

# Automatische Konfiguration eines Mehrkamerasystems zur Verfolgung von Objekten in unstrukturierten Umgebungen

Diplomarbeit von

Ann-Kristin Grosselfinger

An der Fakultät für Informatik  
Institut für Anthropomatik

Betreuer:

Prof. Dr.-Ing. Rainer Stiefelhagen<sup>1</sup>

Dr. rer. nat. Wolfgang Hübner<sup>2</sup>

David Münch<sup>2</sup>

Dauer: 01. Juni 2012 - 30. November 2012

<sup>1</sup>Karlsruher Institut für Technologie KIT - Fakultät für Informatik, Institut für Anthropomatik

<sup>2</sup>Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung, Gutleuthausstr. 1, 76275 Ettlingen



---

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig angefertigt und keine anderen, als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ettlingen, 30. November 2012

---

Ann-Kristin Grosselfinger

## **Zusammenfassung**

Bei der inhaltsbasierten Videoanalyse werden die aktuell vorkommenden Situationen in einer Szene automatisch erkannt und protokolliert. Dies ermöglicht einem Beobachter entsprechend auf die erkannten (Gefahren-)Situationen zu reagieren. Ein Teil eines automatischen echtzeitfähigen Systems zur Videoanalyse ist die Maximierung der Information über bestimmte lokale Ausschnitte einer Szene. Dazu können aktive Kameras verwendet werden. In dieser Arbeit wird zu einem Master-Slave Kamerasystem ein Konfigurationsverfahren vorgestellt, welches das Zusammenspiel einer Übersichtskamera mit großem Sichtfeld und einer aktiven Kamera über eine Motorkarte ermöglicht. Diese ordnet den Bildkoordinaten der Übersichtskamera Motorkoordinaten der aktiven Kamera zu. Unter Verwendung von SURF Deskriptoren zu lokalen Bildmerkmalen in den Bildern beider Kameras wird eine Korrespondenz zwischen den Kamerabildern ermittelt. Anhand dieser Korrespondenz wird eine Homographie bestimmt und mit dieser der Mittelpunkt des Bildes der aktiven Kamera transformiert. Durch spiralförmiges Bewegen der aktiven Kamera werden auf diese Weise mehrere Zuordnungen zwischen aktuellen Motorkoordinaten und den durch die Transformation erlangten Bildkoordinaten ermittelt. Aus diesen Zuordnungen werden unter Verwendung von linearer Regression mit RANSAC Interpolationsfunktionen bestimmt und mit diesen eine flächendeckende Motorkarte für alle Pixel im Übersichtsbild erstellt. Mit diesem Verfahren wurden Konfigurationen in verschiedenen Diskursbereichen durchgeführt, und zwar wurde das System in urbanem Gebiet in Bodennähe, auf ca. zehn Meter Höhe und in ca. 25 Meter Höhe aufgestellt. Des Weiteren wurden das Verfahren in Innenräumen und im Eingangsaußenbereich, auch bei Dämmerung und Dunkelheit, erfolgreich durchgeführt. Die Ergebnisse dieser Konfigurationsdurchläufe wurden anschließend qualitativ und in großen Teilen soweit möglich auch quantitativ evaluiert. Das in dieser Arbeit vorgestellte Verfahren funktioniert in allen Diskursbereichen mit durchschnittlichen Fehlern kleiner ein Grad. Allerdings gibt es, wie bereits im Vorfeld erwartet, aufgrund der verwendeten lokalen Bildmerkmale bei Szenen mit wenig Textur, wie z.B. einem strukturarmen, homogenen Flur, nicht ausreichen viele Korrespondenzen zwischen den beiden Kameras. Das vorgestellte Verfahren ist insbesondere für Diskursbereiche im Außenbereich gut geeignet und kann somit in das am Fraunhofer IOSB existierende System zur Videoanalyse eingebaut werden.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Problemstellung und Ziel der Arbeit . . . . .	2
1.2	Aufbau der Arbeit . . . . .	2
<b>2</b>	<b>Verwandte Arbeiten</b>	<b>3</b>
2.1	Hardware . . . . .	3
2.2	Motivation der Steuerung . . . . .	5
2.3	Organisation der Steuerung . . . . .	6
2.4	Konfiguration und Kalibrierung . . . . .	7
<b>3</b>	<b>Grundlagen</b>	<b>11</b>
3.1	SURF Merkmale . . . . .	11
3.1.1	Detektion markanter Regionen . . . . .	11
3.1.2	Bilden des Deskriptors . . . . .	15
3.2	NNDR: Neares Neighbor Distance Ratio . . . . .	17
3.3	RANSAC: Random Sample Consensus . . . . .	18
3.4	Schätzung der Projektivität. . . . .	18
3.5	Lineare Regression . . . . .	20
3.6	Hintergrund-Vordergrund Segmentierung . . . . .	21
3.7	Blobdetektor . . . . .	23
<b>4</b>	<b>Design</b>	<b>25</b>
4.1	Hardware . . . . .	25
4.2	Software . . . . .	26
<b>5</b>	<b>Umsetzung</b>	<b>33</b>
5.1	Spiralförmige Bewegung der Kamera . . . . .	34
5.2	SURF-Merkmale, deren Zuordnung und Ermitteln der Projektivität . . . . .	35
5.3	Interpolation der Motor- und Bildkoordinaten . . . . .	39
5.4	Erstellen einer flächendeckenden Motorkarte . . . . .	43
<b>6</b>	<b>Experimente und Auswertung</b>	<b>45</b>
6.1	Experiment A: Am Boden stehendes Stativ mit Blick nach Süden . . . . .	47

6.2 Experiment B:	Stativ im zweiten Stockwerk mit Blick nach Süden . . . .	52
6.3 Experiment C:	Stativ auf dem Turm mit Blick nach Süden . . . . .	55
6.4 Experiment D:	Stativ am Boden mit Blick nach Osten . . . . .	57
6.5 Experiment E:	Außenkameras des Kameranetzwerks am Eingang . . . .	62
6.6 Experiment F:	Stativ im Flur des Gebäudes . . . . .	64
<b>7 Diskussion und Ausblick</b>		<b>67</b>
<b>A Zeitplan</b>		<b>69</b>
<b>B Datenblätter</b>		<b>71</b>
<b>C Daten</b>		<b>75</b>
<b>Abbildungsverzeichnis</b>		<b>87</b>
<b>Literaturverzeichnis</b>		<b>96</b>

# Kapitel 1

## Einleitung

Bei der inhaltsbasierten Videoanalyse ist von prinzipiellem Interesse was, wann, wie, warum und wo in der beobachteten Szene passiert, wer genau daran beteiligt ist und was aller Voraussicht nach als nächstes geschehen wird. Für einige Ableitungen über die aktuelle Szene reichen grobe Übersichtsbilder, für viele Ableitungen ist jedoch mehr Detailinformation notwendig und eine minimale Auflösung des Bildausschnitts wird benötigt. Dabei kann es sich beispielsweise um das Erkennen von Fahrzeugkennzeichen handeln, um die Identifikation von Personen durch Gesichtserkennung, oder einfach die Verifikation von verfolgten Objekten. Die hierzu notwendige minimale Auflösung kann mit einer Übersichtskamera oft nicht erreicht werden, wohingegen aktive Pan-Tilt-Zoom-Kameras diese hochaufgelösten Bildausschnitte liefern können. Neben der Auflösung können auch andere Faktoren einen Grund für den Einsatz von unterschiedlichen oder aktiven Kameras liefern: Beispielsweise die Lichtverhältnisse, die dafür sorgen können, dass eine Kamera in dem Augenblick gegen das Licht gerichtet und somit 'geblendet' ist, oder zu wenig Licht, was den Einsatz einer Kamera im Infrarotbereich sinnvoll machen könnte. Auch Verdeckungen in der Szene, die in einer Kamera mit anderer Blickrichtung nicht vorhanden sind, oder Objekte, die aus verschiedenen Richtungen unterschiedlich gut zu identifizieren sind, sprechen für den Einsatz mehrerer Kameras.

