensmerkmal wird anschließend parallel zu der zuvor bestimmten Orientierung ein rechteckiges Fenster der 20-fachen Bildskalierung aufgespannt. Dieses Fenster wird in 16 gleich große Subfenster aufgeteilt, die mit Haar-Wavlet-Filtern Gradientenrichtungen der 25 Bildpunkte in dem jeweiligen Subfenster berechnet. Für das Subfenster wird ein 4 dimensionaler Vektor aufgespannt, der aus den Grauwerten der Harris-Wavlets besteht. Die Vorgehensweise ist in Abbildung 2.24 vereinfacht dargestellt [14].

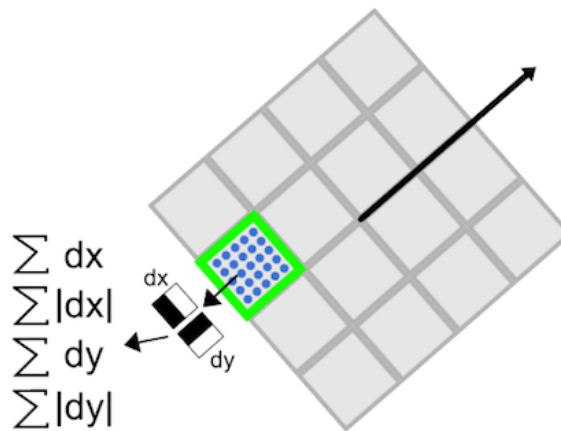


Abbildung 2.24.: Vorgehensweise zur Charakterisierung der Interessensmerkmale [21]

Durch Zusammenfassen des Subfensters wird ein Deskriptor-Vektor mit 64 Elementen erzielt. Die Abbildung 2.25 stellt die Wavlet-Antworten in einem Subfenster dar. Diese geben die Pixelintensität der jeweiligen Ausrichtungen an [14].

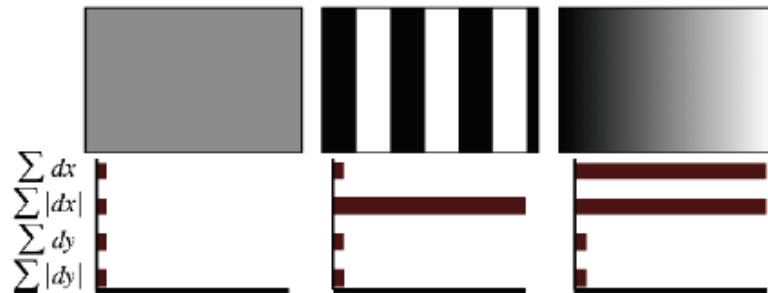


Abbildung 2.25.: Wavlet-Antworten im Subfenster [22]

Für die Wiederfindung der Interessensmerkmale werden die Elemente der Deskriptor-Vektoren durch Distanzberechnungen aus Abschnitt 2.4.1. miteinander verglichen. Geringe Distanzen deuten auf eine hohe Übereinstimmung der zu vergleichenden Interessensmerkmale und führen zu Korrespondenzen [14].

## 2.8. (BRISK)

### Binary-Robust-Invariant-Scalable-Keypoints-Algorithmus

Für eine genaue Beschreibung der erkannten Interessensmerkmale verwendet der BRISK-Algorithmus wie beim SIFT- und SURF-Algorithmus einen Deskriptor, der anschließend mit weiteren Deskriptoren aus nachfolgenden Eingangsbildern verglichen wird. Besonders auffällig beim BRISK-Algorithmus ist die hohe Zuverlässigkeit der Erkennung und Wiederfindung bei geringer Rechenzeit [23].

Der BRISK-Detektor bildet eine Kreislinie mit einer definierten Länge von 16 Pixeln. Um eine Ecke zu identifizieren wird der Grauwert im Mittelpunkt des Kreises mit den Grauwerten auf der Kreislinie verglichen. Für ein beschleunigtes Verfahren werden zunächst die Grauwerte der Bildpunkte 1 und 9 verglichen. Bei Intensitätsunterschieden folgt eine weitere Überprüfung der Grauwerte mit 5 und 13, wie aus Abbildung 2.26 hervorgeht. Wenn aus den Vergleichen Intensitätsunterschiede erkannt werden, ist ein möglicher Eckpunkt vorhanden. Um eine eindeutige Ecke zu identifizieren, werden die übrigen Grauwerte auf der Kreislinie mit dem Grauwert im Mittelpunkt verglichen. Ein identifizierter Eckpunkt, der in seiner Umgebung am stärksten geprägt ist, definiert hierbei das Maximum [23].

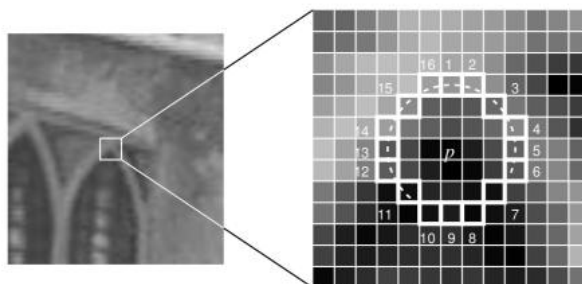


Abbildung 2.26.: Vergleich der benachbarten Grauwerte im Suchkreis [23]

Wie bereits aus Abschnitt 2.3.3. hervorgeht, werden für die skalierungsinvarianten Interessensmerkmale skalierte Bildebenen verwendet, die aus Oktaven und Suboktaven bestehen [16].

In jeder Oktave und Suboktave werden die Maxima der Eckpunkte ermittelt. Das gefundene Maximum eines Eckpunktes auf einer Oktave wird mit dem Maximum der vorherigen und nachfolgenden Suboktave verglichen. Für ein Maximum eines Eckpunktes auf der Suboktave erfolgt ein Vergleich des Maximums mit den benachbarten Oktaven. Dies ermöglicht eine zuverlässige Identifikation eines Eckpunktes, um mehrfach detektierte Eckpunkte zu beseitigen [16].

Für eine subpixelgenaue Positionsbestimmung eines detektierten Eckpunktes wird in jeder Skalierungsebene des Multiskalenraums eine zweidimensionale, quadratische Funktion im Mittelpunkt des identifizierten Eckpunktes aufgestellt. Die Maxima der Funktionen ergeben die subpixelgenaue Position auf den Skalierungsebenen. Für eine genaue Bestimmung der Position im Skalenraum, wird an den neu ermittelten Positionen der Skalierungsebenen eine eindimensionale Parabel angelegt. Das Maximum der eindimensionalen Parabel bestimmt die Skalenposition.

Für eine exakte Positionsbestimmung der Eckpunkte im Eingabebild wird mit dem Interpolationsverfahren die Skalenposition zwischen den Skalierungsebenen ermittelt. Die Abbildung 2.27 zeigt eine allgemeine Darstellung des BRISK-Detektors [16].

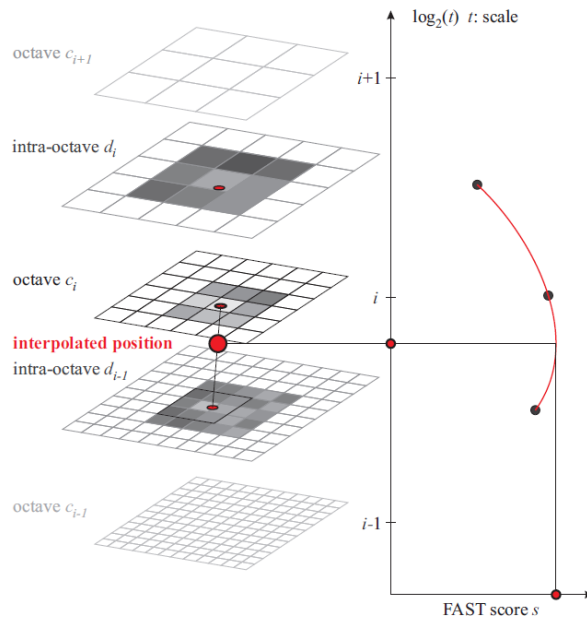


Abbildung 2.27.: Bestimmen der subpixelgenauen Position im Multiskalenraum [16]

Um den Eckpunkt wird eine kreisförmige Maske aufgestellt, die aus mehreren gleich großen Hilfspunkten besteht. Mit einer Gaußfunktion werden für die Hilfspunkte Glättungswerte bestimmt und mit einem Glättungsradius versehen [Abbildung 2.28]. Mit zunehmendem Abstand zum Interessensmerkmal steigt der Glättungsradius der Hilspunkte [16].

Für die Orientierung der Interessensmerkmale werden aus mehreren Hilfspunktpaaren die Gradienten aus den Glättungswerten bestimmt, die anschließend gemittelt werden [Gleichung 2.31] [16].

$$g(i, j) = \frac{1}{\|D(i, j)\|_2} \sum_{(p_i, p_j) \in D} \left[ \frac{p_j - p_i}{\|p_j - p_i\|_2} \cdot \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|_2} \right] \quad \text{mit: } D \Rightarrow \|p_j - p_i\|_2 > \delta_{min} \quad (2.31)$$

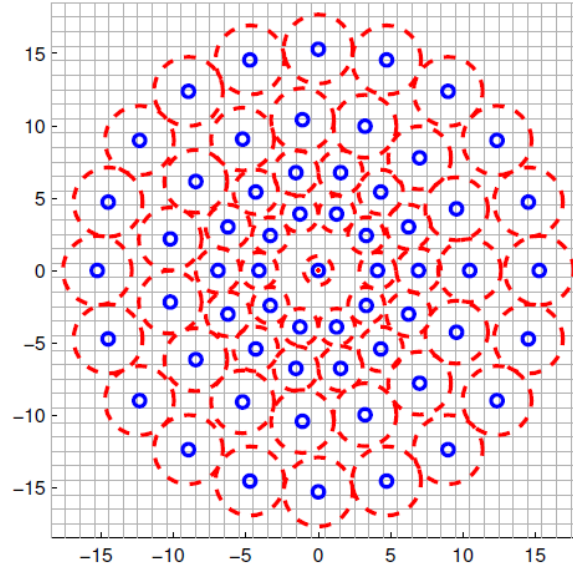


Abbildung 2.28.: Kreismaske um einen detektierten Eckpunkt [16]

Um eine Rotationsinvarianz zu erzeugen, wird erneut eine kreisförmige Maske am Interessensmerkmal parallel zur Orientierung angelegt, die durch die ermittelte Skalenposition entsprechend skaliert wird. Die neuen Hilfspunkte werden miteinander in einer festen Reihenfolge verglichen, um gleiche Deskriptoren in nachfolgenden Bildern wiederzufinden. Die Anzahl der Vergleiche bestimmt die Bitanzahl, die zu einem Bitstring zusammengeführt werden. Die einzelnen Bits bestehen, wie aus Gleichung 2.32 hervorgeht aus 2 Werten [16].

$$\underline{d}_{bitstring} = [\underline{d}_1, \dots, \underline{d}_n] \quad \text{mit:} \quad \underline{d}_n \begin{cases} 1 & \text{für } I(p_j, \sigma_j) > I(p_i, fr\sigma_i) \\ 0 & \text{für } I(p_j, \sigma_j) \leq I(p_i, fr\sigma_i) \end{cases} \quad (2.32)$$

Um mögliche Korrespondenzen in einer Bildfolge zu erkennen, werden die Deskriptoren mit dem Ähnlichkeitsmaß in Gleichung 2.10 aus Abschnitt 2.4.1. verglichen. Eine große Ähnlichkeit zweier Deskriptoren ist durch eine geringe Distanz der Bitstringelemente gekennzeichnet [16].

# 3 Analyse der Feature-Tracker-Algorithmen

## 3.1. Auswertung der Trackingverfahren

In diesem Kapitel werden die Trackingverfahren zunächst mit Hilfe von Experimenten aus anderen Versuchen gegenübergestellt. Diesbezüglich werden die Vor- und Nachteile der Trackingverfahren unter verschiedenen Bedingungen der Bildaufnahme untersucht. Die Ergebnisse der Untersuchungen beziehen sich hierbei auf die mathematischen Grundlagen aus dem vorherigen Kapitel. Aus dem Vergleich der Trackingverfahren werden anschließend die resultierenden Fehler analysiert und in den Feature Tracker Simulator übertragen.

### 3.1.1. Vergleich der Trackingverfahren

Im Allgemeinen erfüllt ein zuverlässiger Feature Tracker die Voraussetzungen einer exakten Positionsbestimmung im Bildausschnitt durch Robustheit gegenüber dem Bildrauschen, Invarianz gegenüber den photometrischen und geometrischen Veränderungen, Bildverzerrungen durch Bewegungen, geringe Rechenzeiten und durch eine robuste Wiederfindung der Interessensmerkmale [24].

Für einen qualitativen Vergleich der Trackingverfahren werden nachstehend die Detektoren und Deskriptoren getrennt voneinander betrachtet.

### 3.1.2. Vergleich der Detektoren

Detektoren bestehen aus verschiedenen Verfahren der Erkennung von Interessensmerkmalen. Die Eckendetektoren, wie der Harris-, Shi-Tomasi- und Fast-Eckendetektor aus Kapitel 2 erfordern zwar geringe Rechenzeiten um Interessensmerkmale zu erkennen, sind jedoch nicht skalierungsinvariant und liefern infolgedessen ungenaue Positionen bei skalierungs- und sichtverändernden Bildsequenzen. Im Gegensatz zu den Eckendetektoren sind Blobdetektoren skalierungsinvariant und ermitteln auch bei skalierungs- und sichtverändernden Bildfolgen subpixelgenaue Positionen der Interessensmerkmale, die jedoch durch die Untersuchungen im Skalenraum hohe Rechenzeiten erfordern. Blobdetektoren sind zwar für eine robuste Erkennung von Interessensmerkmalen in Bildsequenzen bekannt, haben jedoch den Nachteil, dass eine große Eckenanzahl in einem engen Bildbereich vernachlässigt wird [24].



	Corner Detector	Blob Detector	Rotation Invariant	Scale Invariant	Affine Invariant	Repeatability	Localization Accuracy	Robustness	Efficiency
Harris	x		x			+++	+++	++	++
Shi-Tomasi	x		x			+++	+++	++	++
FAST	x		x	x		++	++	++	++++
SIFT		x	x	x	x	+++	++	+++	+

Abbildung 3.1.: Vergleich der Detektoren [24]

Die Abbildung 3.1 stellt die Detektoren unter den oben genannten Bedingungen auf. Eine hohe Effizienz der Detektoren ist hierbei mit einer großen Anzahl der (+)-Marker versehen. Des Weiteren ist aus der Abbildung 3.1 zu erkennen, dass die Detektoren des SIFT- und SURF-Algorithmus bei einer kontinuierlichen Sichtveränderung bis zu einem gewissen Grad invariant gegenüber affinen Abbildungen sind [24].

Durch die Verwendung von Wavlet-Filtern der Fast-Hessematrix wird aus der Abbildung 3.1 ersichtlich, dass die Rechenzeit des SURF-Algorithmus geringer ist als beim SIFT-Algorithmus.

### 3.1.3. Vergleich der deskriptiven Trackingverfahren

Aus den Grundlagen der Trackingverfahren geht hervor, dass die SIFT-, SURF- und BRISK- Algorithmen Interessensmerkmale für die Wiedererkennung charakterisieren.

Für einen Vergleich der deskriptiven Trackingverfahren werden die Korrespondenzen der Deskriptoren, die aus einer Anzahl richtiger und falscher Korrespondenzen eines Bildpaares bestehen, untersucht. Zur Ermittlung der Sensitivität der Korrespondenzen, wurden die Deskriptoren mit einer Recall- und Precision-Funktion der binären Klassifikation beurteilt [25].

$$Recall = \frac{Correct \ matches}{Correspondences} \quad (3.1)$$

$$1 - Precision = \frac{Falsematches}{correct \ matches + false \ matches} \quad (3.2)$$

Die Recall-Funktion in Gleichung 3.1 beschreibt hierbei die Performanz der Korrespondenzsuche. Zur Evaluierung der Treffergenauigkeit von Korrespondenzen wird die 1-Precision-Funktion in Gleichung 3.2 verwendet. Bei einer zuverlässigen Korrespondenzsuche liefern die Funktionen hohe Werte [25].

Für die Evaluation der Deskriptoren wurden verschiedene Versuche analysiert. In dem nachfolgenden Versuch wurden die Deskriptoren auf Rotationen, Skalierungen, Verzerrungen, Bildrauschen, Sichtänderungen und Auflösungsreduktionen der Bilder untersucht. Der Datensatz aus dem Versuch bestand aus einer Graubildfolge aus einer natürlichen Umgebung. Aus [25] können die verwendeten Kalibrierwerte des bildgebenden Sensors und die Parameter der Trackingalgorithmen nachgeschlagen werden.

Die Tabellen 3.1 stellen die Deskriptoren aus dem Versuch unter den Bedingungen mit den Sensitivitätsfunktionen gegenüber [25].

Sichtänderung			Rotation		
Descriptor	Precision	Recall	Descriptor	Precision	Recall
SIFT	14.6	39.6	SIFT	44.2	76.4
SURF	31.7	44.0	SURF	62.7	75.3
BRISK	20.7	42.2	BRISK	37.7	63.5
Skalierung			Verzerrung		
Descriptor	Precision	Recall	Descriptor	Precision	Recall
SIFT	57.5	83.4	SIFT	35.7	60.9
SURF	60.7	81.2	SURF	63.4	66.7
BRISK	35.1	59.6	BRISK	21.5	53.4
Bildrauschen			Auflösungsreduktion		
Descriptor	Precision	Recall	Descriptor	Precision	Recall
SIFT	45.7	72.9	SIFT	26.9	36.7
SURF	73.5	80.7	SURF	78.2	85.8
BRISK	47.5	62.3	BRISK	15.9	42.7

Tabelle 3.1.: Vergleich der evaluierten Deskriptoren aus [25]

Die in Tabelle 3.1 resultierenden Werten aus den Gleichungen 3.1 und 3.2 beschreiben für diesen Datensatz aus dem Versuch [25] die Genauigkeit der deskriptiven Trackingverfahren.

Aus den Tabellenwerten wird deutlich, dass der SURF- und SIFT-Deskriptor ähnliche Werte bei einigen Bedingungen erreichen. Interessant ist hierbei, dass bei einer Auflösungsreduktion der SURF-Deskriptor gegenüber den anderen Deskriptoren deutlich besser abschneidet. Der Grund für die bessere Ergebniserzielung ist auf die Berechnung der Integralbilder zurückzuführen, die durch Vergrößern des Bildausschnittes ebenfalls eine Auflösungsreduktion bewirken und somit einen robusteren Vergleich der Interessensmerkmale ermöglichen.

Unter Rotationen, Skalierungen und Verzerrungen liefert der BRISK-Deskriptor schlechte Ergebnisse. Insgesamt fallen die Ergebnisse bei allen Deskriptoren bei Sichtänderungen und Bildverzerrungen schlechter aus.

In einem weiteren Versuch, wurden speziell die SURF- und SIFT-Deskriptoren unter ähnlichen Bedingungen für die simultane Lokalisierung und Kartenerstellung (SLAM) untersucht. Aus dem Versuch geht hervor, dass bei sehr großen Rotationen ohne Deskriptor-Orientierung starke Präzisionsverluste unter 10% zustande kommen [26].

Auffällig ist hierbei, dass mit der Deskriptor-Orientierung nur das SIFT-Trackingverfahren eine Verbesserung der Ergebnisse erzielte. Des Weiteren konnte aus dieser Untersuchung festgestellt werden, dass bei schnellen Bildbewegungen das SIFT-Trackingverfahren bessere Ergebnisse erzielte.