Eine aktive Kamera könnte so gesteuert werden, dass sie ständig interessante Entwicklungen im Blickfeld behält. Als interessant einzustufen sind hierbei alle Entwicklungen, welche die Mehrdeutigkeit der Situationsinterpretation verringern und somit die Sicherheit der Interpretation erhöhen. Dies kann beispielsweise dadurch geschehen, dass neu im Blickfeld erscheinende Objekte zur Durchführung einer Identifikation fokussiert werden und ansonsten der Fokus auf gerade stattfindende oder zukünftig erwartete Situationen gelenkt wird. Dies kann sowohl Protokollierungs- und Überwachungszwecken dienen als auch der Verbesserung der sensorischen Erfassung selbst. Auf das Zusammenspiel von statischen und aktiven Kameras aufbauend können weitere inhaltsbezogene Videoauswerteverfahren angewandt werden, wie zum Beispiel auch die begriffliche

Situationserkennung. Die Situationserkennung kann nun eigenständig Information von den aktiven Kameras anfordern, um mögliche Unsicherheit auf hoher semantischer Ebene zu minimieren und das Wissen über die aktuell beobachtete Szene zu maximieren.

## 1.1 Problemstellung und Ziel der Arbeit

Um ein korrektes Zusammenspiel von Übersichtskameras und aktiven Kameras zu ermöglichen ist eine initiale robuste Konfiguration des Systems notwendig, damit in der Übersichtskamera entdeckte interessante Bereiche mit einer aktiven Kamera in höherer Auflösung angefahren werden können. Ziel der Arbeit ist das Erarbeiten einer sinnvollen Konfigurationsmethode und die Umsetzung dieser Konfiguration für ein gegebenes Mehrkamerasystem, das aus einer Übersichtskamera und einer aktiven Pan-Tilt-Zoom-Kamera besteht. Durch die Konfiguration soll das System in die Lage versetzt werden, gezielt bestimmte Bereiche in der Szene, anhand des Übersichtsbildes, mit der aktiven Kamera anzufahren. Dadurch soll die Verfolgung von Objekten ermöglicht werden. Es wird exemplarisch der Prototyp eines Mehrkamerasystems realisiert, der die Funktionalität von der Konfiguration des Systems bis hin zu einem einfachen Objektverfolgungsverfahren bereitstellt.

## 1.2 Aufbau der Arbeit

In Kapitel 2 werden vorhandene visuelle Systeme vorgestellt. verschiedene Ansätze bezüglich Hardwareaufbau 2.1, Motivation 2.2 und Steuerung 2.3 eingeordnet und verglichen. Der letzte Abschnitt des Kapitels, 2.4, geht auf die zur Kalibrierung und Konfiguration des Systems verwendeten Methoden diverser Arbeitsgruppen ein. Kapitel 3 erläutert einige Grundlagen zu Verfahren, die, in der in dieser Arbeit vorgestellten Konfigurationsmethode, angewandt werden. Kapitel 4 beschäftigt sich mit der Hardware und Software des für diese Arbeit verwendeten Systems. Kapitel 5 beschreibt die entwickelte Konfigurationsmethode. Kapitel 6 beinhaltet die durchgeführten Experimente mit der Konfiguration des Systems in verschiedenen Diskursbereichen. Zum Abschluss der Arbeit rekapituliert Kapitel 7 das Erreichte und liefert einen Ausblick auf mögliche Weiterentwicklungen.

# Kapitel 2

## Verwandte Arbeiten

Es gibt verschiedene Aspekte nach denen man visuelle Systeme, oder Teile daraus, klassifizieren beziehungsweise vergleichen kann: Anhand der verwendeten Hardware, d.h. Kameras, anhand der Motivation, die der Steuerung des Systems zugrunde liegt, anhand der Art, wie die Steuerung organisiert ist, oder anhand der für Konfiguration und Kalibrierung verwendeten Methoden.

### 2.1 Hardware

Visuelle Systeme können aus nur einer Kamera bestehen, beispielsweise einer autonomen Pan-Tilt-Zoom-Kamera, das heißt einer Kamera, die durch Schwenken, Neigen und Zoomen einen bestimmten Bereich anvisieren kann. Ein Einkamerasystem mit einer Pan-Tilt-Zoom-Kamera kann einerseits ein Übersichtsbild liefern und andererseits Detailbilder mit höherer Auflösung durch fokussieren auf jeweils interessante Bereiche, wodurch jedoch für diesen Zeitraum der Überblick über die gesamte Szene verloren geht.

Dieses Problem tritt bei der Verwendung von mindestens zwei Kameras nicht auf. Ein oft verwendeter Aufbau ist daher die sogenannte Master-Slave Konfiguration. Sie besteht klassischerweise aus einer statischen Weitwinkelkamera und einer Pan-Tilt-Zoom-Kamera als aktive Kamera. Während die statische Kamera ununterbrochen für ein Übersichtsbild sorgt, kann die aktive Kamera das fokussieren auf interessante Stellen übernehmen. Ein weiteres Konzept ist das des Kameranetzwerkes, bei dem beliebig viele Kameras und die unterschiedlichsten Kameratypen zum Einsatz kommen können. Wobei der Trend von einer einzelnen Kamera, über wenige oder nur stationäre Kameras, hin zu heterogenen Systemen aus vielen unterschiedlichen Kameras und zunehmender Verwendung von Pan-Tilt-Zoom-Kameras geht, was auch den meisten in der Praxis existierenden Kameranetzwerken entspricht. Das in Abbildung 2.1 dargestellte System Pfinder (‘person finder’) (Wren u. a., 1997) ist ein Einkamerasystem. Es dient dem Verfolgen



Abbildung 2.1: Pfänder System mit ALIVE, einem Programm zur Interaktion mit virtuellen Agenten.

von Personen und der Interpretation ihres Verhaltens, beispielsweise zum Dressieren eines virtuellen Hundes. Im  $W^4$  System ('Who? When? Where? What?') (Haritaoglu u. a., 2000) können mehrere Personen detektiert und verfolgt werden, sowie ihre Aktivitäten erkannt und überwacht. Im Video Surveillance and Monitoring (VSAM) Projekt (Collins u. a., 2000) kommt ein System mit unterschiedlichsten Kameras und Sensoren, wie in Abbildung 2.2 zu sehen, zum Einsatz: Farb-Pan-Tilt-Zoom-Kameras, Wärmebild-Kameras, Kameras auf einem Kleinbus, die an unterschiedlichen Orten zum Einsatz kommen können, für Luftbilder Kameras in einem Flugzeug, an den Universitäten Columbia und Lehigh entwickelte 'Omnicaseras' und das 'indoor activity monitoring' System von Texas Instruments. Die verschiedenen Architekturen kann man Abbildung 2.3 und dem Tutorial (Michelsoni u. a., 2010) entnehmen, eine Historie bekannter Systeme findet man in (Hu u. a., 2004).



Abbildung 2.2: Unterschiedliche Kameras, die im System des VSAM Projektes genutzt werden: a) Farb-Pan-Tilt-Zoom; b) Wärmebild; c) Kamera auf einem parkenden Kleinbus; d) Bilder aus der Luft; e) Columbia-Lehigh Omnicamera; f) Texas Instruments 'indoor activity monitoring system'. Aus (Collins u. a., 2000).

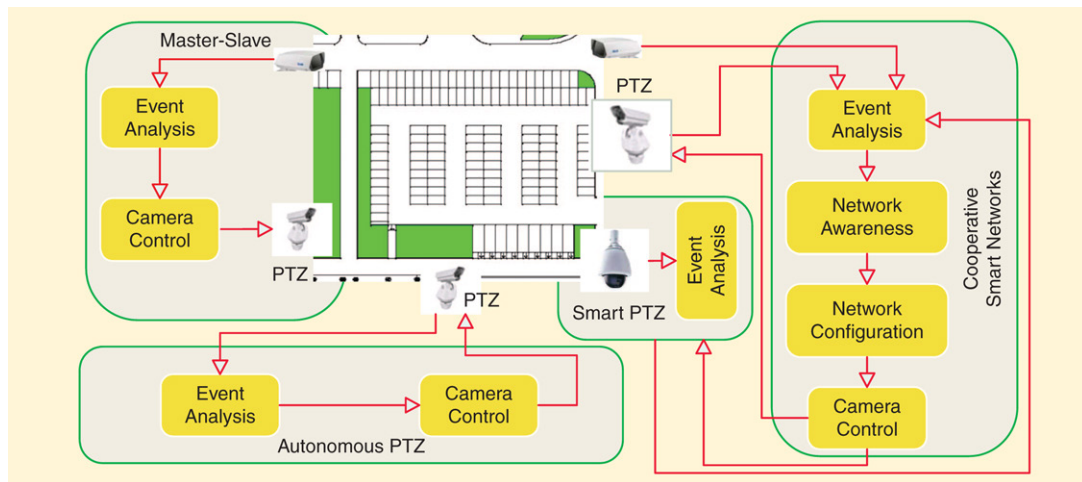


Abbildung 2.3: Verschiedene Architekturen für Kamera-Netzwerke aus (Michelsoni u. a., 2010).