### 3.1.4. Rechenzeit der Trackingverfahren

Für den Vergleich der Rechenzeit für die in dieser Arbeit relevanten Trackingverfahren, wurden weitere Versuche analysiert. In [27] wurde das deskriptorlose KLT-Trackingverfahren und die deskriptorbasierten SIFT-, SURF- und BRISK- Trackingverfahren unter Echtzeitaufnahmen der Fluoroskopie untersucht. Die Fluoroskopie, die mit Röntgenstrahlen Bildaufnahmen der Anatomie erzeugt, nutzt zur Bewegungsanalyse der Organe Trackingverfahren. In dieser Untersuchung lag der Schwerpunkt auf der Echtzeitfähigkeit, Genauigkeit und Zuverlässigkeit der Trackingverfahren. Die Gesamtrechenzeit der Trackingverfahren zwischen zwei Echtzeitaufnahmen ist in Tabelle 3.2 aufgeführt. Hierbei stellt die relative Zeit mit Gleichung 3.3 eine Beziehung zwischen der Gesamtrechenzeit und den korrespondierenden Interessensmerkmalen her [27].

$$t_{relative} = \frac{T_{absolut}}{\frac{\text{detected features (in current image)}}{\text{matching features (in further images)}}} \quad (3.3)$$

relative Zeit	deskriptorlose Trackingverfahren	deskriptive Trackingverfahren		
-	KLT	SIFT	SURF	BRISK
$t_{relative}[ms]$	0.068	61.636	7.866	4.381

Tabelle 3.2.: Vergleich der evaluierten Deskriptoren nach der Rechenzeit aus [27]

Aus der Tabelle 3.2 ist ersichtlich, dass das SIFT-Trackingverfahren eine sehr hohe Rechenzeit erfordert. Dies ist auf die aufwendige Berechnung im Multiskalenraum und auf den Vergleich der Elemente des mehrdimensionalen Deskriptor-Vektors zurückzuführen. Das BRISK-Trackingverfahren fällt unter den deskriptiven Trackingverfahren durch seinen Vergleich der einfachen binären Deskriptor-Vektoren am besten aus. Durch Anwendung der Integralbilder und der primitiven Bestimmung der Grauwertintensitäten durch Wavlet-Filter sind Interessensmerkmale mit dem SURF-Trackingverfahren schnell und zuverlässig zu bestimmen. Das KLT-Trackingverfahren liefert in dieser Untersuchung die schnellste Rechenzeit. Die Ursache hierfür liegt bei der schnellen Erkennung mit dem einfachen Shi-Tomasi-Eckendetektor und die aus dem optischen Fluss resultierenden Interessensmerkmale.

In einem weiteren Versuch [16], in der die 3 deskriptiven Trackingverfahren unter anderem auf die Rechenzeit hin untersucht wurden, erzielten der SIFT-Algorithmus sehr hohe und der SURF- und BRISK-Algorithmus geringe Rechenzeiten. Für den Versuch wurden hierbei Farbbildbildfolgen einer natürlichen Umgebung verwendet.

## 3.2. Fehlermerkmale der Trackingverfahren

Bereits aus dem Vergleich der Trackingverfahren geht hervor, dass durch eine Veränderung der Umgebung die Wiederfindung der Interessensmerkmale in nachfolgenden Bildausschnitten abnimmt und somit zu Positionsfehlern der Interessensmerkmale führen kann. Grundsätzlich lassen sich Fehlermerkmale auf diskontinuierliche Bildfolgen zurückverfolgen. Typische Fehlermerkmale einer diskontinuierlichen Bildfolge sind starkes Bildrauschen, verzerrte Bildaufnahmen, Sichtänderungen, Überdeckungen der Interessensmerkmale, Helligkeitsänderungen, translatorische und rotatorische Änderungen, sowie Skalierungsänderungen. Die Ursachen der Fehlermerkmale lassen sich nicht immer nur auf die Berechnungen der Algorithmen zurückführen. Schlechte Kalibrierungen der bildgebenden Sensoren und infolgedessen erzeugtes Bildrauschen, geringe Bildauflösungen, schlechte Beleuchtungsbedingungen oder Linsenverzerrungen können die Fehleranfälligkeit enorm verstärken. Im Folgenden werden die Ursachen der Fehlermerkmale aus den Trackingverfahren näher untersucht [24].

### 3.2.1. Deskriptorlose Trackingverfahren

#### KLT Trackingverfahren

Wie bereits aus den Grundlagen bekannt, nutzt das KLT-Trackingverfahren für die Wiederfindung der Interessensmerkmale keine Deskriptoren. Für die Verfolgung der Interessensmerkmale mit dem optischen Fluss in einer Bildsequenz liefert das Trackingverfahren konstante iterativ ermittelte Transformationsparameter in Bildpaaren, die im verschobenen Bildausschnitt die neue Position des Interessensmerkmals bestimmt. Durch den begrenzten Bildausschnitt können nur lokale Interessensmerkmale verfolgt werden, die nach Austreten des Bildausschnittes verloren gehen. Die verlorenen Interessensmerkmale werden anschließend durch neue detektierte Interessensmerkmale ersetzt. Daraus folgend bestehen für die Interessensmerkmale feste ID-Zuweisungen nur im lokalen Bildausschnitt, die nach Verschwinden des Interessensmerkmals erlöscht [28][Abbildung 3.2].

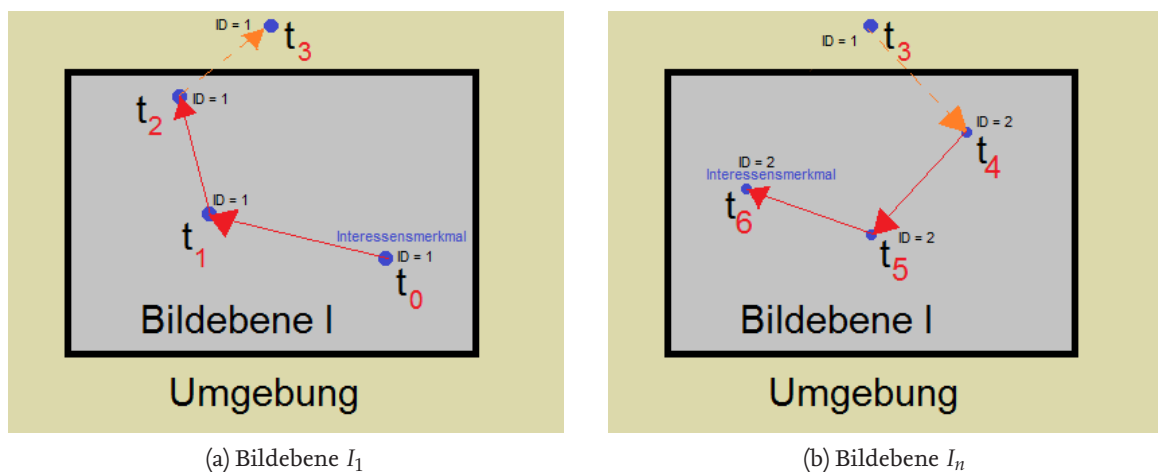


Abbildung 3.2.: Neue ID-Zuweisung des Interessensmerkmals bei erneutem Eintreten in die Bildebene

Deskriptive Trackingverfahren, die im Multiskalenraum Interessensmerkmale untersuchen, finden diese durch den Vergleich der Deskriptor-Vektoren wieder.

Das KLT-Trackingverfahren ist außerdem sehr empfindlich gegenüber starken diskontinuierlichen Bildbewegungen [29]. Für kontinuierliche Bildbewegungen sind genaue Ergebnisse der Transformationsparameter und daraus folgend die Position der Interessensmerkmale realisierbar. Für eine optimale Berechnung des KLT-Trackingverfahrens wird eine kontinuierliche Bildbewegung angenommen [29]. Eine diskontinuierliche Bildbewegung führt zu Berechnungsfehlern, die durch Akkumulieren der Berechnungsfehler zu Driftfehlern der Interessensmerkmale führen können [29]. Für die Gleichungen des KLT-Trackingverfahrens wird des Weiteren angenommen, dass die Bildgrauwerte der Bildpunkte in Bildsequenzen konstant bleiben, was unter realen Gegebenheiten nur bedingt erreicht werden kann [29]. Verändernde Bildgrauwerte, wie beispielsweise Schattenentwürfe oder Helligkeitsänderungen, führen zu abweichenden Eigenwerten des Shi-Tomasi-Eckendetektors, der in Gleichung 2.18 zu fehlerhaften Differenzen und infolgedessen zur inkorrekten Positionsbestimmung der Interessensmerkmale führt. Dies kann zu verrauschten Bildpunkten führen, die durch weitere Helligkeitsänderungen Driftfehler hervorrufen [29].

Eine weitere Annahme beim KLT-Trackingverfahren ist, dass benachbarte Bildpunkte einer Bildszena gleiche oder ähnliche Geschwindigkeiten besitzen [29]. Insbesondere in kleinen Bildausschnitten, in denen für die Berechnung des optischen Flusses Bildpunkte auf Kanten liegen können, sind die Bewegungen nur senkrecht zur Grauwertkante oder parallel zum Grauwertgradienten zu ermitteln. Dieses Problem wird auch als Aperture-Problem bezeichnet. Aufgrund des Aperturproblems können Bewegungen nur senkrecht zur Grauwertkante oder parallel zum Grauwertgradienten erkannt werden. [29] In der Abbildung 3.3 (a) und (c) wird deutlich, dass hierbei die Bewegungsrichtung des Bildpunktes nicht präzise erfasst werden können.

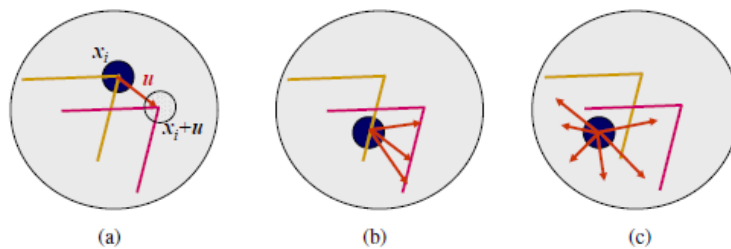


Abbildung 3.3.: Aperture-Probleme bei schlecht detektierten Interessensmerkmalen im Bild [19]

Eckendetektoren, die nicht skalierungs- und rotationsinvariant sind, führen bei Rotationen oder Skalierungen des Bildausschnittes in Gleichung 2.17 zu falschen Differenzen und somit zu stark abweichenden Transformationsparametern, die verrauschte Positionen oder Driftfehler erzeugen [18].

Die zustande kommenden Driftfehler durch Akkumulationsfehler können durch Überkreuzungen zu einer temporären Überlagerung der Interessensmerkmale führen. Die Überlagerungen können hierbei Verwechslungen der ID-Zuweisung der Interessensmerkmale bewirken [18]. Perspektivische Verdeckungen oder verrauschte Bildaufnahmen durch die schlechte Kalibrierung der bildgebenden Sensoren erzeugen ebenfalls Berechnungsfehler und können Positionsrauschen und Driftfehler verstärken [18].

Bei der Verwendung der Eckendetektoren können Trackingfehler wie beispielsweise das Kantendriften entstehen.

Durch die Verwendung des Shi-Tomasi-Eckendetektors werden robuste Interessensmerkmale durch den Vergleich der Eigenwerte mit einem vordefinierten Konvergenzlimit der Hesse-Matrix untersucht. Bei einem zu klein gewählten Konvergenzlimit können Kanten dennoch erkannt werden, wenn beispielsweise für den kleinsten Eigenwert  $\lambda_{min} \approx 0$  und für den größten Eigenwert  $\lambda_{max} \gg 0$  deutlich größere Werte angenommen werden. Dies kann zur falschen Differenzierung der Gleichung 2.9 führen, da bei der Verfolgung die Interessensmerkmale auf der Kante ähnliche Grauwerte besitzen und sich somit nicht eindeutig detektieren lassen. Dies kann zu einem Kantendriftfehler führen [29].

### 3.2.2. Deskriptive Trackingverfahren

#### SIFT-, SURF und BRISK-Trackingverfahren

Aus den Vergleichen der Trackingverfahren geht hervor, dass die deskriptiven Trackingverfahren bei Rotationen, Translationen und Skalierungen gegenüber dem KLT-Trackingverfahren bessere Ergebnisse erzielen.

Die deskriptiven Trackingverfahren nutzen die Charakteristik der Interessensmerkmale, um einen robusteren Vergleich zu ermöglichen. Anders als beim KLT-Trackingverfahren werden zur Wiederfindung die gespeicherten charakterisierenden Interessensmerkmale in Deskriptor-Vektoren durch Ähnlichkeitsmaße miteinander verglichen. Dies ermöglicht das Wiederfinden von verlorenen Interessensmerkmalen aus vorherigen Bildausschnitten [Abbildung 3.4].

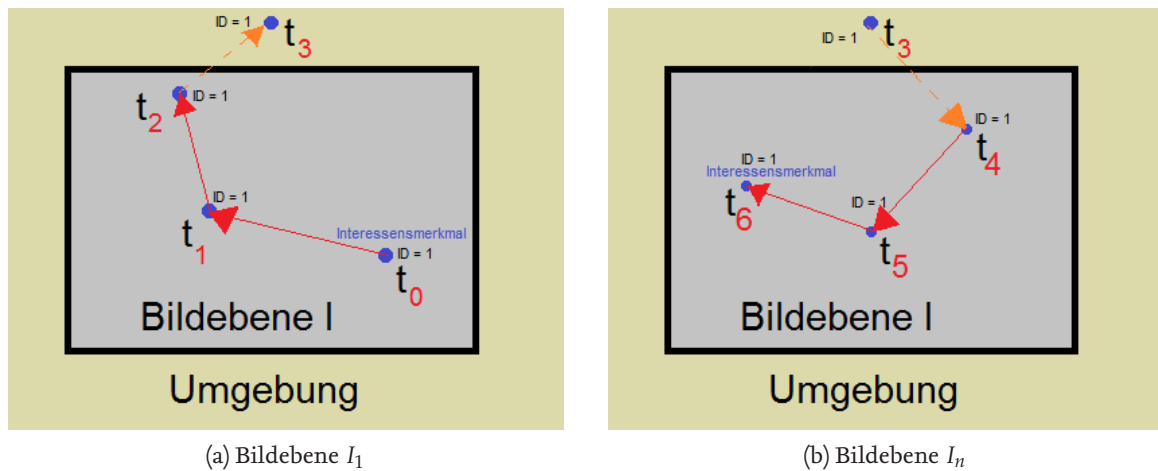


Abbildung 3.4.: Vorherige ID-Zuweisung des Interessensmerkmals bei erneutem Eintreten in die Bildebene

Wie aus den Grundlagen bereits bekannt, werden die Bilder nach Extrema in den einzelnen Skalenebenen untersucht. Interessensmerkmale entstehen durch den Vergleich der Bildgrauwerte der benachbarten Bildpunkte. Hierzu werden, wie in Abbildung 3.5 dargestellt, die diskreten Bildgrauwerte mit einer kontinuierlichen Difference-of-Gaussian-Funktion angenähert, die eine Approximation der Laplace-Funktion darstellt.

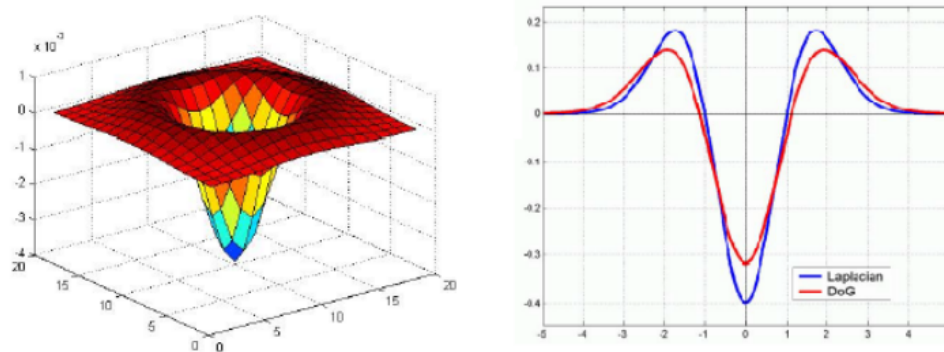


Abbildung 3.5.: Stetige Bildfunktion der jeweiligen Skalenebene als approximierte Laplace-Funktion [30]

Mit der Ableitung der Difference-of-Gaussian-Funktion in Gleichung 2.21 aus Abschnitt 2.6 und durch Nullsetzen sind die Extrema für jede Skalenebene zu ermitteln. Durch das Interpolationsverfahren werden die subpixelgenauen Positionen der Interessensmerkmale bestimmt, die durch die Diskretisierung in Bildpunkte Abweichungen hervorruft. Hierbei können Abweichungen vom Interessensmerkmal um einen Bildpunkt in jede Richtung entstehen [31].

Diese entstehen, wenn die subpixelgenaue Position genau zwischen zwei Bildpunkten liegt und zu einem der Bildpunkte zugewiesen wird. [Abbildung 3.6] Die entstandenen Abweichungen und die daraus folgende wechselnde Bildpunktzuweisung in einer Bildsequenz, ist die Ursache für das Verrauschen der Positionen der Interessensmerkmale [31].

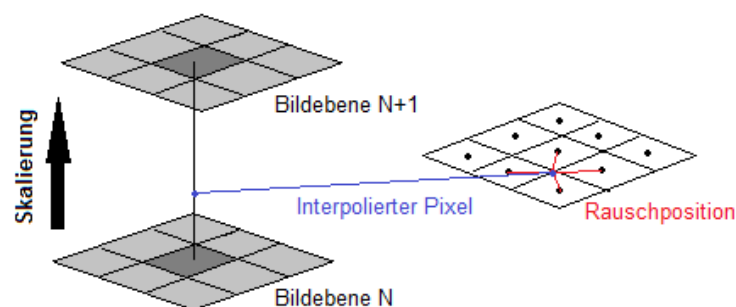


Abbildung 3.6.: Abweichung der subpixelgenauen Position durch die Diskretisierung



Beim Vergleich der Charakteristika von Deskriptor-Vektoren in Bildpaaren, werden die Elemente durch Ähnlichkeitsmaße in Beziehung gesetzt. Eine Korrespondenz von Interessensmerkmalen tritt ein, wenn hierbei minimale Distanzen erzielt werden. Fehlerhafte Korrespondenzen tauchen auf, wenn zwar minimale Distanzen der Deskriptor-Vektoren erzielt werden, diese jedoch aus nicht identischen Bildpunktpaaren bestehen. Die fehlerhaften Korrespondenzen bilden hierbei die Mismatches in Bildsequenzen [32].

Das Sift-Trackingverfahren unterdrückt Interessensmerkmale, die auf kantenartigen Bildstrukturen liegen, durch den Vergleich der Eigenwerte der Hessematrix in Gleichung 2.8 aus Abschnitt 2.2.2. Durch die Unterdrückung sind robuste Interessensmerkmale erkennbar, die zu keinem Kantendriftfehler führen [7].

Aus den Grundlagen zum Surf-Trackingverfahren geht hervor, dass zur Detektion der Interessensmerkmale die Fast-Hessematrix verwendet wird. Durch die Berechnung der Determinanten aus der Fast-Hessematrix wird in dem Bildausschnitt nach lokalen Maxima gesucht. Mit dem stetigen Laplacian-of-Gaussian-Filter, die Extrema in den Skalenebenen ermittelt, werden die Skalenebenen durch Anwendung der Wavlet-Filter diskretisiert. Die Diskretisierung bewirkt durch die Gewichtung eine Normierung der gefundenen Blobs, die zu verfälschten Positionen der Interessensmerkmale führt. Des Weiteren ist die Größe der Filterfläche abhängig von den ermittelten Grauwerten der Integralbilder, die zu einer verrauschten Position der Interessenspunkte führen.

Analog zum SIFT-Trackingverfahren werden Interessensmerkmale auf kantenartigen Bildstrukturen gleichfalls durch den Vergleich der Eigenwerte aus der Hessematrix unterdrückt, die Kantendriftfehler der Interessensmerkmale verhindert [7].

Die Elemente der Deskriptor-Vektoren des SURF-Trackingverfahrens, werden auch hier durch Distanzbestimmungen der Ähnlichkeitsmaße miteinander verglichen. Hierbei können ebenfalls fehlerhafte Korrespondenzen durch minimale Distanzen der Elemente berechnet werden, die in Bildsequenzen als Mismatches verstanden werden [32].

Beim BRISK-Trackingverfahren werden für die Detektion der Interessensmerkmale Kreismasken verwendet, in der durch den Vergleich der Grauwerte auf der Kreislinie in mehreren Skalenebenen potenzielle Eckpunkte ermittelt werden. Die subpixelgenaue Position der Interessensmerkmale, die durch das Interpolationsverfahren zwischen Eckpunkten aus Abbildung 2.27 bestimmt wird, führt durch die Diskretisierung auch hier zu Abweichungen und zur wechselnden Bildpunktzuweisung in Bildsequenzen. Dies kann eine verrauschte Position der Interessensmerkmale erzeugen.

Auch beim BRISK-Trackingverfahren können Mismatches durch fehlerhafte Korrespondenzen entstehen. Durch den deterministischen Vergleich der binären Deskriptor-Vektoren, kann eine größere Anzahl der fehlerhaften Korrespondenzen entstehen, da hierbei die Bitstringelemente aus Gleichung 2.32 nur die Distanzen zu den benachbarten Hilfspunkten zur Beschreibung der Interessensmerkmale berücksichtigt. [32].

Eine wichtige Voraussetzung um zuverlässige Interessensmerkmale bei allen genannten Trackingverfahren wiederzufinden, ist die Größe des Bildausschnittes. Beim KLT-Trackingverfahren steigert ein zu klein gewählter Bildausschnitt das Positionsrauschen, da in einem kleinen Bildbereich des Eingabebildes wenige und auch nicht robuste Interessensmerkmale durch Helligkeitsänderungen zu finden sind. Größere Bildausschnitte finden zwar eine große Anzahl der Interessensmerkmale und sind robuster gegenüber Ausreißern, beeinflussen jedoch erheblich die Rechenzeit. Bei diskontinuierlichen Bildsequenzen besteht in größeren Bildausschnitten die Gefahr der Driftprobleme.



Eine optimale Größe wurde in [33] mit einer Breite von 7 Pixeln und einer Höhe von 17 Pixeln gewählt.

Bei dem SIFT-Trackingverfahren beeinflusst die Anzahl und die Größe der Subfenster das Wiederfinden der Interessensmerkmale. Auch beim Surf-Trackingverfahren ist die Anzahl der Subfenster entscheidend für eine optimale Wiederfindung der Interessensmerkmale. Im nachfolgenden Diagramm werden bei variierender Anzahl und Größe der Subfenster die Trackingverfahren mit der Präzisionsgleichung aus Gleichung 3.1 evaluiert [26].

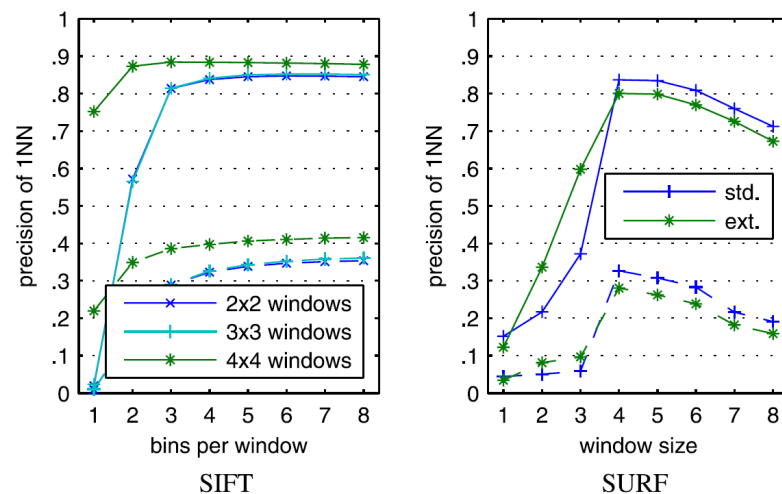


Abbildung 3.7: Präzision der Korrespondenzsuche bei variierender Fenstergröße [26]

Aus den Diagrammen wird deutlich, dass mit einer Subfenstergröße von  $4 \cdot 4$  Pixeln und 8 Subfenstern das SIFT-Trackingverfahren und das SURF-Trackingverfahren mit 4 Subfenstern bei einer Standardkonfiguration des Suchradius optimale Werte liefert [26]. Generell trägt eine hohe Anzahl der Subfenster des SIFT- und SURF- Trackingverfahrens zu einer Steigerung der Dimensionen der Deskriptor-Vektoren bei, die durch den Vergleich der Deskriptor-Vektoren eine Erhöhung der Rechenzeit erzielt.

### 3.3. Konklusion

Die Untersuchungen haben gezeigt, dass bei verändernden Aufnahmebedingungen die SURF- und SIFT-Trackingverfahren unter optimalen Einstellungen für das Wiederfinden der Interessensmerkmale am zuverlässigsten ist. Das SIFT-Trackingverfahren, das zwar robuste Interessensmerkmale detektiert und diese charakterisiert, erreicht hierbei die höchste Rechenzeit. Das schnelle BRISK-Trackingverfahren zeigt, dass nur unter bedingten Aufnahmebedingungen robust und zuverlässig Interessensmerkmale detektiert und wiedergefunden werden. Das KLT-Trackingverfahren, der im Gegensatz zu den deskriptiven Trackingverfahren die kürzeste Rechenzeit erzielt, reagiert auf starke Veränderungen der Aufnahmebedingungen sehr empfindlich. Mit Hilfe des optischen Flusses lassen sich die Interessensmerkmale in einem Bildausschnitt verfolgen, jedoch können verlorene Interessensmerkmale nicht wiedererkannt werden.

### 3.4. Auflistung der Fehlermerkmale

Aus Abschnitt 3.2.1. geht hervor, dass beim KLT-Trackingverfahren keine Deskriptoren verwendet werden und für jedes neu detektierte Interessensmerkmal eine neue ID zugewiesen wird. Die deskriptiven Trackingverfahren nutzen die Eigenschaftsbeschreibung der Interessensmerkmale, um Interessensmerkmale in nachfolgenden Bildern wiederzufinden. Hierbei können Mismatch-Fehler, wie aus dem Vergleich der Wiederfindbarkeit der Interessensmerkmale aus Abschnitt 3.1.3. und Abschnitt 3.2.2. hervorgeht, auftreten.

Des Weiteren konnte in Abschnitt 3.2.1. herausgefunden werden, dass durch diskontinuierlich Bildbewegungen Driftfehler und durch verändernde Bildgrauwerte Rauschfehler beim KLT- Trackingverfahren entstehen können. In Abschnitt 3.2.2. konnte für die deskriptiven Trackingverfahren herausgefunden werden, dass durch die genaue Positionermittlung der Interessensmerkmale Rauschfehler entstehen können. Durch Akkumulieren der Rauschfehler können infolgedessen auch Driftfehler bei den deskriptiven Trackingverfahren ausgelöst werden.

Durch den Vergleich der Eigenwerte des Shi-Tomasi-Eckendetektors beim KLT-Trackingverfahren, können wie aus Abschnitt 3.2.1. hervorgeht, Kantendriftfehler entstehen. Auch das BRISK- Trackingverfahren kann durch den einfachen Vergleich der Bildgrauwerte um ein Interessensmerkmal und der damit einhergehenden geringen Eigenschaftsbeschreibung der Interessensmerkmale Kantendriftfehler verursachen. Die SIFT- und SURF-Trackingverfahren, die das Interessensmerkmal durch die genaue Beobachtung der Pixelintensitäten der benachbarten Bildpunkte beschreibt und somit eine robustere Korrespondenzsuche ermöglicht, sind für Kantendriftfehler nicht anfällig.

Die Tabelle 3.3 stellt vereinfacht die Fehlermerkmale der jeweiligen Trackingverfahren dar.

	deskriptorlose Trackingverfahren	deskriptive Trackingverfahren		
Fehlermerkmale	KLT	SIFT	SURF	BRISK
ID-Generierung	Ja	Nein	Nein	Nein
Rauschfehler	Ja	Ja	Ja	Ja
Driftfehler	Ja	Ja	Ja	Ja
Mismatch-Fehler	Nein	Ja	Ja	Ja

Tabelle 3.3.: Auflistung der Fehlermerkmale der jeweiligen Trackingverfahren

### 3.5. Anwendung eines realen Feature Trackers

Für die nachfolgende Implementierung der Fehlermerkmale der Trackingverfahren in die Simulationsumgebung wurden die erkannten Fehlermerkmale aus den vorherigen Untersuchungen in einem realen Feature Tracker, dem DIP-Framework, untersucht. Hierbei wurden die Fehlerhäufigkeiten und die Fehlerausmaße beobachtet. Das zur Verfügung gestellte DIP-Framework besteht unter anderem aus diversen Trackingverfahren, die auf Bildsequenzen angewendet werden können. Für die relevanten Trackingverfahren in dieser Arbeit bestanden bereits für die Trackingverfahren Standardeinstellungen, die jedoch auch vom Anwender individuell angepasst werden können. In Abbildung 3.8 ist die GUI-Oberfläche des Programmes dargestellt, in der beispielhaft für die gegebene Bildsequenz gelbe KLT-Interessensmerkmale dargestellt sind. Mit Hilfe der realen Feature Tracker konnten sichtbare Eindrücke für die entstandenen Fehlermerkmale gewonnen werden. Aus den empirischen Ermittlungen konnten für die verschiedenen Trackingverfahren im nachfolgenden Kapitel die Standardparameter zur Häufigkeit und zum Ausmaß der Fehlermerkmale abgeschätzt werden.

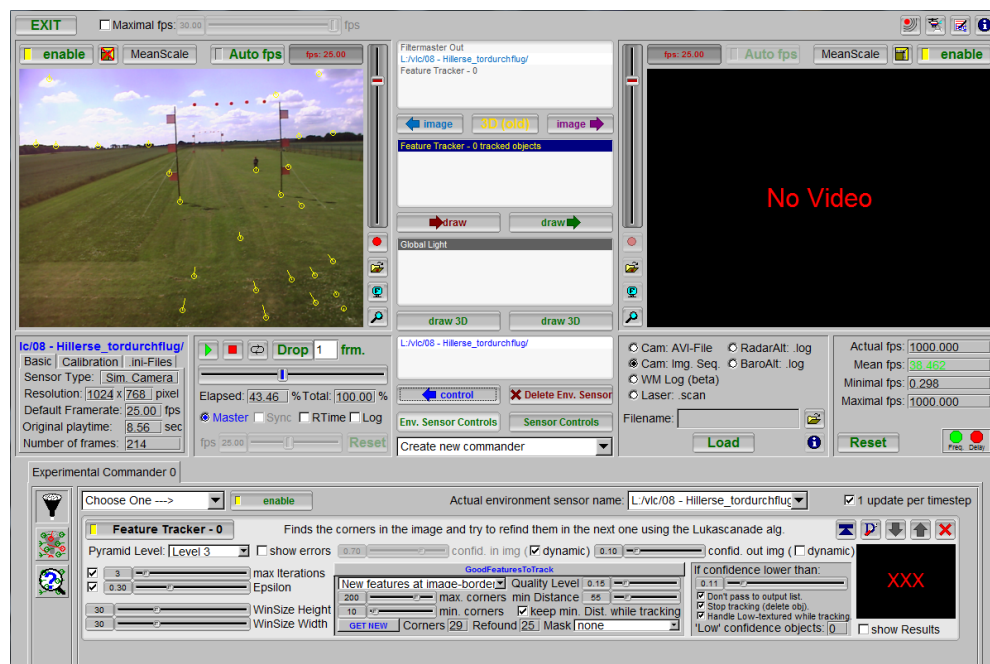


Abbildung 3.8.: GUI-Oberfläche des DIP-Frameworks

## 4 Simulationsumgebung

#### 4.1. Beschreibung der vorhandenen Simulation

Der Feature Tracker Simulator in Abbildung 4.1 erfasst aus einer virtuellen Lochkamera an einem simulierten Luftfahrzeug [Abbildung 4.2] eine virtuelle dreidimensionale Umgebung, die durch eine Punktwolke repräsentiert ist. Anschließend werden aus der Punktwolke zufällige Punkte entnommen und im Visualisierungsfenster als besondere Interessensmerkmale dargestellt [Abbildung 4.3]. Die dargestellten Interessensmerkmale sind fehlerfrei und werden zur Entwicklung eines möglichst realitätsnahen Feature Trackers mit Fehlermerkmalen erweitert.

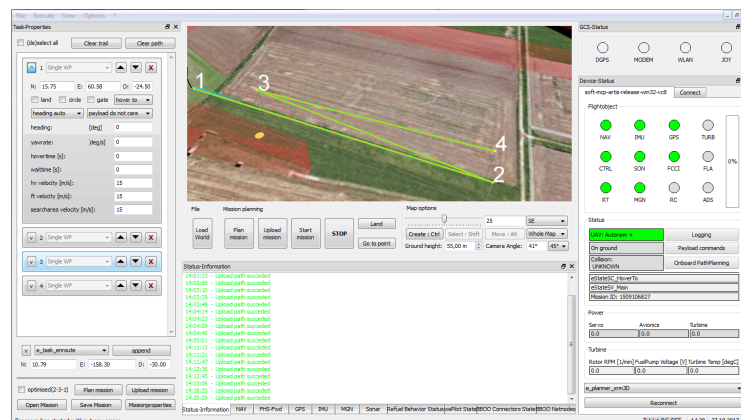


Abbildung 4.1.: Programmoberfläche des Feature Tracker Simulators



Abbildung 4.2.: Luftfahrzeug im Feature Tracker Simulator

### 4.1.1. Feature Tracker Simulator

Die Interessensmerkmale unterscheiden sich zwischen Landmarken und natürlichen Interessensmerkmalen, die in einem Visualisierungsfenster dargestellt werden. Detektierte Landmarken sind Interessensmerkmale, die eine bekannte Position im Bezugskoordinatensystem haben, mit der eine eindeutige Lage- und Orientierungsbestimmung des Luftfahrzeuges berechnet werden kann. In der Simulation können Flugrouten manuell auf der Simulationskarte erstellt werden [Abbildung 4.1]. Durch eine Programmausführung wird die erstellte Flugroute mit dem simulierten Luftfahrzeug erflogen und die identifizierten Interessensmerkmale des Feature Trackers im Visualisierungsfenster dargestellt [Abbildung 4.3].

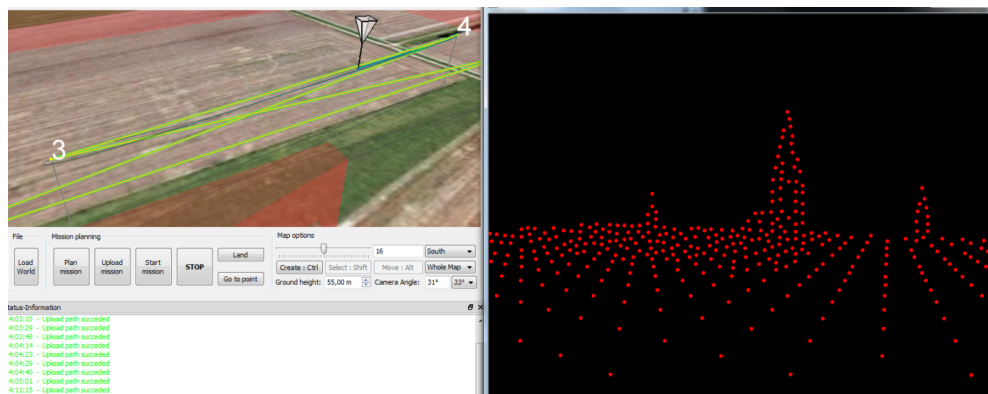


Abbildung 4.3.: Visualisierungsfenster des Lochkameramodells mit ideal identifizierten Interessensmerkmalen

Für die Ausführung des Feature Tracker Simulators wurden durch Eingabebefehle in der Windows-Kommandozeile das Luftfahrzeug und die Simulationskarte ausgewählt. Durch Aufrufen eines GUI-Fensters wurde eine Verbindung zur Simulationssoftware hergestellt, der für den Empfang, der visuellen Darstellung des Luftfahrzeuges und der Simulationsumgebung zuständig ist.

Für die Darstellung der Interessensmerkmale im Visualisierungsfenster wurde der Feature Emulator ausgeführt.

Die Flugstrecken in der Simulationsumgebung konnten durch manuell erstellte Wegpunkte abgefliegen werden [Abbildung 4.1].

### 4.1.2. Objekterkennung mittels des Lochkameramodells

Die Objekte in der Simulationsumgebung bestehen aus Positionskoordinaten, die im Weltkoordinatensystem beschrieben werden. Für eine Abbildung in der Bildebene, werden Rotations- und Translationstransformationen zwischen dem Welt-, Kamera- und Pixelkoordinatensystem bestimmt. Die transformierten Objektkoordinaten im Kamerakoordinatensystem, lassen sich wie folgt durch die mathematische Beschreibung mit Gleichung 4.1 berechnen [4].

$$\underline{X}_c = \underline{R}_{zyx_{wc}} (\underline{X}_c - t_{wc}) \quad (4.1)$$

Die transformierten Objektkoordinaten im Kamerakoordinatensystem, werden anschließend durch eine perspektivische Projektion in die Bildebene gebracht. Die perspektivische Projektion des Lochkameramodells in Abbildung 4.4, besteht aus der Bild-, Brenn- und Objektebene. Ein ausgehender gerader Lichtstrahl von einem Punkt aus der Objektebene, der durch die infinitesimale Lochblende geht, trifft hierbei die Bildebene an einem Punkt. Die mehrfachen Punkte auf der Bildebene erzeugen ein zweidimensionales Bild. Um die dreidimensionalen Weltkoordinaten in ein zweidimensionales Bild zu transformieren, werden hierbei die Bildkoordinaten mit der Gleichung 4.2 skaliert.