Neben den klassischen Master-Slave-Systemen bestehend aus einer aktiven Pan-Tilt-Zoom- und einer statischen Weitwinkelkamera (Badri u. a., 2007; Liao u. a., 2011; Low u. a., 2011; Senior u. a., 2005; Xiong und Ding, 2012; Yi u. a., 2008) werden oftmals auch mehrere Pan-Tilt-Zoom-Kameras einer statischen Kamera zugeordnet (Bellotto u. a., 2012; Mohanty und Gellaboina, 2011) oder von beiden Kameratypen mehrere Exemplare verwendet (Szwoch u. a., 2011; You u. a., 2010). Der ‘Smart Control Room’ am Fraunhofer IOSB ist beispielsweise mit elf Kameras bestückt, unter denen sich eine Fischaugenkamera und zwei aktive Kameras befinden, (IJsselmuiden und Stiefelhagen, 2010). Das System von (Del Bimbo, Dini, Lisanti u. a., 2010) besitzt keine rein statische Kamera, sondern besteht aus mehreren Pan-Tilt-Zoom-Kameras, siehe Abbildung 2.4, von denen jede, je nach Bedarf, die Rolle des Masters übernehmen kann.

## 2.2 Motivation der Steuerung

Ziel der Steuerung des Systems kann beispielsweise die Objektverfolgung über mehrere Kameras hinweg sein (Everts u. a., 2007). Ein anderes mögliches Ziel ist das Erlangen von qualitativ hochwertigen Aufnahmen bestimmter Objekte oder Details. Welche Anforderungen zum Einstufen der Qualität eine Rolle spielen kann dabei variieren. Es können je nach Aufgabe die Beleuchtung, die Auflösung des Bildmaterials, das Vermeiden von Verdeckung und das Erzielen von Ansichten aus verschiedenen Blickrichtungen eine Rolle spielen. Konkrete Aufgabe wäre beispielsweise das Identifizieren von Autokennzeichen (Mohanty und Gellaboina, 2011; Tian u. a., 2008), oder Aufnahmen, die zur Personen-

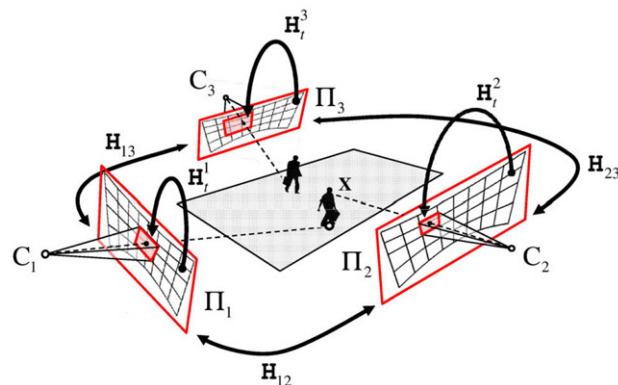


Abbildung 2.4: Paarweise Zusammenhang zwischen mehreren Pan-Tilt-Zoom Kameras aus (Del Bimbo, Dini, Lisanti u. a., 2010). Die Homographien  $H_{12}$ ,  $H_{13}$  und  $H_{23}$  liefern die Transformationen zwischen je zwei der Referenzebenen  $\Pi_1$ ,  $\Pi_2$ ,  $\Pi_3$  der Kameras  $C_1$ ,  $C_2$  und  $C_3$ .  $H_t^1$ ,  $H_t^2$ ,  $H_t^3$  setzen die Referenz-Bildebenen  $\Pi_1$ ,  $\Pi_2$ ,  $\Pi_3$  in Beziehung zur aktuellen Bildebene zum Zeitpunkt  $t$ . Wird das Ziel  $X$  beispielsweise von  $C_1$  als Master überwacht und von  $C_2$  als Slave fokussiert, so werden die Bildkoordinaten von  $X$  mittels  $H_{12}$  von  $\Pi_1$  nach  $\Pi_2$  transformiert und anschließend mittels  $H_t^2$  von  $\Pi_2$  in das momentane Sichtfeld der Kamera  $C_2$ .

identifikation verwendet werden können, im ‘Face cataloger’ (Hampapur u. a., 2003) und in (Marchesotti u. a., 2003; Yi u. a., 2008). In einem System, das automatisch Vorlesungen aufzeichnet (Tsuruoka u. a., 2001), ist die zu erfüllende Aufgabe eine optisch ansprechende Nahaufnahme des Dozenten. Es können auch mehrere Aufgaben zugleich in Angriff genommen werden, so verfolgt das System von (Bellotto u. a., 2012) Personen über mehrere Kameras hinweg und sorgt dabei bei passender Gelegenheit für Nahaufnahmen zu deren Identifikation.

## 2.3 Organisation der Steuerung

Eine weitere Klassifizierung visueller Systeme geschieht nach (Bellotto u. a., 2012) darüber, wie die Steuerung organisiert ist. Hier wird zwischen Steuerung auf Bildbereichsebene (Picture Domain Camera Control (PDCC)), Szenenbereichsebene (Scene Domain Camera Control (SDCC)) und begrifflicher Ebene (Conceptual Level Camera Control (CLCC)) unterschieden, je nachdem, ob das Ziel nur darin besteht die Nahaufnahme eines interessanten Objektes in der Bildmitte zu halten, mit Hilfe von 3D Weltkoordinaten die optimale Kamera zum passenden Zeitpunkt zu verwenden, wenn beispielsweise mehr als ein Objekt zur selben Zeit von Interesse ist, das System mit Verdeckungen zurechtkommen muss, oder Ansichten von verschiedenen Seiten gewünscht sind, oder auf der



begrifflichen Ebene zeitliche und räumliche Informationen verarbeitet werden zur Auswahl der zu fokussierenden Stelle. Zur ersten Gruppe PDCC zählen beispielsweise alle Arbeiten, die einen einfachen Blobdetektor nutzen. In die Kategorie SDCC fällt die Arbeit von (Tsuruoka u. a., 2001), die ein ‘fuzzy control’ Schema verwendet oder die Arbeit (Hampapur u. a., 2003), in der einfache geometrische Schlussfolgerung und statische Regeln Verwendung finden. Unter CLCC fallen die Arbeiten (Bellotto u. a., 2012; Brémond u. a., 2006; Utsumi u. a., 2012; Zouba u. a., 2010).

## 2.4 Konfiguration und Kalibrierung

In jedem Fall eines Mehrkamera-Systems mit einer aktiven Kamera muss bewerkstelligt werden, dass diese auf eine gewünschte Stelle fokussieren kann. Diese Stelle kann entweder in Form von 3D-Weltkoordinaten festgelegt sein, oder durch Bildkoordinaten in einer oder mehreren anderen Kameras festliegen. Im Falle eines Master-Slave-Systems bedeutet dies, dass man zu den Bildkoordinaten des Übersichtsbildes, das die Master-Kamera liefert, passende Pan-Tilt-Zoom-Werte zum Ansteuern der aktiven Kamera benötigt.