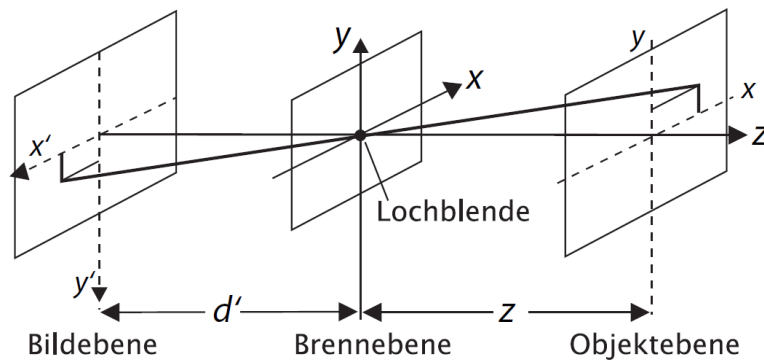


Abbildung 4.4.: Mathematisches Prinzip der Lochkamera [4]

$$\underline{X}_c = \begin{bmatrix} x_p & y_p & z_p \end{bmatrix}^T \rightarrow \tilde{\underline{X}}_c = \begin{bmatrix} \frac{x_p}{z_p} & \frac{y_p}{z_p} & 1 \end{bmatrix}^T \quad (4.2)$$

Für eine elegante und einheitliche Berechnung des Abbildungsprozesses, werden homogene Koordinaten eingesetzt, die aus einem 4 dimensional Spaltenvektor und einer 4-4-Matrix bestehen. Die Matrix ist hierbei aus mehreren Transformationsmatrizen zusammengesetzt, die aus extrinsischen und intrinsischen Kameraparametern bestehen. [4]

Mit den extrinsischen Kameraparametern werden die Rotationen und Translationen beschrieben. Die intrinsischen Kameraparameter bestimmen die kameraspezifischen Parameter, die aus der Lage der optischen Achse, der Brennweite und der Skalierungsgröße der Bildebene bestehen. Die Gleichung 4.3 bestimmt mit Hilfe der Transformationsmatrizen den Bildpunkt in der Bildebene, die Matrix  $\underline{\underline{C}}$  beschreibt hierbei die Bildverschiebung, die Matrix  $\underline{\underline{P}}$  die perspektivische Projektions und die Matrix  $\underline{\underline{S}}$  die Skalierung des Bildes. Eine ausführliche Formulierung der Matrizen ist im Anhang A.1. zu finden [4].

$$\underline{X}_p = \underline{\underline{C}} \cdot \underline{\underline{S}} \cdot \underline{\underline{P}} \cdot \underline{R}_{zyx} \cdot \underline{T} \quad (4.3)$$

Die zufällig entnommenen Punkte aus der Punktwolke des Feature Tracker Simulators werden auf die Bildebene projiziert und als Interessensmerkmale im Visualisierungsfenster wiedergegeben. Die Interessensmerkmale bestehen hierbei aus der Position der Bildebene, der ID-Zuordnung, den Kameraparametern und der Art der besonderen Interessensmerkmale, die zwischen Landmarken und Interessensmerkmalen unterscheidet.

## 4.2. Extrahieren der Fehlermerkmale in die Simulation

Der Feature Tracker Simulator nutzt für die Darstellung der Punktwolke die OpenCV Programm-bibliothek. In der Bildebene des Lochkameramodells, die in der Simulation als Visualisierungsfenster dargestellt ist, werden die zufällig ausgewählten Punkte aus der Punktwolke als Landmarken und Interessensmerkmale abgebildet. Die Bildebenen folgen hierbei einem iterativen Aufbau. Für die Realisierung der Fehlermerkmale aus dem vorherigen Kapitel werden jeweils überladene Funktionen im Programmcode des Feature Tracker Simulators erstellt. In der bereits idealen Simulation, werden die idealen Interessensmerkmale mit den überladenen Fehlerfunktionen zu künstlichen Fehlermerkmalen modelliert.

Im Folgenden werden die jeweiligen überladenen Fehlerfunktionen vorgestellt. Für eine möglichst realistische Simulation werden zu den Fehlermerkmalen Häufigkeiten und Gewichtungen aufgestellt.

### 4.2.1. Inkrementelle ID-Generierung

Der ideale Feature Tracker Simulator besitzt bereits für die generierten natürlichen Interessensmerkmale und Landmarken feste ID-Zuordnungen.

Für das KLT-Trackingverfahren, das Interessensmerkmale nur in der aktuellen Bildebene wiederfindet, wurde zur inkrementellen ID-Generierung eine überladene Funktion erstellt, die als Eingabeparameter die sichtbaren Interessensmerkmale erhält. Innerhalb der überladenen Funktion wurden für die Interessensmerkmale neue IDs erzeugt, die mit einem Startwert von 1 inkrementell hochgezählt werden. Für die Landmarken, die als robuste Interessensmerkmale gelten, wurden durch boolesche Abfragemethoden bereits erkannte Landmarken aus vorherigen Bildausschnitten mit der alten ID-Zuordnung versehen.



Der Rückgabewert der überladenen Funktion liefert die neuen ID-Zuordnungen der besonderen Interessensmerkmale. Im Ablaufdiagramm in Abbildung 4.5 sind die wesentlichen Programmschritte nochmals dargestellt.

Für die Darstellung der ID-Zuordnung und der Unterscheidung von Landmarken und Interessensmerkmalen im Visualisierungsfenster wurden Bildverarbeitungsbefehle aus OpenCV verwendet. Die deskriptiven Trackingverfahren, die mit Charakteristiken der Interessensmerkmale arbeiten, können verlorene Interessensmerkmale in einer Bildsequenz wiederfinden. Die Funktion der inkrementellen ID-Generierung wird somit für die deskriptiven Trackingverfahren nicht aufgerufen.

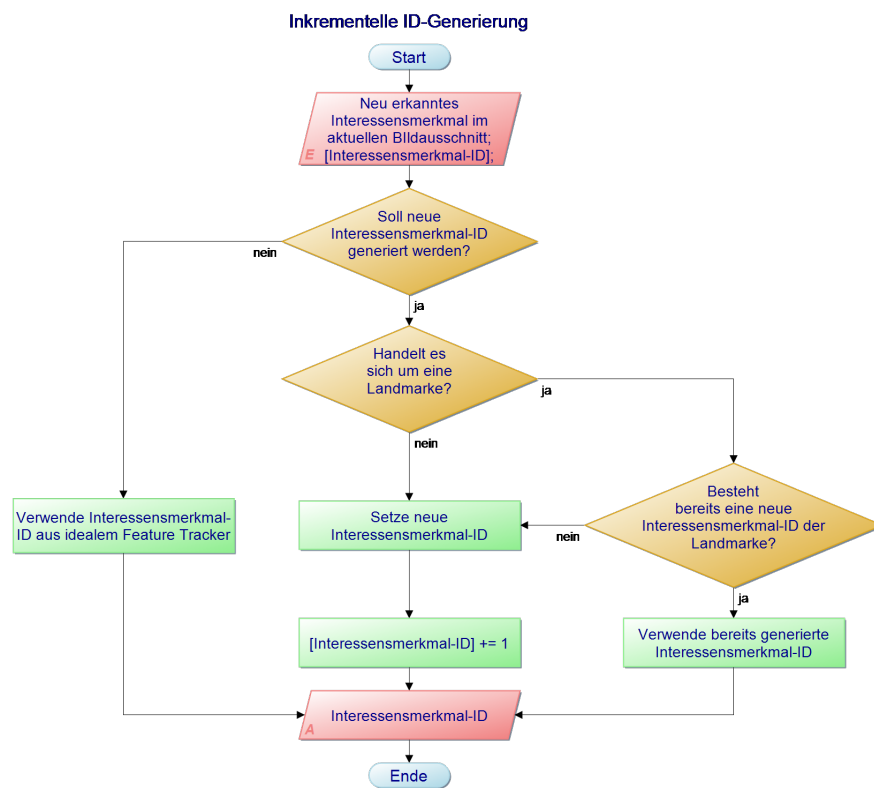


Abbildung 4.5.: Funktion zur inkrementellen ID-Generierung



### 4.2.2. Rauschen

Für die Rauschentwicklung der Interessensmerkmale wurde eine überladene Funktion erstellt, die als Eingabeparameter die besonderen Interessensmerkmale aus der vorherigen und aktuellen Bildebene und den Parameter für die Auftrittswahrscheinlichkeit des Rauschfehlers erhält.

In der überladenen Funktionsdefinition wurde zur Rauschentwicklung eine bivariate Normalverteilung eingesetzt, die in den Filter-Operatoren der Trackingverfahren zur Erstellung der stetigen Bildfunktion und Erkennung von besonderen Interessensmerkmalen genutzt wird. Die Diskretisierung der stetigen Bildfunktion in Bildpunkte hat zur Folge, dass die ermittelte Position des Interessensmerkmals zwischen zwei Pixeln liegen kann und eine Positionsabweichung verursacht. Die Abbildung 3.6 aus Abschnitt 3.2.2. stellt hierbei die Funktion der Filter-Operanten dar. Ein Interessensmerkmal ist durch das Maximum der Funktion gekennzeichnet.

Die genaue Einstellung der erstellten bivariaten Normalverteilung ist in Abschnitt 4.3.1. vorzufinden. Die künstlich erzeugte Fehlerposition wird anschließend durch eine Eintrittswahrscheinlichkeit auf ein beliebiges Interessensmerkmal übertragen.

Um die Fehlerposition der Interessensmerkmale in nachfolgenden Bildebenen zu erhalten, wurde eine euklidische Distanz zwischen der idealen Position und der generierten Fehlerposition der Interessensmerkmale im aktuellen Bild bestimmt. Für die Erhaltung der Fehlerposition der Interessensmerkmale in nachfolgenden Bildern, wurden die Abweichungsdistanzen mit der entsprechenden Interessenmerkmal-ID zugeordnet und diese in nachfolgenden Bildern bei wiedergefundenen Interessensmerkmalen übernommen. Eine direkte Übernahme der Fehlerposition, ohne Berücksichtigung der Abweichungsdistanzen in nachfolgenden Bildern, bewirkte trotz Fortbewegung des simulierten Luftfahrzeuges zu keiner Aufnahme neuer Punkte im Visualisierungsfenster bei einer verändernden virtuellen Umgebung. Des Weiteren wurden die Positionen der Interessensmerkmale nur durch zufällig auftretende Rauschfehler im Bild verändert. Durch das Mitliefern der Abweichungsdistanzen der Interessensmerkmale konnten die vorherigen Fehlerpositionen auf die iterativ erzeugten Idealpositionen im aktuellen Bild rekonstruiert werden.

Durch die Positionsaktualisierung der Interessensmerkmale konnten auf vorhandenen Fehlerpositionen weitere akkumulierende Fehlerpositionen erzeugt werden. Infolgedessen liefert die überladene Funktion die erzeugten Fehlerpositionen für die Visualisierung und die euklidische Distanz der Fehlerpositionen.

Die implementierte Rauschentwicklung beschränkt sich hierbei nur auf die Fehlberechnungen der Filter-Operatoren. Schlechte Bildaufnahmen durch beispielsweise falsche Kalibrierungen der Bildsensoren verstärken das Rauschen und können durch Einstellen der Kameraparameter ebenfalls in der Simulation mitberücksichtigt werden. Die wesentlichen Programmschritte der Rauschentwicklung sind im Ablaufdiagramm in Abbildung 4.6 abgebildet.

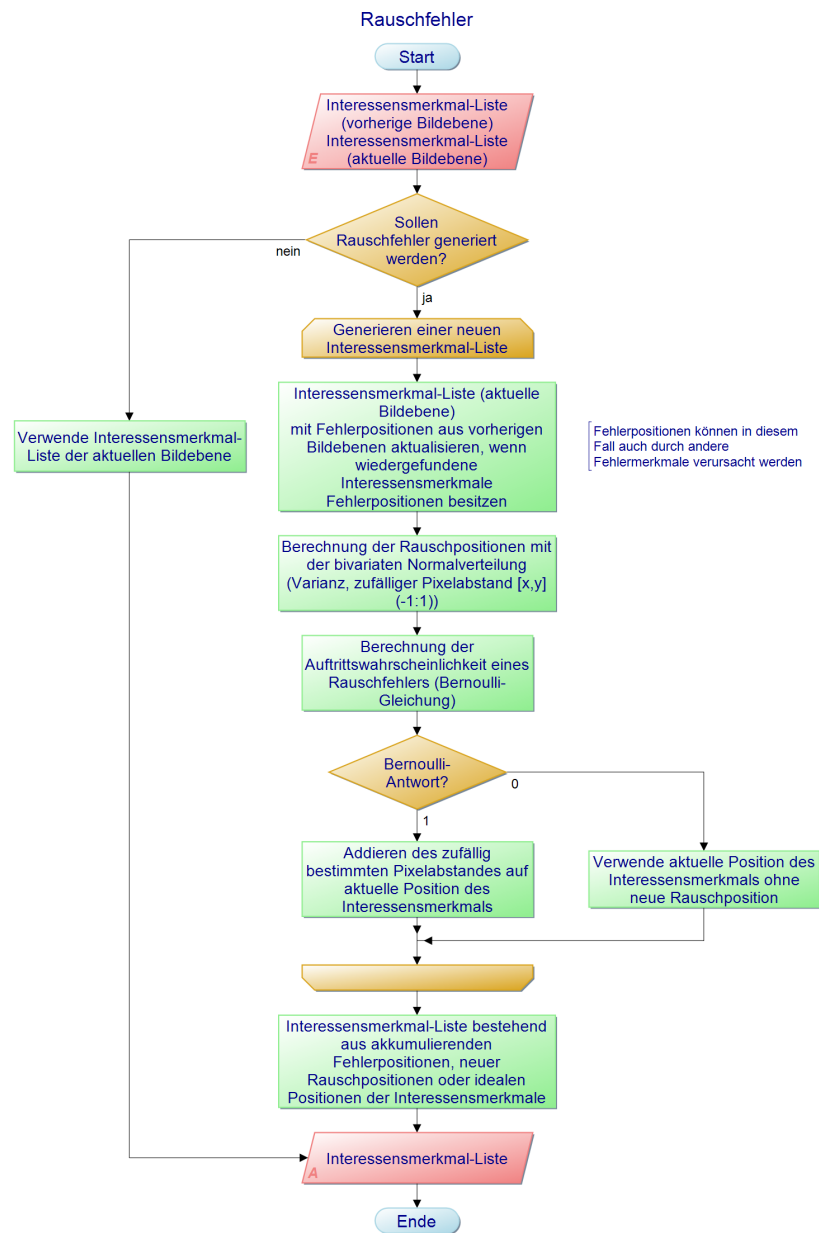


Abbildung 4.6.: Fehlerfunktion der Rauschfehler

### 4.2.3. Driftfehler

Driftfehler entstehen durch Akkumulationsfehler des Rauschfehlers oder durch schnelle Bildbewegungen beim KLT Trackingverfahren. Die Driftfehler, die durch schnelle Bildbewegungen verursacht werden, wurden in der überladenen Funktion der Rauschentwicklung ergänzt. Unter realen Bedingungen werden die Annahmen des KLT Trackingverfahrens aus Kapitel 3 verletzt, die zu falschen Transformationen des Bildausschnittes führen. Durch die Fortbewegungen des Luftfahrzeuges in der Simulation entstehen diskontinuierliche Bildbewegungen, die Driftfehler der Interessensmerkmale begünstigen. Für die Generierung der Driftfehler wurde eine weitere euklidische Distanz der Positionen zwischen allen vorherigen und aktuellen korrespondierenden Interessensmerkmalen genutzt, um die Strecke der Bildpunktbewegungen zu bestimmen. Mit einem arithmetischen Mittel wurde zunächst die Durchschnittsgröße der Bildpunktbewegungen bestimmt. Interessensmerkmal mit einer überdurchschnittlichen Bildpunktbewegung wurden als potenzielle Driftfehler erkannt.

Für eine realistische Simulation der Driftfehler wurden diese mit Wahrscheinlichkeiten der Normalverteilung und Bernoulli-Verteilung aus den potenziellen Driftfehlern ausgewählt, die im Abschnitt 4.3. näher beschrieben sind. Für die ausgewählten Driftfehler werden anschließend aus den Bildpunktbewegungen zufällige Längen entnommen und auf die Interessensmerkmale aufaddiert. Die entstandenen Driftfehler werden in der euklidischen Distanz der Fehlerabweichung für die nachfolgenden Bildebenen mitberücksichtigt. In Abbildung 4.7 sind die wesentlichen Funktionschritte in einem Ablaufdiagramm dargestellt.

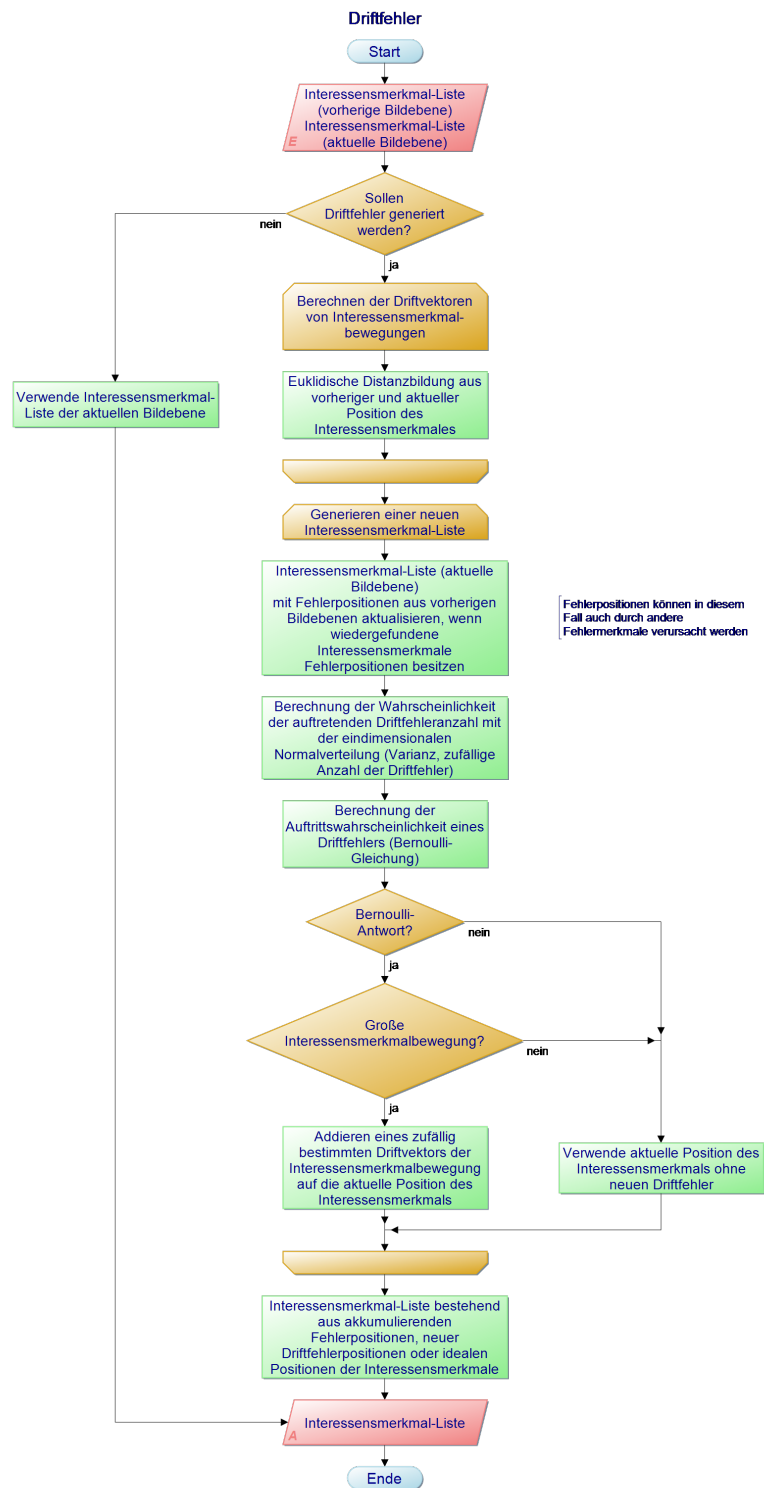


Abbildung 4.7.: Fehlerfunktion der Driftfehler

#### 4.2.4. Kantendriftfehler

Die überladene Funktion des Kantendriftfehlers, der durch nicht robuste Interessensmerkmale beim KLT Trackingverfahren zustande kommt, erhält die Eingabeparameter der Interessensmerkmale aus der vorherigen und aktuellen Bildebene und den Parameter für die Häufigkeit des Kantendriftfehlers. Zur Generierung der Kantendriftfehler, wurde mit der Hough-Transformation eine Funktion entwickelt, die aus Interessensmerkmalen Kanten in der Bildebene wiedererkennt.

In der Bildebene wird mit einem Zufallsgenerator ein beliebiges Interessensmerkmal ausgewählt, um den in einem definierten Radius die benachbarten Interessensmerkmale lokalisiert werden. Die lokalisierten Interessensmerkmale werden mit der Hough-Transformation auf Kanten untersucht. Die Hough-Transformation basiert auf der Hesseschen Normalform der Geradengleichung in Gleichung 4.4 [3].

$$r_p = x_p \cos(\theta) + y_p \sin(\theta) \quad \text{mit: } [0, 2\pi] = \{\theta \in \mathbb{R} | 0 \leq \theta \leq 2\pi\} \quad (4.4)$$

Hierbei werden die Polarkoordinaten im Parameterraum gesucht, in der sich Parametergeraden schneiden. Für eine Hough-Transformierung eines ausgewählten Interessensmerkmals wird die aktuelle Bildkoordinate aus der Bildebene als eine Funktion des Winkels in Abhängigkeit des Radius im Parameterraum dargestellt. Eine Bildkoordinate des Interessensmerkmals erzeugt somit im Parameterraum keine Gerade, sondern eine sinusförmige Funktion, deren Amplitude und Phase von der jeweiligen Bildkoordinate abhängt. In Abbildung 4.8 sind beispielhaft 4 Bildgeraden dargestellt, die im Parameterraum sinusförmige Kurven annehmen [3].

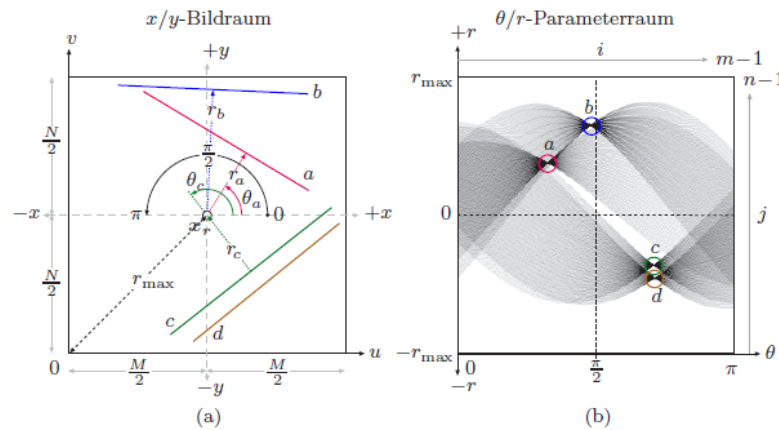


Abbildung 4.8.: Hough-Transformation der 4 Bildgeraden [3]

Für die Erzeugung von Kantendriftfehlern werden die Bildkoordinaten der ausgewählten Interessensmerkmale transformiert und mit einer diskreten Winkelschrittweite im Parameterraum dargestellt. Der Parameterraum wird anschließend nach maximalen Überschneidungen der transformierten Bildkoordinaten untersucht. Mit der höchsten Übereinstimmung einer transformierten Bildkoordinate im Parameterraum und der Überschreitung eines festgelegten Schwellwertes von  $t = 3$ , wird ein Interessensmerkmal auf einer Kante erkannt. Die Winkelschrittweite wurde mit  $\theta = 10^\circ$  festgelegt. Um Diskretisierungsfehler der Winkelangaben zu berücksichtigen, wurde für den Vergleich der übereinstimmenden Geraden im Parameterraum eine Toleranz von  $\pm 2\%$  ge-

wählt.

Die erkannten Interessensmerkmale auf Kanten werden anschließend mit einer Eintrittswahrscheinlichkeit aus einer Normalverteilung und einer Bernoulli-Verteilung versehen, die potenzielle Interessensmerkmale für den Kantendriftfehler bestimmt. Eine ausführliche Beschreibung der Eintrittswahrscheinlichkeit für das Kantendriften ist in Abschnitt 4.3 zu finden. Zur Generierung der Positionsabweichung für das Kantendriften der Interessensmerkmale, wurden mit einem Zufalls-generator aus einem annehmbaren Abweichungsbereich die Positionsfehler bestimmt. Mit der Fehlerfunktion des Kantendriften werden die Interessensmerkmale mit den Fehlerpositionen und die ID-Zuordnungen der Interessensmerkmale mit Kantendriftfehlern ausgegeben, um in nachfolgenden Bildern anhaltende Kantendriffs fortzuführen.

Die Fehlerposition und die ID-Zuordnung der fehlerhaften Interessensmerkmale werden nach dem First-In-Last-Out-Queuing-Verfahren (FILO) in einer dynamischen Liste abgespeichert. Zur Anpassung der dynamischen Liste an die Häufigkeit der Kantendriftfehler werden die letzten Einträge der Kantendriftfehler bei einer kleiner gewordenen Häufigkeit gelöscht und bestehende Kantendriftfehler weiter ausgeführt. Die Liste wird mit neuen Einträgen an erster Listenstelle ergänzt, wenn eine größere Häufigkeit vorliegt. Die Abbildung 4.9 stellt das Verfahren zur dynamischen Liste vereinfacht dar.

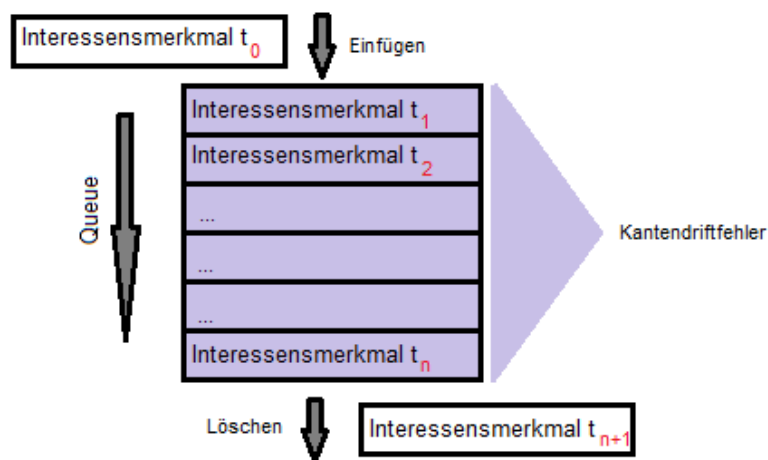


Abbildung 4.9.: Dynamische Liste für die Interessensmerkmale mit Kantendriftfehlern

Das Ablaufdiagramm in Abbildung 4.10 zeigt die wesentlichen Funktionsschritte.

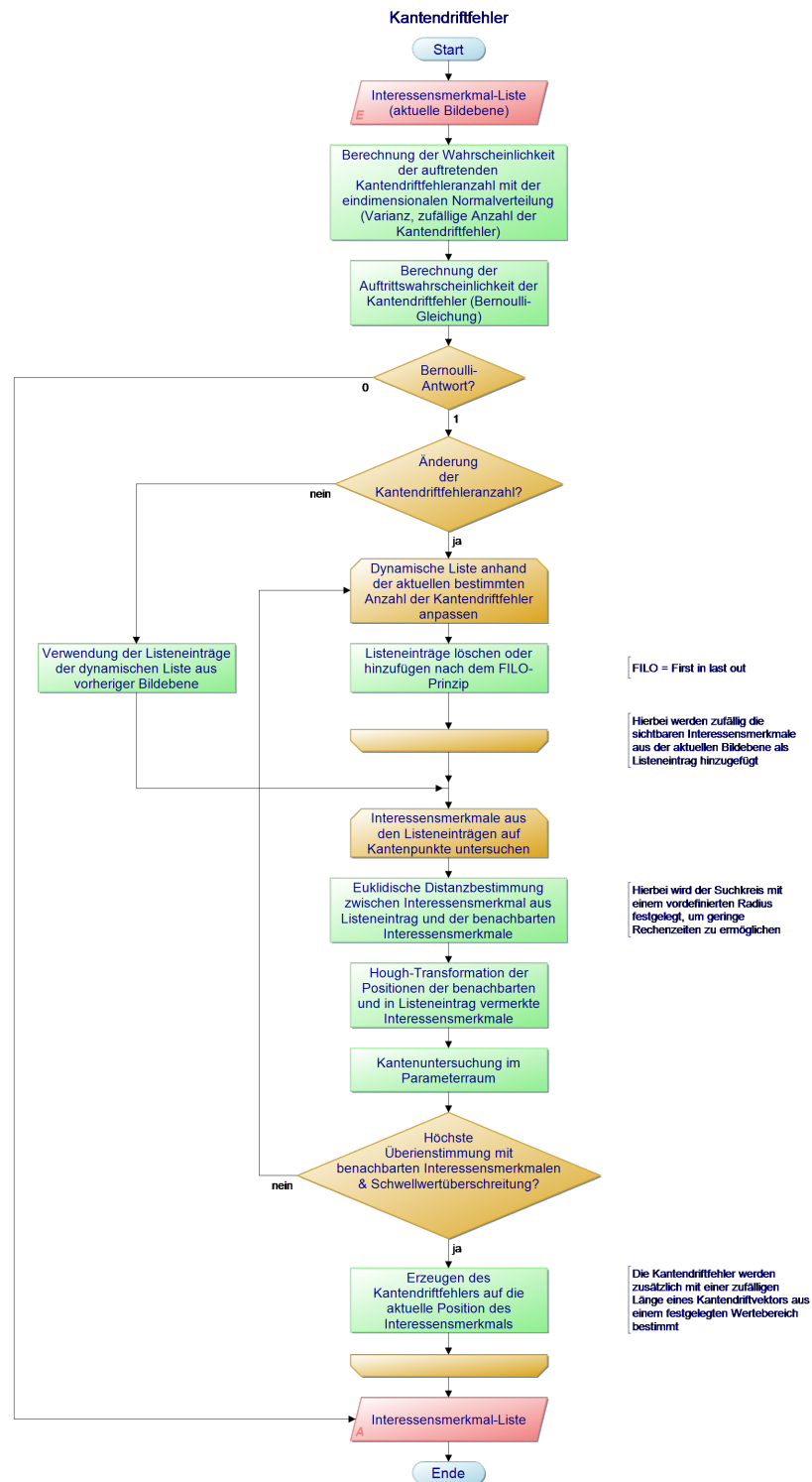


Abbildung 4.10.: Fehlerfunktion der Kantendriftfehler

### 4.2.5. Mismatches

Zur Mismatch-Bildung der deskriptiven Trackingverfahren wurde eine überladene Funktion erstellt, die als Eingabeparameter die ID-Zuordnung der Interessensmerkmale erhält. Mit einer Eintrittswahrscheinlichkeit der Bernoulli-Gleichung in Gleichung 4.5 wird zunächst das Auftreten eines Mismatch-Fehlers bestimmt.

$$f(x) = \begin{cases} p & , \quad x = 1 \\ p - 1 & , \quad x = 0 \end{cases} \quad (4.5)$$

Liefert die Eintrittswahrscheinlichkeit einen positiven Wert, so wird ein beliebiges Interessensmerkmal der aktuellen Bildebene ausgewählt. Ist das Interessensmerkmal eine Landmarke, werden weitere Interessensmerkmale untersucht, die keine Landmarken sind. Landmarken gelten als robuste Interessensmerkmale und werden in Bildsequenzen immer wiedergefunden. Für das ausgewählte Interessensmerkmal wird durch den Zufallsgenerator eine neue ID-Zuordnung aus den nicht sichtbaren Interessensmerkmalen ohne Landmarken gewonnen. Die neue ID-Zuordnung des Interessensmerkmals wird anschließend von der Fehlerfunktion zurückgegeben. Das Ablaufdiagramm für die Mismatch-Bildung in Abbildung 4.11 stellt die wesentlichen Funktionsabschnitte dar.



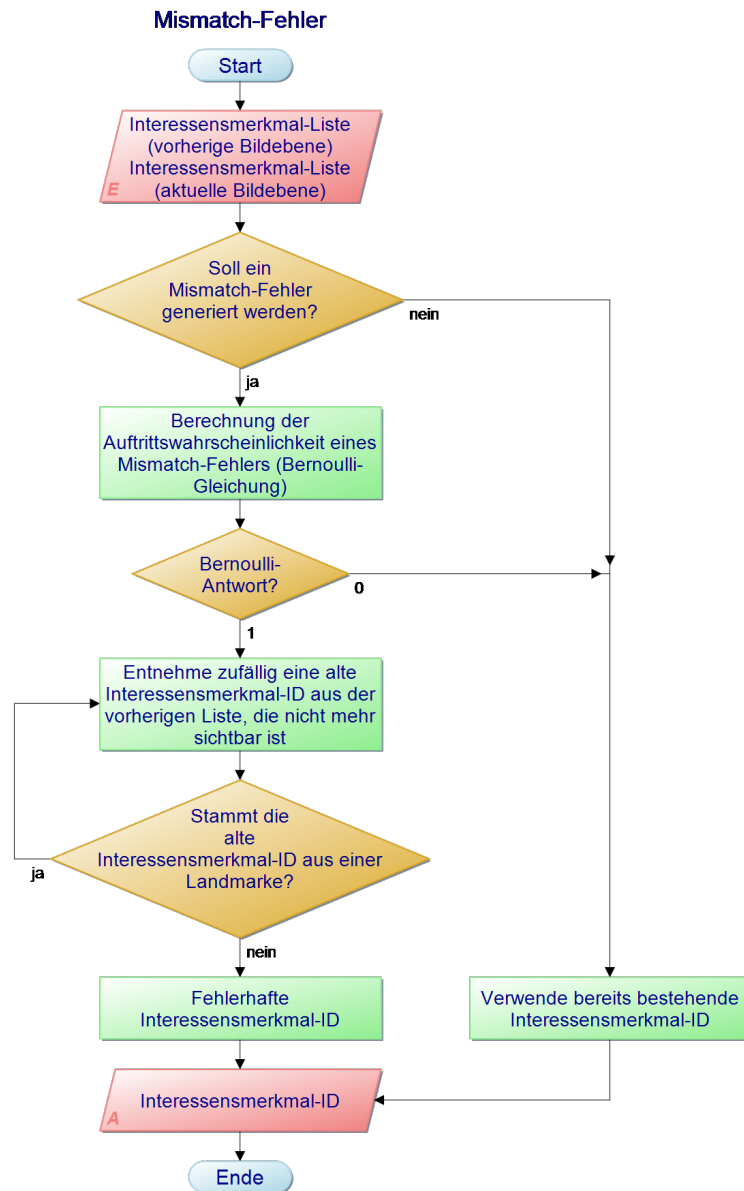


Abbildung 4.11.: Fehlerfunktion zur Mismatch-Bildung

#### 4.2.6. Rechenzeit

Für die Simulation der verschiedenen Rechenzeiten der Trackingverfahren wurden aus der Windows-Standardbibliothek Wartefunktionen in einer Parameterklasse verwendet. Um die Berechnungszeit eines Bildpaares zu realisieren, wurden die Wartefunktionen zwischen zwei Bildern ausgeführt. Die Parametereinstellungen der Rechenzeiten finden sich in Abschnitt 4.3. wieder.

## 4.3. Häufigkeit der Fehlermerkmale

### 4.3.1. Rauschen

Die bivariate Normalverteilung stellt eine Nachbildung der Filteroperatoren dar, mit der künstlich eine verrauschte Position erzeugt wird. Aus Kapitel 3 geht bereits hervor, dass durch die Diskretisierung der Filteroperatoren Abweichungen um  $\pm 1$  Pixel entstehen können. Um die Häufigkeit der Abweichungen zu beeinflussen, kann durch Variieren der Varianz die Amplitude der bivariaten Normalverteilung verändert werden, die eine Änderung der Verteilungsdichte bewirkt. Um aus der bivariaten Normalverteilung eine verrauschte Position zu erhalten, wird diese auf ein Interessensmerkmal gelegt, indem das Maximum der bivariaten Normalverteilung auf das Interessensmerkmal verschoben wird. Mit einem Zufallsgenerator werden zunächst reelle Zahlen aus dem Intervall  $[x, y] [-1, 1]$  für die Koordinatenrichtungen entnommen, die anschließend für die bivariate Normalverteilung verwendet werden. Der Funktionswert liefert die Wahrscheinlichkeitsdichte für den zufällig ausgewählten nicht diskretisierten Bildpunkt.

Die berechnete Wahrscheinlichkeitsdichte wird im Anschluss mit einer idealisierten Wahrscheinlichkeitsdichte gewichtet. [Gleichung 4.6] Die idealisierte Wahrscheinlichkeitsdichte wird ebenfalls mit einer bivariaten Normalverteilung bestimmt, die ab einer Pixelgröße von  $\pm 0,5$  infinitesimale Werte liefert und somit keinen Beitrag zur Rauscherzeugung mehr leistet. In Abbildung 4.12 ist die idealisierte und veränderliche bivariate Normalverteilung als zweidimensionaler Plot dargestellt.

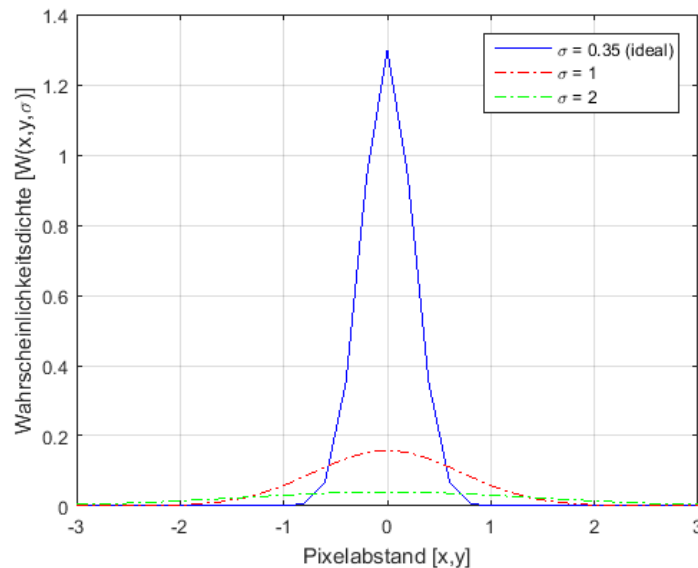


Abbildung 4.12.: Rauscherzeugung mit der bivariaten Normalverteilung

$$\tilde{W}(x, y, \sigma) = \frac{W_{\text{variable}}(x, y, \sigma)}{W_{\text{ideal}}(x, y, \sigma)} \quad \text{mit:} \quad W(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{x^2 + y^2}{2\sigma^2}\right)} \quad (4.6)$$

Mit der gewichteten Wahrscheinlichkeitsdichte folgt anschließend mit der Bernoulli-Verteilung die Berechnung des Fehlerauftretens. Bei einer beispielsweise großen Varianz wird durch die große Streuungsfläche eine stark verrauschte Position erzeugt, die für eine große Auftrittswahrscheinlichkeit sorgt.

Mit der bestimmten Auftrittswahrscheinlichkeit wird dann das Fehlerauftreten bestimmt. Hierbei bestimmt der Wert 1 ein Fehlerauftreten und der Wert 0 kein Auftreten eines Fehlers. Bei einem Fehlerauftreten wird der reelle Abstand diskretisiert und auf die ideale Position des Interessenspunktes aufaddiert, wenn der Abstand von 0,5 Pixeln überschritten wird. Da alle Trackingverfahren Filteroperatoren verwenden und diese Normalverteilungen zur Erkennung als auch zur Beschreibung verwenden, wird für jedes Trackingverfahren ein Rauschen generiert.