Bei der dafür nötigen Konfiguration des Systems unterscheidet man zwischen der sogenannten starken und schwachen Kalibrierung. Bei starker Kalibrierung werden intrinsische und extrinsische Parameter aller beteiligten Kameras bestimmt und daraus der Zusammenhang zwischen 2D-Bildkoordinaten einer Kamera zu Pan-Tilt-Werten einer anderen Kamera über die 3D-Weltkoordinaten des dargestellten Punktes hergestellt. Dabei können die Kameras jede für sich separat kalibriert werden, oder es kann, wie in (Horaud u. a., 2006), eine Kalibrierung als Stereosystem durchgeführt werden. Die Methode von (Tsai, 1987), die von (Marcenaro u. a., 2002; Marchesotti u. a., 2003; Szwoch u. a., 2011) verwendet wird, beinhaltet neben sechs extrinsischen Parametern (drei für Rotation und drei für Translation) noch fünf intrinsische Parameter für die Linse (Brennweite, Verzerrungskoeffizient, zwei Koordinaten des Zentrums der Linsenverzerrung und Skalierungsfaktor) und sechs intrinsische Parameter für den Kamerasensor. Bei (Szwoch u. a., 2011) werden die 3D-Weltkoordinaten zusätzlich in GPS-Koordinaten umgerechnet. Die Arbeit von (Jain u. a., 2006) beachtet zusätzlich Unregelmäßigkeiten in der Rotation preiswerterer Pan-Tilt-Zoom-Kameras und arbeitet mit entsprechend erweiterten Kalibrierungsgleichungen. In (S. N. Sinha und Pollefeys, 2006) wird dagegen ein vereinfachtes Kameramodell verwendet. In (Davis und Chen, 2003) dienen bewegte LED-Lichtpunkte bei der Kalibrierung dem Finden von Punkt-Korrespondenzen. In (Bodor u. a., 2004) wird die Pan-Tilt-Zoom-Kamera als Manipulator mit zwei Drehgelenken und einem virtuellen Translationsgelenk für die Distanz des Kamerazentrums zu den entsprechenden Weltkoordinaten modelliert und inverse Kinematik zur Bestimmung der Pan-Tilt-Werte genutzt. In (Yi u. a., 2008) wird ein handgezeichnetes Gitter zur Bestimmung der extrinsischen Parameter verwendet.

Schwache Kalibrierung dagegen kommt ohne Bestimmung der 3D-Weltkoordinaten aus. Stattdessen wird eine Tabelle von  $x$ - $y$ -Koordinaten im Bild einer Kamera zu Pan-Tilt-Werten einer anderen Kamera erstellt. Diese sogenannte Lookup Tabelle (LUT) wird verwendet um für alle weiteren Bildkoordinaten die zugehörigen Motorkoordinaten zu interpolieren. Zum Erstellen der LUT wird bei (Zhou u. a., 2003) manuell ein bestimmter Punkt im Übersichtsbild in die Bildmitte der aktiven Kamera gesteuert und danach den Bildkoordinaten des Punktes die zur Ansteuerung verwendeten Motorkoordinaten zugeordnet. Per Hand annotierte Rechtecke, die in der Szene stehende Personen umschließen, werden in (Mohanty und Gellaboina, 2011) zum Finden von Punkt-Korrespondenzen zwischen zwei Kamerabildern verwendet. In (Senior u. a., 2005) werden Bewegungsspuren von Fußgängern in der Bodenfläche zur Korrespondenzfindung verwendet und mit einem Modell für Kopfhöhe an bestimmten Stellen sowie einem Modell für zugehörige Pan-Tilt-Werte kombiniert. Zur Interpolation werden beispielsweise in (Liao u. a., 2011) die vertikal und horizontal nächstgelegenen Punkte in der LUT zu einfachen trigonometrischen Berechnungen herangezogen, (You u. a., 2010) nutzt eine gewichtete Interpolationsfunktion und in (Badri u. a., 2007) wird Thin-Plate-Spline (TPS) Interpolation verwendet.

Bildbasierte Merkmale wie Scale-Invariant Feature Transform (SIFT) (D. G. Lowe, 1999) oder Speeded Up Robust Features (SURF) (Bay u. a., 2008) spielen bei der schwachen Kalibrierung eine zunehmende Rolle als Grundlage zur Bildregistrierung.

Andere Verfahren nutzen Eigenschaften der Szene, wie beispielsweise Fahrbahnmarkierungen (Song und Tai, 2006) oder Fluchtpunkte von Straßen (Khan und Shah, 2003). Die Arbeit von (S. Sinha und Pollefeys, 2010) findet Punkt-Korrespondenzen über menschliche Silhouetten und Epipolargeometrie. In (Lv u. a., 2006) werden durch herumlaufende Personen vertikale Liniensegmente der selben Höhe an unterschiedlichen Stellen in der Szene gefunden und genutzt. Auch die menschliche Bewegung als solche kann wie in (Tresadern und Reid, 2008) zur Korrespondenzfindung dienen.

Bei der Merkmalsbasierten schwachen Kalibrierung, die ein gemeinsames Koordinatensystem voraussetzt und somit von ähnlichen Blickrichtungen ausgeht, wird je nach Verfahren die Pan-Tilt-Zoom-Kamera soweit wie möglich herausgezoomt (Liao u. a., 2011; Low u. a., 2011) und danach Merkmals-Vektoren der lokalen Bildmerkmale im Übersichtsbild denen im Bild der Pan-Tilt-Zoom-Kamera zugeordnet. Dies geschieht beispielsweise mittels KD-Tree Korrespondenzfindung (Low u. a., 2011) oder normalisierter Kreuzkorrelation (Liao u. a., 2011). Diese Vorgehensweise hat den Nachteil, dass selbst das größtmögliche Sichtfeld der Pan-Tilt-Zoom-Kamera kleiner sein kann, als jenes der Übersichtskamera. Daher verwenden die Verfahren in (Del Bimbo, Dini, Grifoni u. a., 2008; Del Bimbo, Dini, Lisanti u. a., 2010; S. N. Sinha und Pollefeys, 2006; You u. a., 2010) statt des Bildes der Pan-Tilt-Zoom-Kamera ein mit Hilfe deren Bewegung erstelltes Mosaikbild, das den gesamten Sichtbereich der Master-Kamera abdeckt, wobei zu jedem Einzelbild des Mosaiks die zugehörigen Motorkoordinaten der Pan-Tilt-Zoom-Kamera fest-

gehalten werden. Die Arbeit (Wu und Radke, 2012) schränkt dies auf das Speichern der Merkmals-Vektoren der Einzelbilder anstelle des gesamten Mosaikbildes ein. Bei (Badri u. a., 2007) wird die Pan-Tilt-Zoom-Kamera zu jedem im Übersichtsbild gefundenen Bildmerkmal iterativ dahin gesteuert, dass das zugeordnete Merkmal in der Bildmitte der Pan-Tilt-Zoom-Kamera liegt.

Das Verfahren von (Xiong und Ding, 2012) kommt ohne vorherige Konfiguration aus: es wird ein interessanter Bildbereich im Master-Bild bei Bedarf direkt über Korrespondenzen von SIFT-Merkmalen im Bild der Pan-Tilt-Zoom-Kamera gefunden und zentriert.

Bei dem System in dieser Arbeit handelt es sich um ein Master-Slave-System. Das Verfahren wurde so gewählt, dass es den Vorteil, dass beide Kameras annähernd die selbe Blickrichtung haben, ausschöpft, um eine effiziente Kalibrierung ohne Vorkenntnisse oder zusätzliche Randbedingungen durchführen zu können. Für ein zügiges Verfahren wurde auf das Generieren eines Mosaikbildes verzichtet und stattdessen die aus den Einzelbildern der bewegten Pan-Tilt-Zoom-Kamera ermittelten Merkmale und Korrespondenzen direkt in einer LUT verwertet. Lineare Regression als Interpolation der LUT liefert äußerst robuste Werte und das Vorausberechnen dieser interpolierten Werte in einer flächendeckenden LUT für alle Pixel des Master-Bildes sorgt für eine Laufzeit, die die bisherigen Verfahren schlägt und somit dem Einsatz in der Praxis einen Schritt näher rückt.



# Kapitel 3

## Grundlagen

Im Folgenden werden Grundlagen zu Verfahren erläutert, die in dieser Arbeit eingesetzt werden. Dies sind das Verfahren zum Finden von SURF-Deskriptoren zu lokalen Bildmerkmalen, die zur Bildregistrierung eingesetzt werden, NNDR (nearest neighbor distance ratio) als Methode zum Aussortieren nicht eindeutiger Korrespondenzen und RANSAC (random sample consensus) als Verfahren um mit überbestimmten Systemen und Messwerten, die Ausreißer enthalten, umzugehen. Des Weiteren wird eine Methode zur Bestimmung einer Homographie vorgestellt und die elementaren Grundlagen für Hintergrund-Vordergrund-Detektion sowie einen einfachen Blobdetektor geliefert.

### 3.1 SURF Merkmale

Bei der Bestimmung von SURF-Detektoren (Bay u. a., 2008) werden, ebenso wie bei SIFT (D. G. Lowe, 1999), markante Regionen anhand von lokalen Bildmerkmalen erkannt und danach ein rotations- und skalierungsinvarianter Merkmalsvektor (Deskriptor) zu den erkannten Punkten zusammengestellt. Dieser Vektor kann danach zu einer Korrespondenzsuche (Matching) verwendet werden. Die folgenden Grundlagen sind dem Paper (Bay u. a., 2008) entnommen.