#### 4.3.2. Driftfehler

Für das Auftreten von Driftfehlern der Interessensmerkmale können keine genauen Aussagen getroffen werden. Lediglich bei großen Bildbewegungen ist insbesondere das KLT-Trackingverfahren anfällig für Driftfehler. Um eine möglichst realistische Anzahl der Driftfehler zu erzielen, wurde eine eindimensionale Normalverteilung gewählt. Hierbei wird nur die positive Abszisse mit den natürlichen Zahlen betrachtet, die eine Anzahl der Driftfehler angibt. Bei einer Anzahl von 0 Driftfehlern besteht die größte Wahrscheinlichkeitsdichte, die bei aufsteigender Fehleranzahl der Driftfehler mit einem Verlauf der Normalverteilung abnimmt. Um aus der Wahrscheinlichkeitsdichte eine Aussage über das Fehlerauftreten zu erhalten, wird eine Gewichtung mit einer idealisierten eindimensionalen Normalverteilung eingeführt. Diese wird ideal, sobald für das Auftreten eines Driftfehlers eine infinitesimale Wahrscheinlichkeitsdichte vorherrscht und diese das Auftreten eines Driftfehlers nicht ermöglicht. Mit einer verstellbaren Varianz kann hierbei die Streuung der Wahrscheinlichkeitsdichte beeinflusst werden. Mit einem Zufallsgenerator wird eine Anzahl der Driftfehler aus der Abszisse entnommen. Hierbei beschränkt sich die maximale Anzahl auf 10 Driftfehler, da eine größere Anzahl infinitesimale Wahrscheinlichkeiten liefert und diese für die Berechnung der Auftrittswahrscheinlichkeit der Driftfehler vernachlässigt werden kann. Die Abbildung 4.13 stellt die idealisierte und veränderliche Normalverteilung dar.

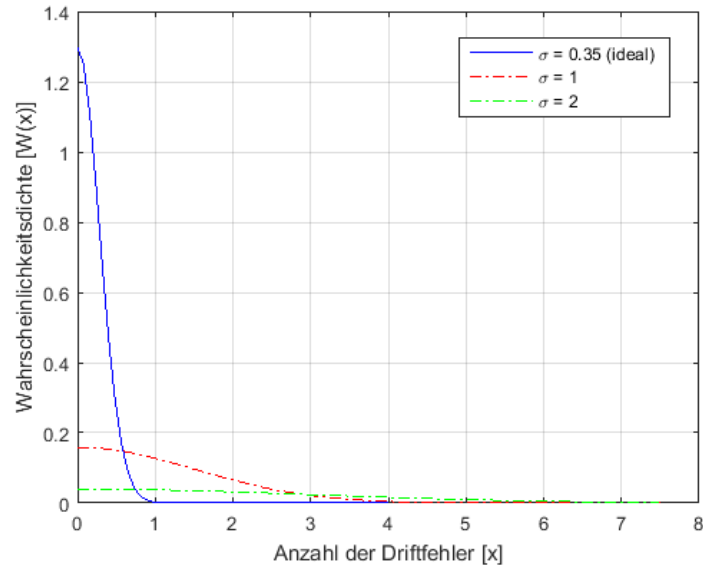


Abbildung 4.13.: Erzeugen der Driftfehler mit der eindimensionalen Normalverteilung

Aus der gewichteten Wahrscheinlichkeitsdichte wird wie bei der Rauscherzeugung mit der Bernoulli-Verteilung das Fehlerauftreten bestimmt.

### 4.3.3. Kantendriftfehler

In Anlehnung an die ermittelte Auftrittswahrscheinlichkeit der Driftfehler wird bei den Kantendriftfehlern in gleicher Weise vorgegangen, da die Driftfehler speziell auf einer Kante stattfinden und somit nur eine Abhängigkeit der Bildstruktur bilden. Aus Beobachtungen des DIP Frameworks und den Vergleichen der Trackingverfahren aus Kapitel 3 konnte festgestellt werden, dass das KLT-Trackingverfahren stark auf auffällige Kanten reagiert.

Der BRISK-Algorithmus reagiert ebenfalls empfindlich auf Kanten. Eine scharfe unveränderte lange Kante hat zur Folge, dass mehrere Interessensmerkmale auf dieser Kante liegen. Durch den geringen Informationsgehalt der Deskriptoren und den deterministischen Vergleichsprozess der Hilfspunkte kann dies zu einem falschen Vergleich führen. Durch die gleichen Interessensmerkmale kann dies für den Betrachter als Kantendrifting wahrgenommen werden.

#### 4.3.4. Mismatches

Durch den Vergleich der Deskriptor-Elemente können Interessensmerkmale wiedergefunden werden. Die robuste Wiederfindung von Interessensmerkmalen hängt von der Anzahl der Deskriptor-Elemente ab. Mit den 128 Deskriptor-Elementen des SIFT-Deskriptors können genauere Vergleiche erzielt werden. Der Surf Algorithmus der aus 64 Deskriptor-Elementen besteht, findet Interessensmerkmale zuverlässig. Durch den geringen Informationsgehalt des BRISK-Trackingverfahrens können beim Vergleich der Interessensmerkmale, insbesondere bei sehr ähnlichen Bildstrukturen Mismatches entstehen. Für das KLT-Trackingverfahren wurde die Funktion deaktiviert, da durch die Verfolgung der Interessensmerkmale neue IDs inkrementell generiert auftreten. Aus dem Kapitel 3 und mit der Beobachtung aus dem DIP-Framework wurden für die Trackingverfahren die Parameter in Tabelle 4.1 festgelegt.

#### 4.3.5. Rechenzeit

Für eine realistische Nachbildung der Rechenzeiten wurden die Gesamtrechenzeiten der Trackingverfahren aus Tabelle 3.2 in Abschnitt 3.1.4. entnommen und mit den Beobachtungen aus dem DIP-Framework approximiert. Die Rechenzeiten der Trackingverfahren sind in Tabelle xx aufgeführt.

#### 4.3.6. Standardparametersatz der Fehlermodelle

Für die Fehlermodelle wurde der Standardparametersatz aus den Vergleichen der Trackingverfahren in Kapitel 3 abgeschätzt.

Das deskriptorlose KLT-Trackingverfahren nutzt keine Bildebenen zur exakten Positionsbestimmung, das im Gegensatz zu den deskriptorbasierten Trackingverfahren stärkere Rauschfehler verursacht, wie aus der Varianz in Tabelle 4.1 hervorgeht.

Für die Auftrittswahrscheinlichkeit der Driftfehler wurde für das KLT- und BRISK- Trackingverfahren in Tabelle 4.1 eine deutlich kleinere Varianz gewählt, da hierbei die variierbare Wahrscheinlichkeitsdichte deutlich ansteigt und nach der Gewichtung mit der idealisierten Wahrscheinlichkeitsdichte größere Werte angenommen werden, die durch die Bernoulli-Verteilung einen stärkeren Einfluss auf das Auftreten der Driftfehler ausüben.

Für den Kantendriftfehler beim KLT-Trackingverfahren werden im Gegensatz zum BRISK- Trackingverfahren kleinere Varianzen in Tabelle 4.1 verwendet, da das KLT-Trackingverfahren auf Kanten empfindlicher reagiert. Für die SIFT- und SURF-Trackingverfahren wurden die Kantendriftfehler ausgestellt.

Für die Mismatch-Fehler wurde die Auftrittswahrscheinlichkeit in Tabelle 4.1 prozentual festgelegt. Hierbei erreicht das SIFT-Trackingverfahren mit dieser Parametereinstellung weniger Mismatch-Fehler, als bei den übrigen deskriptiven Trackingverfahren. Durch die iterative ID-Generierung der Interessensmerkmale entstehen keine Mismatch-Fehler.

Der Vergleich der Deskriptor-Vektoren und die genaue Beschreibung der Interessensmerkmale in Differentialbildern der SIFT-Trackingverfahren erfordern hohe Rechenzeiten. Das SURF-Trackingverfahren erreicht geringere Rechenzeiten, die auf die Verwendung von Integralbildern zurückzuführen sind [Tabelle 4.1]. Das BRISK-Trackingverfahren erreicht unter den deskriptiven Trackingverfahren die geringste Rechenzeit, die auf dem einfachen Vergleich der Binär-Vektoren und der schnellen Detektion der Eckpunkte im Bild basiert [Tabelle 4.1].

<b>Fehlermodell</b>	<b>KLT</b>	<b>SIFT</b>	<b>SURF</b>	<b>BRISK</b>
Rauschfehler $\sigma$	2	4	4	3
Driftfehler $\sigma$	2	6	5	2
Kantendriftfehler $\sigma$	5	0	0	7
Mismatch-Fehler %	0	15	20	30
Rechenzeit $t$ [ms]	2.3	2054.5	262.2	146.0
ID-Generierung	Ja	Nein	Nein	Nein

Tabelle 4.1.: Festgelegter Parametersatz für die Fehlermodelle des Feature Tracker Simulators

## 4.4. Weitere Ergänzungen im Feature Tracker Simulator

Der Feature Tracker Simulator mit den erweiterten Fehlermodellen, stellt die Fehlermerkmale mit den aus den Versuchen und der Beobachtung aufgrund der im Dip-Framework festgelegten Standardparametern gut dar. Für die einfache Bedienung des Feature Trackers wurden für die Auswahl der Trackingverfahren und der Parameteränderung der Fehlermodelle Schieberegler mit der OpenCV Programmbibliothek im Visualisierungsfenster implementiert [Abbildung 4.14]. Mit einem ergänzenden individuellen Trackingverfahren können eigenständige Parameter für die Fehlerfunktionen festgelegt oder jeweilige Fehlerfunktionen ausgestellt werden. Die erweiterte Feature Tracker Simulation ermöglicht des Weiteren eine Unterscheidung der Interessensmerkmale und der Landmarken durch verschiedene Farbgebung. Auch werden ID-Zuweisungen der jeweiligen Interessensmerkmale und Landmarken in der erweiterten Feature Tracker Simulation dargestellt, um die Interessensmerkmale voneinander unterscheiden zu können. Ergänzend zu diesen Erweiterungen besteht die Möglichkeit beim Programmstart zwischen der idealen oder der realitätsnahen Simulation zu wechseln.

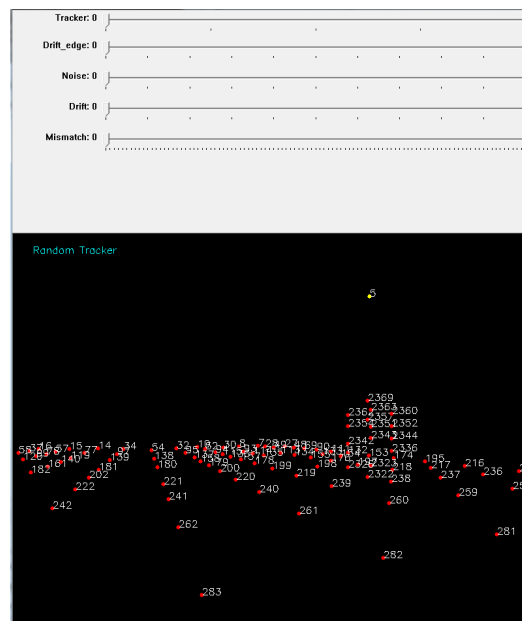


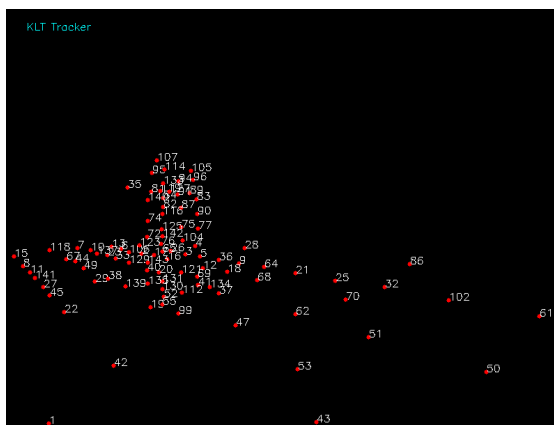
Abbildung 4.14.: Parametereinstellung durch Schieberegler im erweiterten Feature Tracker Simulator

# 5 Ergebnis

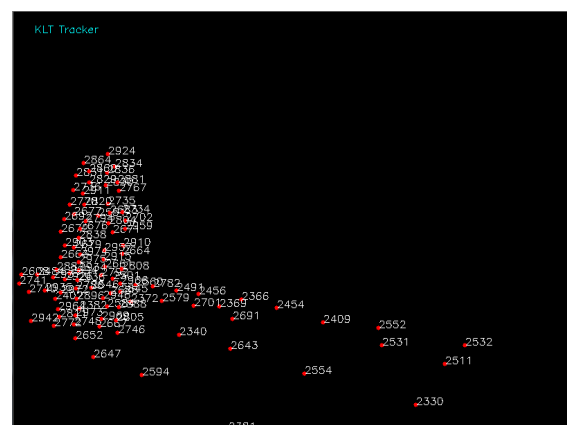
## 5.1. Validierung der implementierten Fehlermodelle

Für eine möglichst realistische Feature Tracker Simulation werden die implementierten Fehlermodelle mit den aus Abschnitt 4.3 eingestellten Parametern getestet und mit dem DIP-Framework verglichen. Das DIP-Framework dient hierbei als eine Art Referenzsystem, das in Abschnitt 3.4 näher beschrieben wurde. Für das DIP-Framework wurden zunächst passende Szenarien ausgesucht, die für einen realen Flugeinsatz in Betracht kommen. Hierfür wurden strukturreiche Graubild- und Farbbildaufnahmen verwendet. Mit den aus Tabelle 4.1 eingestellten Parametern für die Trackingverfahren des DIP-Frameworks wurden die Bildfolgen mit den jeweiligen Trackingverfahren analysiert. Das Verhalten der Interessensmerkmale in der Bildsequenz wurde beobachtet und mit den generierten Interessensmerkmalen der Simulation verglichen.

In der Simulation ist auffällig, dass das KLT-Trackingverfahren die Rechenzeiten, die iterativ erzeugten ID-Zuordnungen und die Driftfehler bei schnellen Bildbewegungen gut nachbildet. Die Abbildung 5.1a und Abbildung 5.1b zeigen die iterativ erzeugten Interessensmerkmal-IDs einer vorherigen und einer nachfolgenden Bildebene. Aus den beiden Bildebenen ist erkennbar, dass das KLT-Trackingverfahren verlorene Interessensmerkmale in nachfolgenden Bildebenen einer gleichen Bildszene nicht mehr wiederfindet und infolgedessen neue Interessensmerkmal-IDs zuordnet.



(a) Bildebene  $t_1$



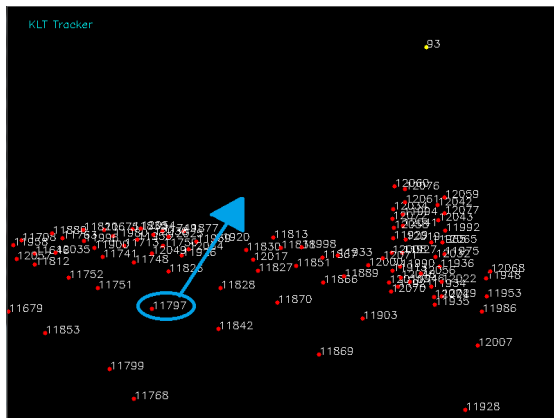


Die aus dem Referenzsystem übertragenen und gelb markierten Interessensmerkmale in Abbildung 5.2 werden nur im sichtbaren Bildbereich verfolgt, die nach Verlassen des sichtbaren Bildbereichs nicht mehr wiedergefunden werden.

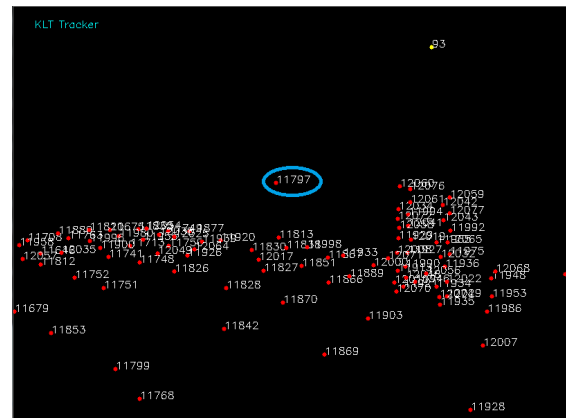


Abbildung 5.2.: Gefundene Interessensmerkmale des KLT-Trackingverfahrens im Referenzsystem

Die Abbildung 5.3a und Abbildung 5.3b zeigen im markierten Bereich die generierten Driftfehler der Interessensmerkmale in der Simulation.



(a) Bildebene  $t_1$



(b) Bildebene  $t_n$

Abbildung 5.3.: Driftfehler des KLT-Trackingverfahrens in der Simulation

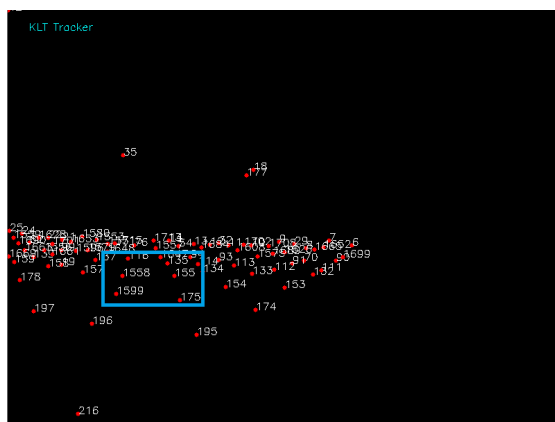
Die entstandenen Driftfehler in einer Bildfolge aus dem Referenzsystem sind in Abbildung 5.4 als gelbe Driftstrecken dargestellt.



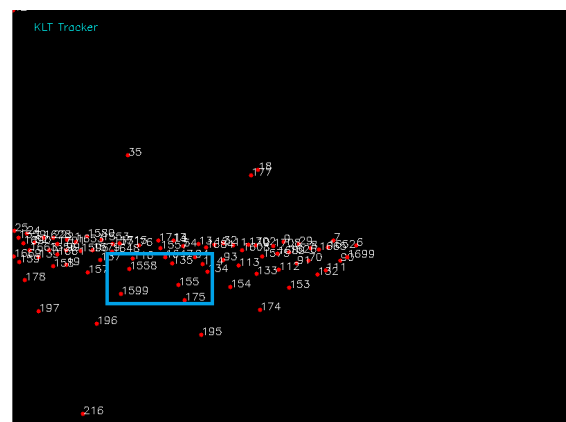
Abbildung 5.4.: Driftfehler des KLT-Trackingverfahrens im DIP-Framework

Die generierten Kantendriftingfehler sind ebenfalls gut wiederzuerkennen, jedoch bestehen hierbei geringe Diskrepanzen bei der Beständigkeit und der Positionsabweichung der Kantendriftingfehler zu dem Referenzsystem, da eine genaue Aussage über die Positionsabweichung der Kantendriftingfehler und der Beständigkeit nicht getroffen werden kann. Die Beständigkeit eines Kantendriftingfehlers eines Interessensmerkmals in der Simulation ist abhängig von der berechneten Eintrittswahrscheinlichkeit und der Dauer der Speicherung in der dynamischen Liste aus Abschnitt 4.2.4..

Das Interessensmerkmal mit der ID-Zuordnung stellt das Kantendrifting zwischen einer vorherigen Bildebene in Abbildung 5.5a und in einer nachfolgenden Bildebene in Abbildung 5.5b dar. Hierbei verschiebt sich das Interessensmerkmal auf der erkannten Kante im markierten Bereich. Zum Vergleich ist das Kantendrifting aus dem Referenzsystem in Abbildung 5.6 beispielhaft dargestellt.