#### 3.1.1 Detektion markanter Regionen

Zum Auffinden von Keypoints, die markante Regionen kennzeichnen, wird eine Hessematrix basierte Merkmals-Detektion durchgeführt.

Die Hessematrix einer Funktion  $f(x, y)$  enthält die zweiten partiellen Ableitungen.

$$H(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \quad (3.1)$$

Bei der Suche nach lokalen Extrema einer stetigen Funktion  $f(x, y)$  dient die Hessematrix als Diskriminante. Ist  $H|_{x,y}$  positiv definit, so liegt an der Stelle  $x, y$  ein lokales Maximum vor und bei einer negativ definiten Matrix ein lokales Minimum. Eine Matrix ist genau dann positiv definit, wenn alle ihre Unterdeterminanten positiv sind. Im vorliegenden Fall der  $2 \times 2$  Hessematrix bedeutet dies:  $\frac{\partial^2 f}{\partial x^2} > 0$  und  $\det(H|_{x,y}) > 0$ . Eine Matrix ist genau dann negativ definit, wenn die Vorzeichen ihrer Unterdeterminanten beginnend mit einem negativen Vorzeichen alternieren. Im vorliegenden Fall der  $2 \times 2$  Hessematrix bedeutet dies:  $\frac{\partial^2 f}{\partial x^2} < 0$  und  $\det(H|_{x,y}) > 0$ .

Anstelle der stetigen Funktion  $f(x, y)$  betrachte man nun die Pixelintensitäten  $I(x, y)$  des Bildes an den Stellen  $x, y$ . Die partiellen Ableitungen zweiter Ordnung können durch Faltung des Bildes mit passenden Filtermasken erzielt werden. Da die zweiten Ableitungen von Intensitätswerten anfällig gegenüber Bildrauschen sind, wird mit einem Gaußfilter geglättet.

Es wird mit  $L_{xx}(x, y; \sigma)$  die Faltung des Bildes mit der zweiten partiellen Ableitung des Gaußkerns  $g(x, y; \sigma)$  in  $x$ -Richtung  $\frac{\partial^2 g(x, y; \sigma)}{\partial x^2}$  bezeichnet. Entsprechend  $L_{yy}(x, y; \sigma)$  in  $y$ - und  $L_{xy}(x, y; \sigma)$  in  $xy$ -Richtung, siehe Abbildung 3.1.

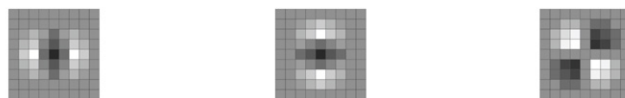


Abbildung 3.1: Filtermasken für  $L_{xx}$ ,  $L_{yy}$  und  $L_{xy}$ .

So erhält man die erweiterte Hessematrix

$$H(x, y; \sigma) = \begin{bmatrix} L_{xx}(x, y; \sigma) & L_{xy}(x, y; \sigma) \\ L_{xy}(x, y; \sigma) & L_{yy}(x, y; \sigma) \end{bmatrix} \quad (3.2)$$

Zur Vereinfachung werden die Filtermasken, wie in Abbildung 3.2 zu sehen, zu  $D_{xx}$ ,  $D_{yy}$  und  $D_{xy}$  approximiert und diskretisiert. Dies erlaubt die Verwendung von Integralbildern zur effizienten Berechnung.

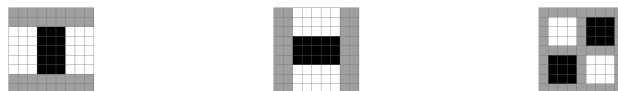


Abbildung 3.2: Approximierte und diskretisierte Filtermasken  $D_{xx}$ ,  $D_{yy}$  und  $D_{xy}$ .

Die  $9 \times 9$  Pixel großen Filter entsprechen einem Skalierungsfaktor  $\sigma = 1,2$ . Durch die

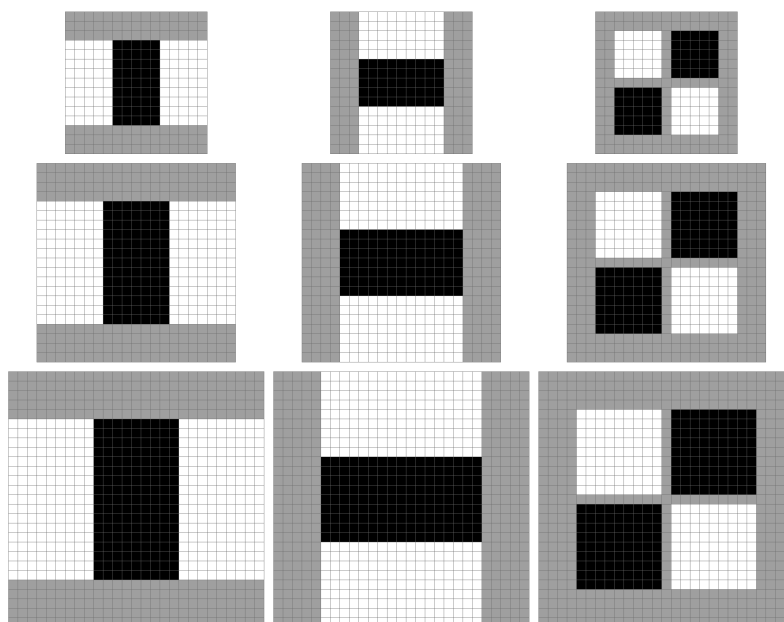


Abbildung 3.3: Skalierung der Filtermasken  $D_{xx}$ ,  $D_{yy}$  und  $D_{xy}$  für den zweiten bis vierten Layer der ersten Oktave.

Approximation muss beim Bestimmen der Determinante noch ein Ausgleichsfaktor eingefügt werden:  $\det(H) = L_{xx}L_{yy} - L_{xy}^2$  wird somit zu

$$\det(H_{approx}) = D_{xx}D_{yy} - (0,9D_{xy})^2 \quad (3.3)$$

Die Berechnung des Ausgleichsfaktors 0,9 anhand der Pixelgröße 9 zu  $\sigma = 1,2$  kann man Gleichung (4) in der Arbeit (Bay u. a., 2008) entnehmen.

Es werden nun mehrere Bildpyramiden aufgebaut durch Skalierung der Filtermatrizen, statt Skalierung des Bildes, wie beispielsweise bei der Bestimmung von SIFT-Deskriptoren. Bei der Skalierung der Filter ergibt sich durch den geometrischen Aufbau, der erhalten bleiben soll, und die Notwendigkeit eines mittleren Pixels die minimal mögliche Vergrößerung von jeweils sechs Pixel, die in Abbildung 3.3 zu sehen ist. Der Skalenraum wird dabei in Oktaven unterteilt. Jede Oktave besteht aus Bildern mit unterschiedlichen Skalen, den Layern. Für jeden Punkt auf dem Bild wird die Determinante der Hessematrix 3.3 aus Filterantworten bestimmt. Dabei werden Antworten unter einem gegebenen Schwellwert ignoriert. Dies ist der ‘hessian threshold’ in der Implementierung, für den Werte von 300 bis 500 abhängig vom Bildkontrast vorgeschlagen werden.

Als nächstes werden potentielle Keypoints mit den 26 Nachbarn der dreidimensionalen Moore-Nachbarschaft, wie in Abbildung 3.4 zu sehen, verglichen, wobei die Skalierung als dritte Dimension fungiert. Nur Positionen mit markanten Werten gegenüber allen

Nachbarn werden dabei gespeichert. Dieser als ‘non-maximum suppression’ bezeichnete Schritt (Neubeck und Van Gool, 2006) sichert im SURF-Verfahren unter anderem die Skalierungsinvarianz.

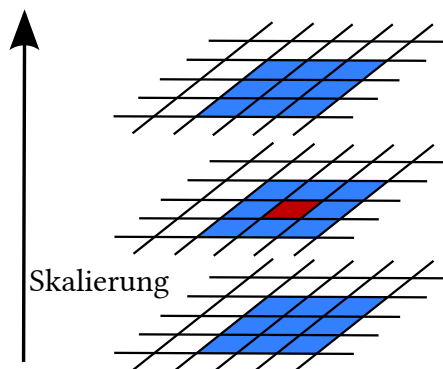


Abbildung 3.4: Die blau markierten Pixel sind im dreidimensionalen Skalenraum die 26 Nachbarn zu dem rot markierten Pixel.

Nun hat man einen Pixel, in dem ein Maximum vorliegt mit einer Blob-Antwort  $B$  (Determinante der Hessematrix) einer  $3 \times 3 \times 3$  Pixel Nachbarschaft. Als nächstes soll die exakte Skalierung und die exakte Position, genauer als auf Pixel gerundet, bestimmt werden. Hierzu wird, wie in (Brown und D. Lowe, 2002) beschrieben, eine 3D Quadrik eingepasst. Man betrachte die Taylorentwicklung verschoben auf den gefundenen potentiellen Keypoint:

$$B(\mathbf{x}) = B + \frac{\partial B^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 B}{\partial \mathbf{x}^2} \mathbf{x} \quad (3.4)$$

mit  $\mathbf{x} = (x, y, \sigma)$  Koordinate im Skalenraum, die den Offset zum als potentieller Keypoint betrachteten Pixel angibt,  $B(\mathbf{x})$  Blob-Antwort an der Stelle  $\mathbf{x}$  und  $B$  sowie die Ableitungen Blob-Antworten an der Stelle des betrachteten Pixels. Durch Nullsetzen der Ableitung dieser Funktion 3.4 kann das Extremum  $\hat{\mathbf{x}}$  bestimmt werden:

$$\hat{\mathbf{x}} = - \frac{\partial^2 B^{-1}}{\partial \mathbf{x}^2} \frac{\partial B}{\partial \mathbf{x}} \quad (3.5)$$

Die Ableitungen werden hierbei durch finite Differenzen von Nachbarpixeln approximiert. Ist in dem so bestimmten Offset  $\hat{\mathbf{x}}$  eine der drei Koordinaten  $x, y, \sigma$  größer 0.5, so bedeutet dies, dass das Extremum näher an einem anderen Pixel liegt und es wird obiger Schritt mit dem Nachbarpixel als potentiell Keypoint wiederholt. Dies geschieht so lange bis alle Koordinaten von  $\hat{\mathbf{x}}$  kleiner 0.5 sind und die Lage des betrachteten Pixels mit Offset  $\hat{\mathbf{x}}$  hinzu addiert als Keypoint vermerkt wird, oder eine festgelegte Anzahl Interpolationsschritte erreicht ist und der als instabil eingestufte Punkt als Keypoint verworfen wird.



### 3.1.2 Bilden des Deskriptors

Zu jedem gefundenen Keypoint wird nun ein Deskriptor gebildet, der beschreibt, wie die Pixelintensitäten in einer zur Skalierung passend gewählten Nachbarschaft verteilt sind. Hierzu werden in einem Umkreis mit Radius  $6\sigma$  Pixel die Gradienten in Form von Haar-Wavelet Filterantworten bestimmt. Die hierbei mit einer Abtaste von  $\sigma$  verwendeten Haar-Wavelets mit Seitenlänge  $4\sigma$  Pixel für  $x$ - und  $y$ -Richtung sind in Abbildung 3.5 dargestellt. Die berechneten Antworten werden als nächstes mit einer Gaußfunktion



Abbildung 3.5: Haar Wavelets. Links für  $x$ -Richtung, rechts für  $y$ -Richtung. Die schwarzen Flächen sind mit 1 gewichtet, die weißen mit  $-1$ .

mit Standardabweichung  $2\sigma$ , wobei  $\sigma$  die detektierte Skalierung bezeichnet, gewichtet und auf den Keypoint zentriert. Danach werden die Antworten wie in Abbildung 3.6 zu sehen, als Punkte im Raum repräsentiert, indem jeweils die horizontale Antwortstärke entlang der Abszisse und die vertikale entlang der Ordinate aufgetragen wird. In einem um den Keypoint rotierenden,  $\frac{\pi}{3}$  breitem Fenster werden jeweils die darin liegenden Punkte zu einem Vektor aufsummiert. Der längste so entstandene Vektor legt die Orientierung des Keypoints fest. Wendet man das SURF Verfahren ‘upright’ an, so wird der zuvor beschriebene Schritt zum Auffinden der Orientierung ausgelassen.

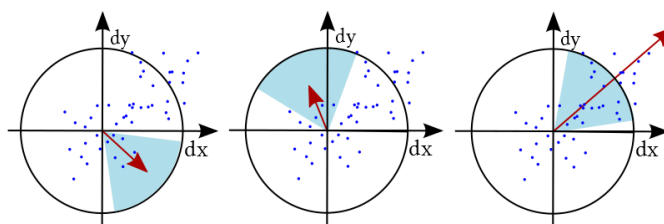


Abbildung 3.6: Zum Bestimmen der Orientierung eines Keypoints rotiert ein  $\frac{\pi}{3}$  breites Fenster um den Ursprung, in dem die als Punkte dargestellten Filterantworten zu einem Vektor aufsummiert werden. Der längste so entstandene Vektor, im obigen Fall der rechte, legt die Orientierung des Keypoints fest.

Über den Keypoint wird ein Quadrat der Größe  $20\sigma \times 20\sigma$  Pixel gelegt und, außer bei ‘upright’, nach der Orientierung ausgerichtet. Das Quadrat wird, wie in Abbildung 3.7 zu sehen, in 16 Unterquadrate unterteilt. In jedem Unterquadrat werden für 25 äquidistante Punkte die Gradienten in  $x$ - und  $y$ -Richtung als Haar-Wavelet Filterantworten bestimmt. Die in Abbildung 3.5 dargestellten Wavelets werden diesmal mit Seitenlänge  $2\sigma$  verwendet und die Antworten mit einer auf den Keypoint zentrierten Gaußfunktion mit Standardabweichung  $3,3\sigma$  gewichtet. Es wird für jedes Unterquadrat ein Merkmalsvektor  $\mathbf{v}$  gebildet, der mit der Bezeichnung der Filterantworten als  $d_x$  und  $d_y$  folgende Form besitzt:

$$\mathbf{v} = \left( \sum_{j=1}^{25} d_x, \sum_{j=1}^{25} d_y, \sum_{j=1}^{25} |d_x|, \sum_{j=1}^{25} |d_y| \right)^T \quad (3.6)$$

Nimmt man alle Merkmalsvektoren der 16 Unterquadrate zusammen, so erhält man für den Keypoint einen 64 Einträge langen Deskriptor. Bei der ‘extended’ SURF-Variante werden doppelt so viele Einträge bestimmt. Dies geschieht durch unterteilen der Summen in Gleichung 3.6 in Teilsummen je nach Vorzeichen der jeweils anderen Filterantwort:

$$\mathbf{v}_{extended} = \left( \sum_{d_y < 0} d_x, \sum_{d_y \geq 0} d_x, \sum_{d_x < 0} d_y, \sum_{d_x \geq 0} d_y, \sum_{d_y < 0} |d_x|, \sum_{d_y \geq 0} |d_x|, \sum_{d_x < 0} |d_y|, \sum_{d_x \geq 0} |d_y| \right)^T \quad (3.7)$$

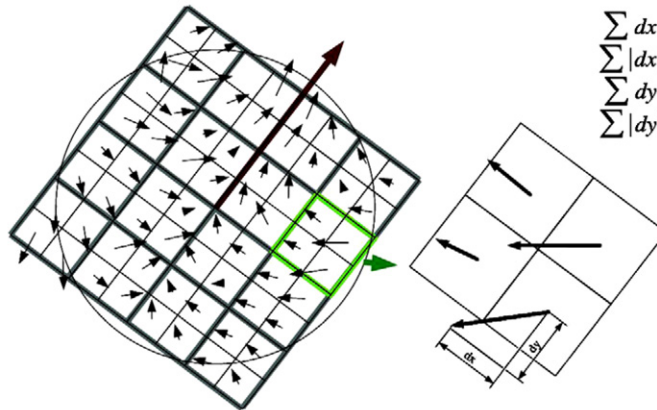


Abbildung 3.7: SURF Deskriptor Aufbau aus (Bay u. a., 2008). Um den gefundenen Keypoint wird, wie links zu sehen, ein Quadrat mit Seitenlänge  $20\sigma$  Pixel gelegt, nach der Orientierung ausgerichtet und in  $4 \times 4$  Unterquadrate unterteilt. Rechte Seite: Für jedes Unterquadrat werden anhand von 25 Punkten darin vier Summen von Haar-Wavelet Filterantworten gebildet. Somit ergibt sich, durch Vereinen der Werte in einem Vektor, ein 64 ( $4 \times 4 \times 4$ ) Einträge langer Deskriptor.