(a) Bildebene  $t_1$



(b) Bildebene  $t_n$

Abbildung 5.5.: Implementierte Kantendriftingfehler des KLT-Trackingverfahrens in der Simulation

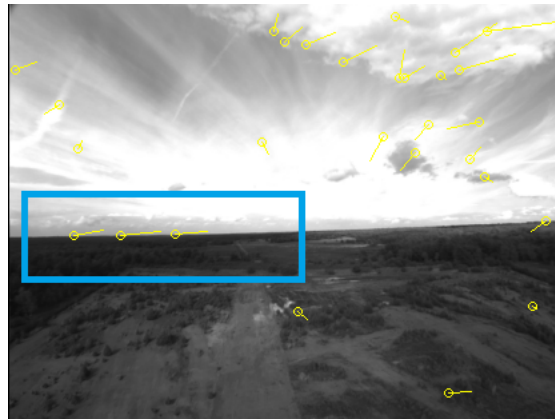


Abbildung 5.6.: Kantendriftfehler des KLT-Trackingverfahrens im DIP-Framework

Bei den deskriptiven Trackingverfahren der Simulation sind die Rechenzeiten deutlich zu erkennen. Diese konnten mit Hilfe der Untersuchungen aus Kapitel 3 und der Beobachtung des DIP-Frameworks gut nachempfunden werden. Auch die Mismatch-Bildung der deskriptiven Trackingverfahren erzielten mit den Parametereinstellungen aus Tabelle 4.1 realistische Ergebnisse. Das SIFT- und SURF-Trackingverfahren lieferten wenige Mismatch-Fehler. Das BRISK-Trackingverfahren hingegen erzielte, ähnlich wie beim DIP-Framework, deutlich höhere Mismatch-Fehler. Die nachfolgende Abbildung 5.7 und Abbildung 5.8 zeigen beispielhaft die Mismatch-Fehler des SURF- und BRISK-Trackingverfahren aus der Simulation. Hierbei konnten verlorene Interessensmerkmale in nachfolgenden Bildfolgen wiedergefunden werden, die durch gleiche ID-Zuordnung gekennzeichnet sind. Die markierten Bereiche in den Bildebenen zeigen hierbei einige generierte Mismatch-Fehler in der Simulation.

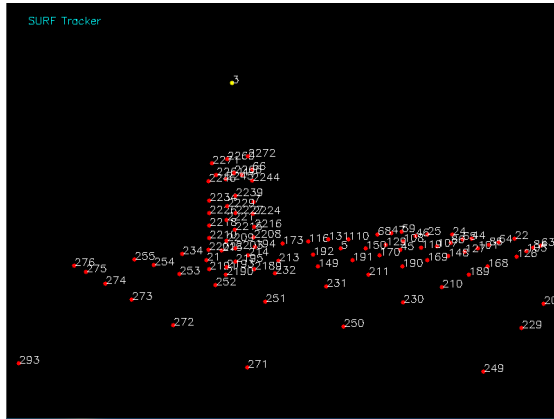
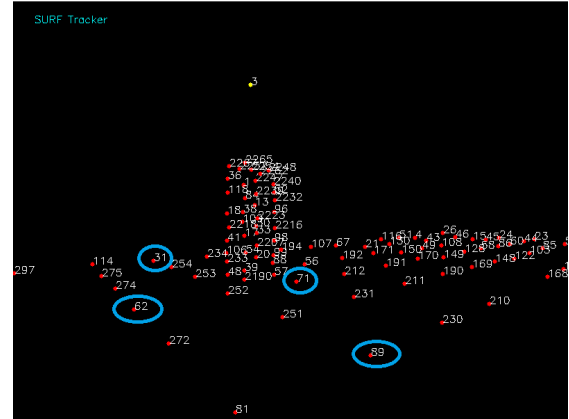
(a) Bildebene  $t_1$ (b) Bildebene  $t_n$ 

Abbildung 5.7.: Implementierte Mismatch-Fehler des SURF-Trackingverfahrens in der Simulation

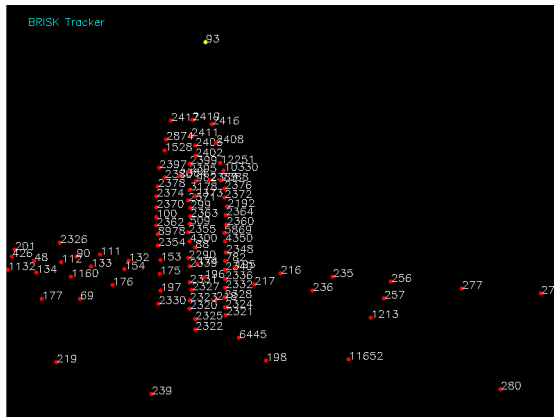
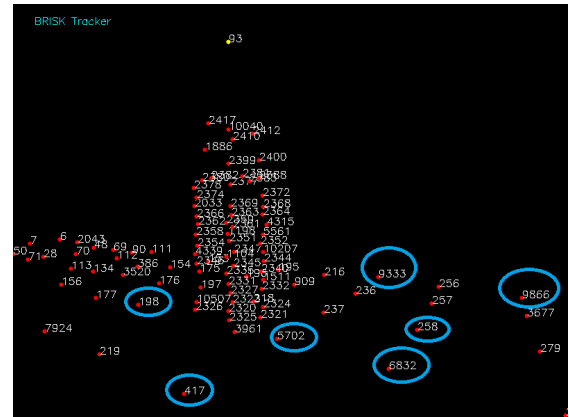
(a) Bildebene  $t_1$ (b) Bildebene  $t_n$ 

Abbildung 5.8.: Implementierte Mismatch-Fehler des BRISK-Trackingverfahrens in der Simulation

Vergleichend zu den simulierten Mismatch-Fehlern zeigen die Abbildung 5.9 und Abbildung 5.10 beispielhaft die Mismatch-Fehler des SIFT- und SURF-Trackingverfahrens aus dem Referenzsystem. Hierbei deuten gleiche Farbkreise auf eine übereinstimmende Korrespondenz und ungleiche Farbkreise auf Mismatch-Fehler, die beim BRISK-Trackingverfahren deutlich stärker ausfallen.

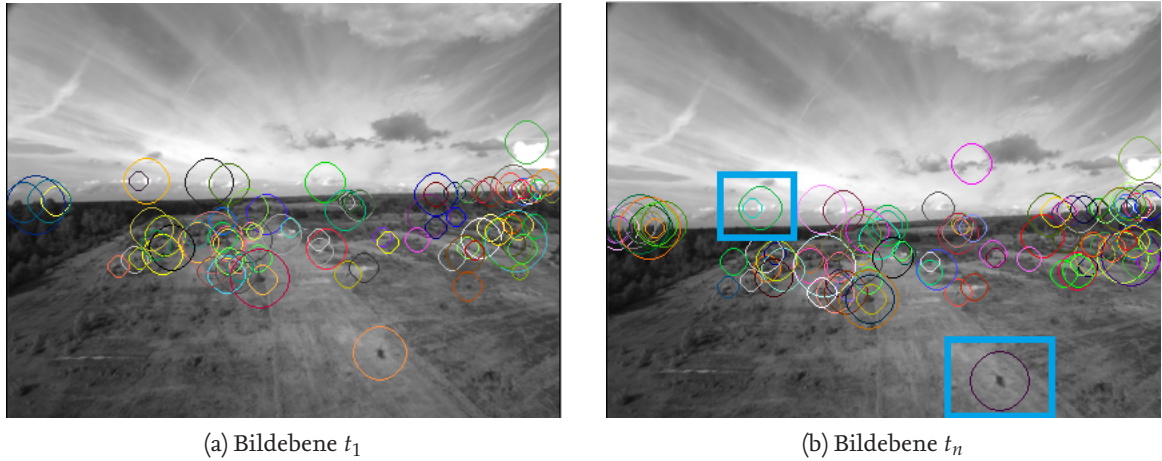


Abbildung 5.9.: Mismatch-Fehler des SURF-Trackingverfahrens im DIP-Framework

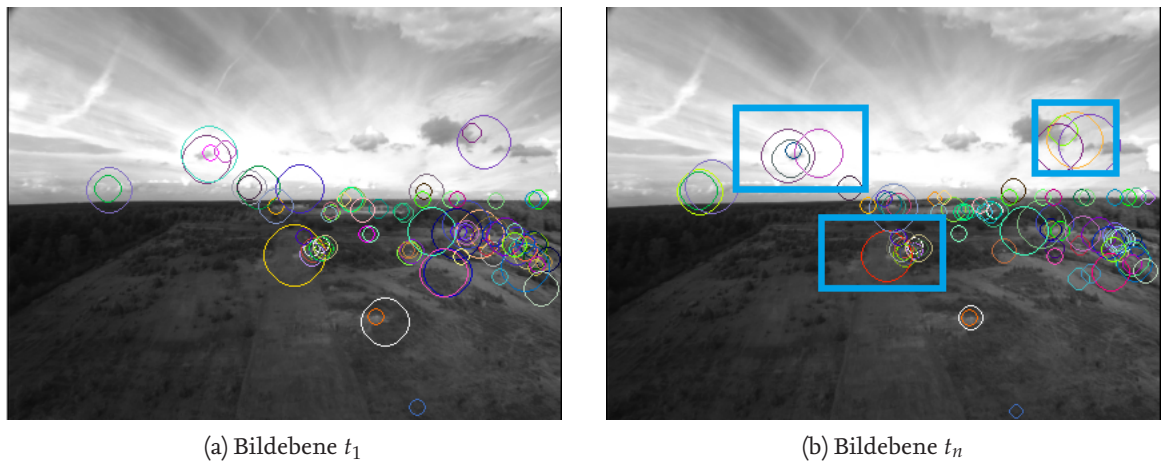


Abbildung 5.10.: Implementierte Mismatch-Fehler des BRISK-Trackingverfahrens im DIP-Framework

Die Rauschfehler konnten unter den Parametereinstellungen aus Tabelle 4.1 bei allen Trackingverfahren gut nachempfunden werden. Auffällig beim KLT-Trackingverfahren ist die hohe Rauschintensität, die auch beim DIP-Framework in strukturschwachen Bildregionen entsteht. Bei den deskriptiven Trackingverfahren, die durch subpixelgenaue Positionen robustere Interessensmerkmale erzielen, sind geringe Rauschintensitäten zu erkennen.

Die generierten und akkumulierenden Rauschfehler des KLT-Trackingverfahrens in der Simulation ist in Abbildung 5.11 deutlich zu erkennen. Vergleichend mit dem KLT-Trackingverfahren aus dem Referenzsystem finden sich Rauschfehler im markierten Bereich in Abbildung 5.12 wieder. Die weniger stark anfälligen Rauschfehler in den deskriptiven Trackingverfahren sind hierbei in der Abbildung 5.13 und zum Vergleich in Abbildung 5.14 mit dem Referenzsystem gegenübergestellt.

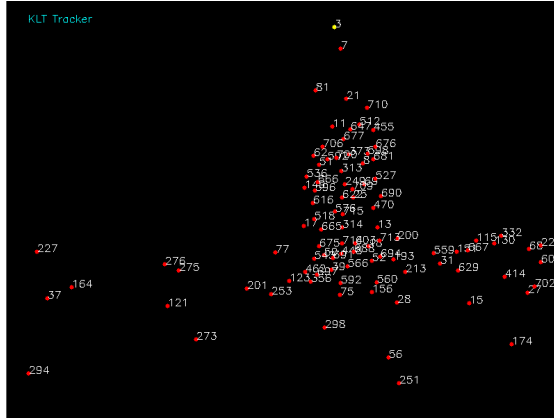
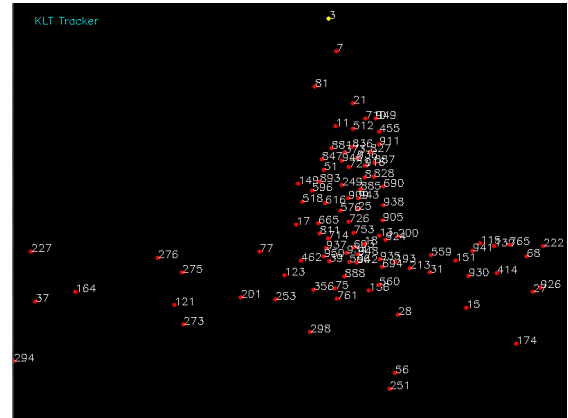
(a) Bildebene  $t_1$ (b) Bildebene  $t_n$ 

Abbildung 5.11.: Rauschfehler des KLT-Trackingverfahrens in der Simulation

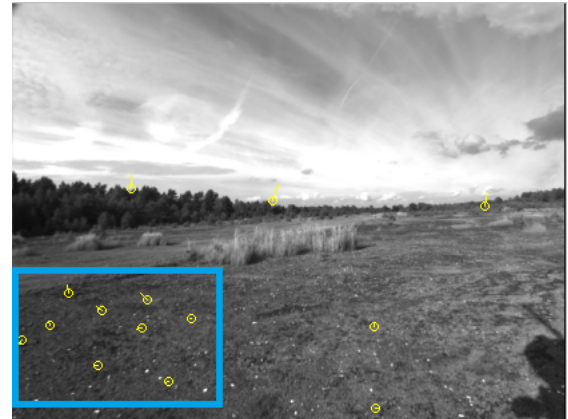
(a) Bildebene  $t_1$ (b) Bildebene  $t_n$ 

Abbildung 5.12.: Rauschfehler des KLT-Trackingverfahrens im DIP-Framework

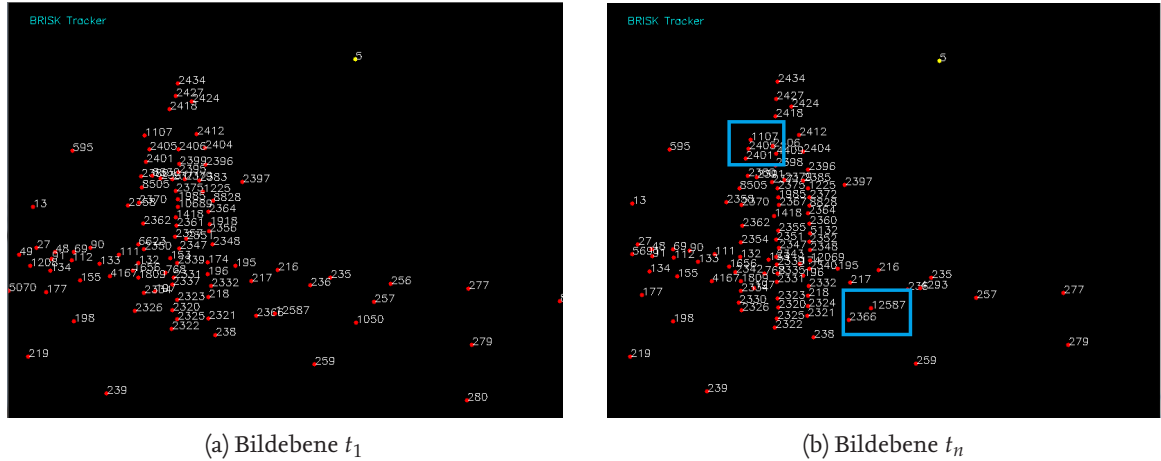


Abbildung 5.13.: Rauschfehler des BRISK-Trackingverfahrens in der Simulation

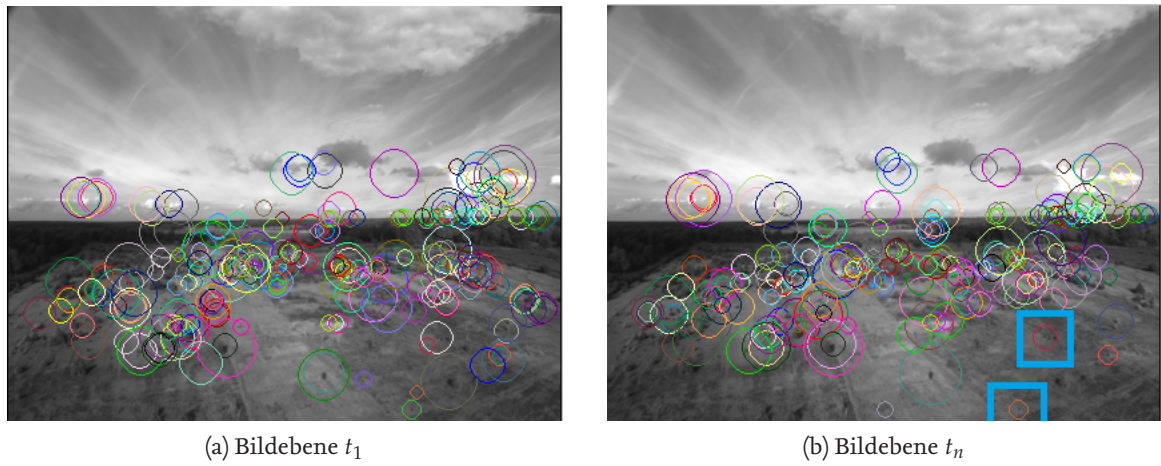


Abbildung 5.14.: Rauschfehler des BRISK-Trackingverfahrens im DIP-Framework



Zur Erkennung der Kantendriftfehler des Brisk-Trackingverfahren dienen die Abbildung 5.15 und Abbildung 5.16. Zum Vergleich stellt der markierte Bereich des BRISK-Trackingverfahrens des Referenzsystems den Kantendriftfehler dar.