## 3.2 NNDR: Neares Neighbor Distance Ratio

Beim Finden von Korrespondenzen zwischen n-dimensionalen Punkten, wie beispielsweise Deskriptor-Vektoren im Merkmalsraum ist eine Art Schwellwert nötig, um Punkte, zu denen es keine sinnvolle Zuordnung gibt, auszuschortieren und unsichere Zuordnungen wenn möglich ebenfalls auszuschließen. Ein fester Schwellwert hat dabei das Problem, dass seine Größe mitunter schwer zu bestimmen ist und innerhalb des Merkmalsraums stark variieren kann. Eine gute Methode zum Erkennen eindeutiger Zuordnungen liefert das Betrachten des zweit nächsten Nachbarn zusätzlich zum nächsten.

Seien  $D_A$  ein Ziel-Deskriptor,  $D_B$  sein nächster Nachbar und  $D_C$  sein zweit nächster Nachbar, wie in Abbildung 3.8 dargestellt. Es bezeichne  $d_1$  die Distanz zwischen  $D_A$  und  $D_B$  und  $d_2$  die Distanz zwischen  $D_A$  und  $D_C$ . Ist das Verhältnis zwischen den Distanzen nahe 1, so ist eine Verwechslung wahrscheinlich und es kann keine eindeutige Zuordnung durchgeführt werden. Je kleiner das Verhältnis, desto eindeutiger kann die Wahl auf den nächstgelegenen statt den zweit nächst gelegenen Nachbarn fallen.

$$NNDR = \frac{d_1}{d_2} = \frac{\|D_A - D_B\|}{\|D_A - D_C\|} \quad (3.8)$$

Die Gleichung 3.8 liefert das Verhältnis zwischen den beiden Distanzen, bekannt als 'Nearest Neighbor Distance Ratio' (NNDR), das als Schwellwert dienen kann: Eine Korrespondenz zwischen Vektor  $D_A$  und  $D_B$  wird genommen, falls für den, dem Zielvektor  $D_A$  zweit nächstgelegenen, Vektor  $D_C$  gilt:  $\|D_A - D_B\| < \|D_A - D_C\| * th$  für einen bestimmten Schwellwert  $th$  mit  $0 < th < 1$ .

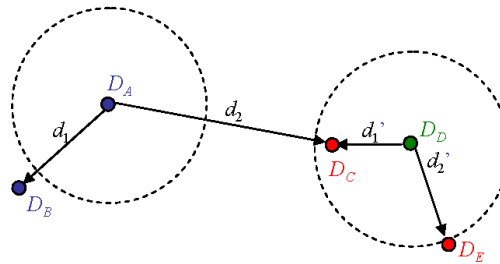


Abbildung 3.8: Bildliche Darstellung von 'Nearest Neighbor Distance Ratio' (NNDR) aus (Szeliski, 2011). Die gepunkteten Kreise stellen einen festen Schwellwert dar, der die Zuordnung  $D_A - D_B$  fälschlicherweise ablehnen würde und der Zuordnung  $D_D - D_C$  fälschlicherweise zustimmen. Mit NNDR würde  $D_A$  korrekt  $D_B$  zugeordnet und  $D_D$  korrekterweise nicht zugeordnet werden, da  $\frac{d_1}{d_2}$  einen kleinen Wert innerhalb eines passend gewählten Schwellwertes liefert und  $\frac{d'_1}{d'_2}$  einen Wert, der außerhalb des selben Schwellwertes liegt.

### 3.3 RANSAC: Random Sample Consensus

‘Random Sample Consensus’, zu deutsch ‘Übereinstimmung mit einer zufälligen Stichprobe’, ist eine Methode, beim Einpassen von Messdaten in ein Modell mit überbestimmten Systemen, d.h. mehr Messdaten als zur Bestimmung nötig sind, und Ausreißern bei den Werten, zurechtzukommen. Der RANSAC Algorithmus wurde erstmals veröffentlicht in (Fischler und Bolles, 1981). Es gibt mittlerweile zahlreiche Varianten des Algorithmus, deren Vergleich man dem Paper (Raguram u. a., 2008) entnehmen kann.

Bei klassischen Methoden, wie beispielsweise der Methode der kleinsten Quadrate, wird versucht ein Modell an alle Messwerte anzupassen. Dadurch können fehlerhafte Werte sowohl die Wahl des Modells als auch das Ergebnis beim Anpassen der Parameter an das Modell so grob verfälschen, dass auch anschließendes Glätten des Ergebnisses nicht ausreicht. Beim RANSAC Algorithmus werden anhand einer Stichprobe aus der Datenmenge die Modellparameter bestimmt. Das so geschätzte Modell wird anschließend für den gesamten Datensatz ausgewertet und die Menge der Daten, die das Modell unterstützen, bestimmt. Diese mit dem Modell übereinstimmenden Daten sind die sogenannten ‘Inlier’, alle anderen, die nicht mit dem Modell konsistent sind, die ‘Outlier’. Dieser Schritt wird so lange wiederholt, bis ein vorgegebener Anteil der Daten mit dem Modell konsistent ist, oder eine vorbestimmte Anzahl von Versuchen überschritten wurde, ohne ein passendes Modell zu finden. Der RANSAC Algorithmus kann anhand von vier Parametern eingestellt werden:

- **Die Stichprobengröße**, d.h. die Anzahl Werte die minimal nötig ist, um die Parameter des gegebenen Modells zu bestimmen.
- **Die Fehlertoleranz** zum Erkennen, ob ein Wert als ‘Outlier’ oder ‘Inlier’ zu betrachten ist.
- **Ein Schwellwert** für das Verhältnis von ‘Inlier’ zu ‘Outlier’, bei dem das Modell als global passend angesehen wird.
- **Die maximale Anzahl an Rekursionsschritten**, nach denen der Versuch, ein passendes Modell zu finden, abgebrochen wird.

### 3.4 Schätzung der Projektivität.

Bei der Schätzung einer Projektivität geht es darum die projektive Transformation zu bestimmen, die Bildkoordinaten  $p_a$  in einem Bild jenen in einem anderen Bild,  $p_b$ , eindeutig zuordnet. Dies ist eine bijektive Abbildung des projektiven Raumes auf sich selbst. Durch Einführung homogener Koordinaten (wie in Gleichung 3.9) handelt es sich um eine lineare Transformation, die sich wie folgt ausdrücken lässt:  $p_b \sim H_{ab}p_a$ . Die Werte

von  $\mathbf{p}_b$  und  $H_{ab}\mathbf{p}_a$  sind dabei durch die Verwendung Homogener Koordinaten nur bis auf einen Skalierungsfaktor gleich. Wobei die Homogenen Koordinaten der Punkte wie folgt bezeichnet seien:

$$\mathbf{p}_a = \begin{bmatrix} x_a \\ y_a \\ 1 \end{bmatrix} \text{ und } \mathbf{p}_b = \frac{\mathbf{p}'_b}{z'_b} = \begin{bmatrix} x_b \\ y_b \\ 1 \end{bmatrix} \text{ mit } \mathbf{p}'_b = \begin{bmatrix} x'_b \\ y'_b \\ z'_b \end{bmatrix} \quad (3.9)$$

$$H_{ab} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (3.10)$$

Wir wollen nun die Matrix  $H_{ab}$  anhand gegebener Punktpaare bestimmen: Im projektiven Raum müssen, wie bereits erwähnt,  $\mathbf{p}_b$  und  $H_{ab}\mathbf{p}_a$  nicht identisch sein, sie können um einen Skalierungsfaktor abweichen. Sie haben allerdings die selbe Richtung und somit ist:  $\mathbf{p}_b \times H_{ab}\mathbf{p}_a = 0$ . Daraus lässt sich durch Umformung ein lineares Gleichungssystem erlangen. Bezeichne  $\mathbf{h}^i$  die  $i$ -te Zeile der Matrix  $H_{ab}$ . Damit ist:

$$H_{ab}\mathbf{p}_a = \begin{bmatrix} \mathbf{h}^{1T} \mathbf{p}_a \\ \mathbf{h}^{2T} \mathbf{p}_a \\ \mathbf{h}^{3T} \mathbf{p}_a \end{bmatrix} \text{ und } \mathbf{p}'_b \times H_{ab}\mathbf{p}_a = \begin{bmatrix} y'_b \mathbf{h}^{3T} \mathbf{p}_a - z'_b \mathbf{h}^{2T} \mathbf{p}_a \\ z'_b \mathbf{h}^{1T} \mathbf{p}_a - x'_b \mathbf{h}^{3T} \mathbf{p}_a \\ x'_b \mathbf{h}^{2T} \mathbf{p}_a - y'_b \mathbf{h}^{1T} \mathbf{p}_a \end{bmatrix} \quad (3.11)$$