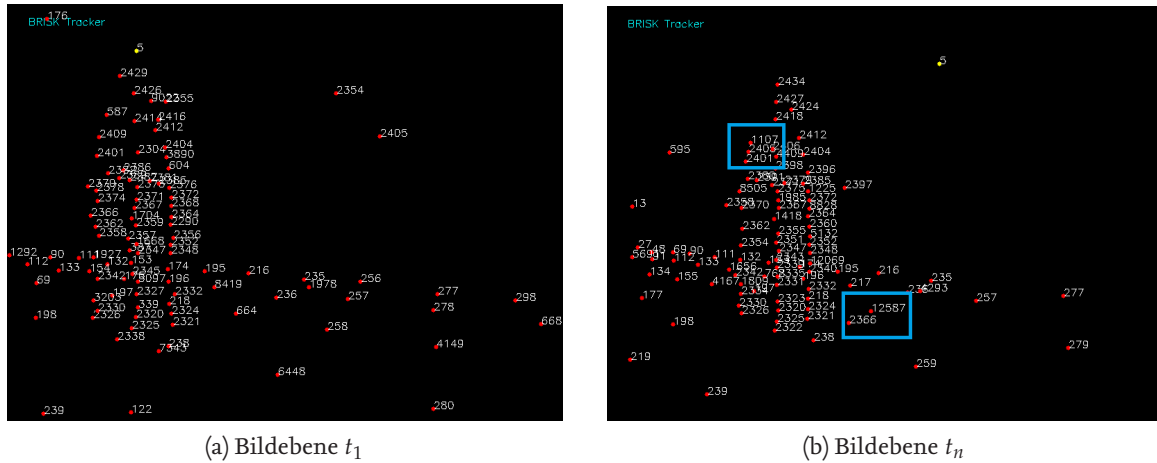


Abbildung 5.15.: Kantendriftfehler des BRISK-Trackingverfahrens in der Simulation

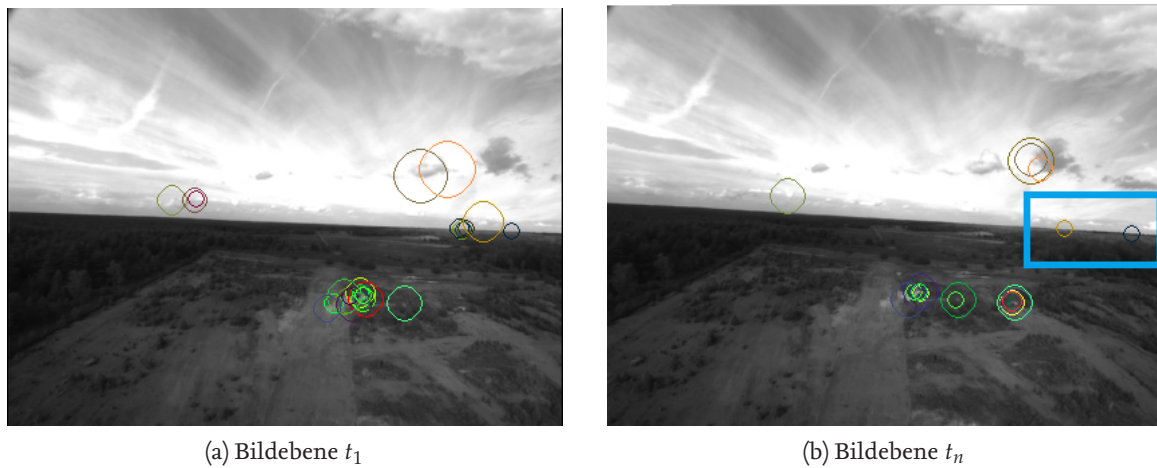


Abbildung 5.16.: Kantendriftfehler des BRISK-Trackingverfahrens im DIP-Framework



# 6 Zusammenfassung

In der vorliegenden Arbeit wurde der ideale Feature Tracker Simulator für den unbemannten Flug mit Fehlermodellen ausgestattet. Die Nachempfindung der Fehlermodelle soll eine realitätsnahe Simulation der natürlichen Feature Tracker für die Flugmissionen ermöglichen. Für das Verständnis wurden zunächst die markerbasierten und markerlosen Trackingverfahren vorgestellt. Insbesondere lag der Schwerpunkt in dieser Arbeit auf den markerlosen Trackingverfahren, die im Feature Tracker Simulator eingesetzt wurden. Hierbei wurden die Bestandteile Detektoren, Deskriptoren und Korrespondenzsuche des markerlosen Trackingverfahrens näher beschrieben. Für die Nachempfindung der Fehlermodelle im Feature Tracker Simulator wurden für die KLT-, SIFT-, SURF- und BRISK-Trackingverfahren zunächst der Aufbau anhand der mathematischen Funktionsweisen vorgestellt. Mit einem anschließenden Vergleich der Bestandteile dieser Trackingverfahren bei verschiedenen Umgebungsbedingungen, die aus externen Versuchen entnommen wurden, konnten die Vor- und Nachteile beschrieben werden. Aus den Vergleichen und der darauffolgenden Untersuchung der mathematischen Funktionen der Trackingverfahren wurden die Fehlerursachen der Fehlermerkmale untersucht. Ergänzend zum Vergleich der Trackingverfahren wurden reale Feature Tracker beobachtet und Feature Tracker des DIP-Frameworks verwendet, um Auftrittswahrscheinlichkeiten und Ausmaße der Fehlermerkmale in einem Parametersatz der jeweiligen Trackingverfahren festzulegen. Die verschiedenen Arten der Fehlermerkmale wurden anhand der entstandenen Fehlerursachen im Trackingverfahren durch Fehlerfunktionen im Feature Tracker Simulator nachempfunden.

Die generierten Fehlermerkmale aus der entsprechenden Fehlerfunktion wurden anschließend nach ihrer Auftrittswahrscheinlichkeit und der Fehlerausmaße aus den Beobachtungen festgelegt. Für die Validierung der extrahierten Fehlermodelle in der Feature Tracker Simulation wurden die generierten Fehlermerkmale mit den Fehlermerkmalen aus dem realen Feature Tracker des DIP-Frameworks verglichen.

# 7 Ausblick

Für die Selbstlokalisierung der Luftfahrzeuge anhand der visuellen Odometrie oder der simultanen Lokalisierungs- und Kartenerstellungsverfahren (SLAM) werden extrahierte Interessensmerkmale aus Feature Trackern verwendet. Hierbei können abweichende Luftfahrzeugpositionen durch Trackingfehler entstehen, die mit dem erweiterten Feature Tracker Simulator genauer analysiert werden können. Mit der Evaluation der Trackingverfahren im Feature Tracker Simulator kann beispielsweise das Fehlerverhalten auf wechselnde Umgebungsbedingungen untersucht werden, aus denen potenzielle Trackingverfahren für die Selbstlokalisierung bestimmt werden können. Potenzielle Trackingverfahren aus der Untersuchung müssen für die Selbstlokalisierung der Luftfahrzeuge den Anforderungen der Luftfahrtrichtlinien entsprechen, um sichere Navigationsverfahren verifizieren zu können.

Die erweiterte Feature Tracking Simulation bietet einen Grundstein für weitere Arbeiten mit zusätzlichen Trackingverfahren, die in dieser Arbeit nicht vorgestellt wurden. Weitere nicht vorhandene Fehlermodelle, die sich aus zusätzlichen Trackingverfahren ergeben, können in der erweiterten Feature Tracking Simulation ergänzt werden. Bei gleichen Fehlermerkmalen, die bei den zusätzlichen Trackingverfahren auftauchen, können die optimalen Parameter der zusätzlichen Trackingverfahren in der Parameterklasse des Feature Trackers angepasst werden.

Die Versuche aus anderen Arbeiten und die genaue Untersuchung der Trackingalgorithmen, die zur Erkennung der Fehlermerkmale beigetragen haben, konnten über das Auftreten und Ausmaß der Fehlermerkmale keine genauen Ergebnisse liefern. Aus Beobachtungen von Bildaufnahmen der realen Feature Tracker wurden hierzu die Parameter der Fehlerfunktionen festgelegt.

In weiteren Arbeiten können durch gezielte statistische Auswertungen das Fehlerauftreten und Fehlerausmaß der Trackingverfahren unter Verwendung von Bildaufnahmen aus Flugversuchen genauer untersucht werden, um den vorhandenen Parametersatz der Trackingverfahren zu optimieren.

Für den erweiterten Feature Tracker Simulator steht des Weiteren nur das vereinfachte Lochkameramodell zur Verfügung. In vorhergehenden Arbeiten wurden bereits für den idealen Feature Tracker Simulator verschiedene Kameramodelle implementiert. Diese können in nachfolgenden Arbeiten zur erweiterten Feature Tracker Simulation ergänzt werden.

# Literaturverzeichnis

- [1] Marcus Tönnis. *Augmented Reality: Einblicke in die Erweiterte Realität*, volume 0 of *Informatik im Fokus*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [2] Nrc-iit publications iti-cnrc | artag revision 1. a fiducial marker system using digital techniques. pages 1–46, 2004.
- [3] Wilhelm Burger and Mark James Burge. *Digitale Bildverarbeitung: Eine algorithmische Einführung mit Java*. X.media.press. Springer Vieweg, Berlin and Heidelberg, 3., vollst. überarb. und erw. aufl. edition, 2015.
- [4] Bernd Jähne. *Digitale Bildverarbeitung und Bildgewinnung*. Springer, Berlin and Heidelberg, 7., neu bearbeitete aufl. 2012 edition, 2012.
- [5] [http://www.mathematik.uni-ulm.de/stochastik/lehre/ws05\\_06/seminar/ausarbeitung\\_wagner.pdf](http://www.mathematik.uni-ulm.de/stochastik/lehre/ws05_06/seminar/ausarbeitung_wagner.pdf).
- [6] Christoph Wagner. *Kantenextraktion - Klassische Verfahren*. PhD thesis, Universität Ulm, Ulm, 2006.
- [7] Alfred Nischwitz, Max Fischer, Peter Haberäcker, and Gudrun Socher. *Computergrafik und Bildverarbeitung: Band II: Bildverarbeitung*. Studium. Vieweg+Teubner Verlag, Wiesbaden, 3., neu bearb. aufl. edition, 2011.
- [8] Oliver Pink. *Bildbasierte Selbstlokalisierung von Straßenfahrzeugen: Zugl.: Karlsruhe, KIT, Diss.*, 2010, volume 17 of *Schriftenreihe / Institut für Mess- und Regelungstechnik, Karlsruher Institut für Technologie*. KIT Scientific Publishing, Karlsruhe, print on demand edition, 2011.
- [9] Tolga Birdal. <https://dsp.stackexchange.com/questions/1714/best-way-of-segmenting-veins-in-leaves>.
- [10] <https://de.mathworks.com/matlabcentral/fileexchange/25865-euclidean-skeleton?focused=5139452&tab=function>.
- [11] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL. 27, NO. 10, pages 1–16, 2005.
- [12] Krystian Mikolajczyk and Cordelia Schmid. Performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630, 2005.
- [13] Jacob Toft Pedersen. Surf: Feature detection & description. pages 1–12, 2011.
- [14] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

- [15] Robert Hofmann. *Implementierung des SURF-Feature-Detektors auf der GPU mit Hilfe von CUDA*. PhD thesis, Universität Koblenz Landau, Koblenz, 2009.
- [16] Stefan Leutenegger, Margarita Chli and Roland Y. Siegwart. *BRISK: Binary Robust Invariant Scalable Keypoints*. IEEE, Piscataway, NJ, 2011.
- [17] Markus Volkmer. *Erkennung mehrerer Objekte durch parallele Suchagenten*. Informatik. Diplomarbeiten Agentur, Hamburg, 2013?
- [18] Jianbo Shi and Carlo Tomasi. Good features to track. *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, 1994.
- [19] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. Springer London, Guildford, Surrey, 1. aufl. edition, 2010.
- [20] Adrian Batzill. *Interaktive Lokalisierung durch Objekterkennung*. PhD thesis, Albert-Ludwigs-Universität Freiburg im Breisgau, Freiburg im Breisgau, 2013.
- [21] eric yuan. <http://eric-yuan.me/surf/>.
- [22] eisert. <https://www.informatik.hu-berlin.de/de/forschung/gebiete/viscom/teaching/media/cphoto10/cph>
- [23] Markus Frey. *Schnelles Kameratracking mit Autokalibrierung auf planaren Oberflächen*. PhD thesis, Albert-Ludwigs-Universität Freiburg im Breisgau, Freiburg, 2013.
- [24] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine*, 19(2):78–90, 2012.
- [25] Gang Hua and Hervé Jégou. *Computer Vision – ECCV 2016 Workshops*, volume 9913. Springer International Publishing, Cham, 2016.
- [26] Steffen Gauglitz, Tobias Höllerer, and Matthew Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *International Journal of Computer Vision*, 94(3):335–360, 2011.
- [27] Thomas M. Deserno, Heinz Handels, Hans-Peter Meinzer, and Thomas Tolxdorff, editors. *Bildverarbeitung für die Medizin 2014: Algorithmen - Systeme - Anwendungen ; Proceedings des Workshops vom 16. bis 18. März 2014 in Aachen*. Informatik aktuell. Springer, Berlin, 2014.
- [28] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging Understanding Workshop*, pages 121–130, 1981.
- [29] Michael Hiebl. *Realtime Optical Flow*. Seminararbeit, Hochschule München, München, 2011.
- [30] Xiaowei cqu. [http://prog3.com/sbdrm/blog/xiaowei\\_cqu/article/details/8067881](http://prog3.com/sbdrm/blog/xiaowei_cqu/article/details/8067881).
- [31] Florian Mücke. *Analyse GPU-basierter Feature Tracking Methoden für den Einsatz in der Augmented Reality*. Diplomarbeit, Universität Augsburg, Augsburg, 2007.
- [32] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

- [33] Shastry, Anand, Schowengerdt, and Robert. Airborne video registration for visualization and parameter estimation of traffic flows. *Pecora 15/Land Satellite Information IV/ISPRS Commission I/FIEOS 2002 Conference Proceedings*, pages 1–10, 2002.

# Abbildungsverzeichnis

2.1.	Baummarker mit retroreflektiver Oberfläche . . . . .	10
2.2.	Flachmarker mit zweidimensionalem Muster [1] . . . . .	11
2.3.	Binärbilder als Flachmarker der markerbasierten Trackingverfahren . . . . .	12
2.4.	Bitfelder und Farbmuster als Flachmarker der markerbasierten Trackingverfahren . . . . .	12
2.5.	Steigung einer Bildfunktion [3] . . . . .	14
2.6.	Detektierte Kante eines Graubildes mit erster partieller Ableitung [5] . . . . .	15
2.7.	Detektierte Kante eines Graubildes mit zweiter partieller Ableitung [6] . . . . .	15
2.8.	Eckendetektion durch Vergleich der Eigenwerte . . . . .	16
2.9.	Blob als Interessensmerkmal . . . . .	17
2.10.	Detektion von Bildlinien eines Flugzeuges [10] . . . . .	18
2.11.	Gauß-Pyramide einer jeweiligen Multiskalen-Oktave . . . . .	19
2.12.	Erstellen des Multiskalenraums [7] . . . . .	20
2.13.	Bestimmen der Gradientenrichtung [7] . . . . .	21
2.14.	Berechnung eines Integralbildes [13] . . . . .	21
2.15.	Berechnung eines Integralbildes . . . . .	22
2.16.	Haar-Wavlet-Filter [15] . . . . .	22
2.17.	Verschiebung im eindimensionalen Fall [3] . . . . .	25
2.18.	Mögliche Transformationen des KLT-Algorithmus [19] . . . . .	26
2.19.	Verschieben des Referenzbildes im Suchbild durch Koordinatentransformation [3] . . . . .	27
2.20.	Vergleich der Nachbarwerte in Differenzbildern [7] . . . . .	29
2.21.	Aus Orientierungshistogrammen entstandene Orientierungsfenster [7] . . . . .	31
2.22.	Erweiterte Diskretisierung der LoG-Filter durch Mittelwertfilter [14] . . . . .	32
2.23.	Orientierung der benachbarten Interessensmerkmale [20] . . . . .	33
2.24.	Vorgehensweise zur Charakterisierung der Interessensmerkmale [21] . . . . .	33
2.25.	Wavlet-Antworten im Subfenster [22] . . . . .	34
2.26.	Vergleich der benachbarten Grauwerte im Suchkreis [23] . . . . .	35
2.27.	Bestimmen der subpixelgenauen Position im Multiskalenraum [16] . . . . .	36
2.28.	Kreismaske um einen detektierten Eckpunkt [16] . . . . .	37
3.1.	Vergleich der Detektoren [24] . . . . .	39
3.2.	Neue ID-Zuweisung des Interessensmerkmals bei erneutem Eintreten in die Bildebene . . . . .	42
3.3.	Aperture-Probleme bei schlecht detektierten Interessensmerkmalen im Bild [19] . . . . .	43
3.4.	Vorherige ID-Zuweisung des Interessensmerkmals bei erneutem Eintreten in die Bildebene . . . . .	44

3.5. Stetige Bildfunktion der jeweiligen Skalenebene als approximierte Laplace-Funktion [30] . . . . .	45
3.6. Abweichung der subpixelgenauen Position durch die Diskretisierung . . . . .	45
3.7. Präzision der Korrespondenzsuche bei variierender Fenstergröße [26] . . . . .	47
3.8. GUI-Oberfläche des DIP-Frameworks . . . . .	49
4.1. Programmoberfläche des Feature Tracker Simulators . . . . .	50
4.2. Luftfahrzeug im Feature Tracker Simulator . . . . .	50
4.3. Visualisierungsfenster des Lochkameramodells mit ideal identifizierten Interessensmerkmalen . . . . .	51
4.4. Mathematisches Prinzip der Lochkamera [4] . . . . .	52
4.5. Funktion zur inkrementellen ID-Generierung . . . . .	54
4.6. Fehlerfunktion der Rauschfehler . . . . .	56
4.7. Fehlerfunktion der Driftfehler . . . . .	58
4.8. Hough-Transformation der 4 Bildgeraden [3] . . . . .	59
4.9. Dynamische Liste für die Interessensmerkmale mit Kantendriftfehlern . . . . .	60
4.10. Fehlerfunktion der Kantendriftfehler . . . . .	61
4.11. Fehlerfunktion zur Mismatch-Bildung . . . . .	63
4.12. Rauscherzeugung mit der bivariaten Normalverteilung . . . . .	64
4.13. Erzeugen der Driftfehler mit der eindimensionalen Normalverteilung . . . . .	66
4.14. Parametereinstellung durch Schieberegler im erweiterten Feature Tracker Simulator . . . . .	69
5.1. Implementierte ID-Generierung des KLT-Trackingverfahrens in der Simulation . . . . .	70
5.2. Gefundene Interessensmerkmale des KLT-Trackingverfahrens im Referenzsystem . . . . .	71
5.3. Driftfehler des KLT-Trackingverfahrens in der Simulation . . . . .	71
5.4. Driftfehler des KLT-Trackingverfahrens im DIP-Framework . . . . .	72
5.5. Implementierte Kantendriftfehler des KLT-Trackingverfahrens in der Simulation . . . . .	72
5.6. Kantendriftfehler des KLT-Trackingverfahrens im DIP-Framework . . . . .	73
5.7. Implementierte Mismatch-Fehler des SURF-Trackingverfahrens in der Simulation . . . . .	74
5.8. Implementierte Mismatch-Fehler des BRISK-Trackingverfahrens in der Simulation . . . . .	74
5.9. Mismatch-Fehler des SURF-Trackingverfahrens im DIP-Framework . . . . .	75
5.10. Implementierte Mismatch-Fehler des BRISK-Trackingverfahrens im DIP-Framework . . . . .	75
5.11. Rauschfehler des KLT-Trackingverfahrens in der Simulation . . . . .	76
5.12. Rauschfehler des KLT-Trackingverfahrens im DIP-Framework . . . . .	76
5.13. Rauschfehler des BRISK-Trackingverfahrens in der Simulation . . . . .	77
5.14. Rauschfehler des BRISK-Trackingverfahrens im DIP-Framework . . . . .	77
5.15. Kantendriftfehler des BRISK-Trackingverfahrens in der Simulation . . . . .	78
5.16. Kantendriftfehler des BRISK-Trackingverfahrens im DIP-Framework . . . . .	78

# Tabellenverzeichnis

3.1. Vergleich der evaluierten Deskriptoren aus [25] . . . . .	40
3.2. Vergleich der evaluierten Deskriptoren nach der Rechenzeit aus [27] . . . . .	41
3.3. Auflistung der Fehlermerkmale der jeweiligen Trackingverfahren . . . . .	48
4.1. Festgelegter Parametersatz für die Fehlermodelle des Feature Tracker Simulators . .	68



# A Anhang

## A.1. Ausführliche Matrizendarstellung des Lochkameramodells

$$\underline{\underline{T}} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Translation um } [t_x, t_y, t_z]^T$$

$$\underline{\underline{R}}_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Rotation um die x-Achse}$$

$$\underline{\underline{R}}_y = \begin{bmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Rotation um die y-Achse}$$

$$\underline{\underline{R}}_z = \begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Rotation um die z-Achse}$$

$$\underline{\underline{S}} = \begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & s_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Skalierung}$$

$$\underline{\underline{P}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{z} & 1 \end{bmatrix} \quad \text{perspektivische Projektion}$$



Julian Bayer

Matrikelnummer: 4504700

Technische Universität Braunschweig

Studiengang: Maschinenbau mit Vertiefungsrichtung Luft- und Raumfahrttechnik