Da  $\mathbf{h}^{iT} \mathbf{p}_a = p_a^T \mathbf{h}^i$  lässt sich das Gleichungssystem  $p_{b_j} \times H_{ab} p_{a_j} = 0$  umstellen zu:

$$\begin{bmatrix} \mathbf{0}^T & -z'_{b_j} \mathbf{p}_{a_j}^T & y'_{b_j} \mathbf{p}_{a_j}^T \\ z'_{b_j} \mathbf{p}_{a_j}^T & \mathbf{0}^T & -x'_{b_j} \mathbf{p}_{a_j}^T \\ -y'_{b_j} \mathbf{p}_{a_j}^T & x'_{b_j} \mathbf{p}_{a_j}^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix} = \mathbf{0}. \quad (3.12)$$

Setze:

$$A_j = \begin{bmatrix} 0 & 0 & 0 & -z'_{b_j} x_{a_j} & -z'_{b_j} y_{a_j} & -z'_{b_j} & y'_{b_j} x_{a_j} & y'_{b_j} y_{a_j} & y'_{b_j} \\ z'_{b_j} x_{a_j} & z'_{b_j} y_{a_j} & z'_{b_j} & 0 & 0 & 0 & -x'_{b_j} x_{a_j} & -x'_{b_j} y_{a_j} & -x'_{b_j} \\ -y'_{b_j} x_{a_j} & -y'_{b_j} y_{a_j} & -y'_{b_j} & x'_{b_j} x_{a_j} & x'_{b_j} y_{a_j} & x'_{b_j} & 0 & 0 & 0 \end{bmatrix},$$

somit entspricht die Formel 3.12 einem Gleichungssystem  $A_j \mathbf{h} = \mathbf{0}$  mit einer Matrix  $A_j \in \mathbb{R}^{3 \times 9}$  vom Rang 2 und einem Vektor  $\mathbf{h} \in \mathbb{R}^9$ .

Für vier Punkt-Korrespondenzen  $j = 1, 2, 3, 4$  erhält man  $A \mathbf{h} = \mathbf{0}$  mit einer Matrix  $A \in \mathbb{R}^{12 \times 9}$  vom Rang 8. Die Matrix  $H_{ab}$  kann, unter Verwendung von RANSAC und Lösen von  $A \mathbf{h} = \mathbf{0}$  mit jeweils 4 zufällig gewählten Korrespondenzen, bestimmt werden.

Als Fehlermaß dient dabei der Rückprojektionsfehler:

$$\sum_j \left( x_{b_j} - \frac{h_{11}x_{a_j} + h_{12}y_{a_j} + h_{13}}{h_{31}x_{a_j} + h_{32}y_{a_j} + h_{33}} \right)^2 + \left( y_{b_j} - \frac{h_{21}x_{a_j} + h_{22}y_{a_j} + h_{23}}{h_{31}x_{a_j} + h_{32}y_{a_j} + h_{33}} \right)^2 \quad (3.13)$$

### 3.5 Lineare Regression

Bei der Regression geht es darum, den funktionalen Zusammenhang zwischen bekannten Einflussgrößen  $\mathbf{x} = (x_1, \dots, x_n)$ , den Regressoren, und beobachtbaren Zielgrößen  $\mathbf{y} = (y_1, \dots, y_n)$ , dem Regressant, zu bestimmen. Die folgenden Grundlagen sind dem Buch (T. Arens u. a., 2008) entnommen: Die Systematische Komponente  $\boldsymbol{\mu}(\mathbf{x})$  beschreibt die Wirkung der Einflussgrößen auf die Zielgröße. Zusätzlich wirken noch Störgrößen  $\epsilon$  auf  $\mathbf{y}$  ein. Ist bekannt, dass es sich bei  $\boldsymbol{\mu}(\mathbf{x})$  um eine lineare Funktion handelt, so spricht man von linearer Regression. Es sind in dem Fall anstelle der Systematischen Komponente  $\boldsymbol{\mu}(\mathbf{x})$  lediglich die unbekannten Regressionskoeffizienten  $\boldsymbol{\beta} = (\beta_0; \dots; \beta_m)^T$  in der Strukturgleichung 3.14 zu bestimmen.

$$\mathbf{y} = \boldsymbol{\mu}(\mathbf{x}) + \epsilon = \sum_{j=0}^m \mathbf{x}_j \beta_j + \epsilon \quad (3.14)$$

Dabei wird zur Vereinfachung  $\mathbf{x}_0 = 1$  ergänzt. Die Einflussgrößen können zu einer Designmatrix zusammengefasst werden:  $\mathbf{X} = (\mathbf{x}_0; \dots; \mathbf{x}_m)$ . Sie spannen einen Modellraum auf:

$$\langle \mathbf{X} \rangle = \mathbf{M} = \left\{ \sum_{j=0}^m \mathbf{x}_j \beta_j : \beta_j \in \mathbb{R} : j = 0, \dots, m \right\}. \quad (3.15)$$

Danach ist  $\boldsymbol{\mu} = \sum_{j=0}^m \mathbf{x}_j \beta_j$  äquivalent zu  $\boldsymbol{\mu} \in \mathbf{M}$ . Durch den Modellraum  $\mathbf{M}$ , beziehungsweise die Designmatrix  $\mathbf{X}$  wird das lineare Modell, jedoch noch nicht seine Parametrisierung festgelegt. Als Schätzwert  $\hat{\boldsymbol{\mu}}$  für  $\boldsymbol{\mu}$  wird nun mit der Methode der kleinsten Quadrate der  $\mathbf{y}$  nächstgelegene Wert aus  $\mathbf{M}$  genommen.

$$\hat{\boldsymbol{\mu}} = \arg \min_{\mathbf{m} \in \mathbf{M}} \|\mathbf{m} - \mathbf{y}\|^2 \quad (3.16)$$

Ein Punkt  $\mathbf{m} \in \mathbf{M}$  im Modellraum hat genau dann minimalen Abstand zu  $\mathbf{y}$ , falls er dessen Projektion in den Modellraum ist:  $\mathbf{m} = \mathbf{P}_M \mathbf{y}$ . Der Kleinst-Quadrat-Schätzer  $\hat{\boldsymbol{\mu}}$  ist somit die Orthogonalprojektion von  $\mathbf{y}$  in den Modellraum  $\mathbf{M}$ :  $\hat{\boldsymbol{\mu}} = \mathbf{P}_M \mathbf{y}$ .

Aus  $\mathbf{M} = \langle \mathbf{X} \rangle$  ergibt sich  $\mathbf{P}_M = \mathbf{X} \mathbf{X}^+$  mit der Moore-Penrose-Pseudoinversen  $\mathbf{X}^+$ . Bei der Schätzung bleibt ein orthogonal zum Modellraum stehender Rest  $\hat{\epsilon}$ , das Residuum, übrig. Zusammengefasst ergibt dies:

$$\mathbf{y} = \hat{\boldsymbol{\mu}} + \hat{\epsilon} \text{ mit } \hat{\boldsymbol{\mu}} \in \mathbf{M} \text{ und } \hat{\epsilon} \perp \mathbf{M}. \quad (3.17)$$

Ebenso gilt:

$$\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\epsilon}}. \quad (3.18)$$

Multipliziert man dies mit  $\mathbf{X}^T$  und beachtet, dass  $\mathbf{X}^T \hat{\boldsymbol{\epsilon}} = \mathbf{0}$ , so erhält man:

$$\mathbf{X}^T \mathbf{y} = \mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\beta}} \Rightarrow \hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (3.19)$$

Mit der Gleichung 3.19 hat man nun einen Schätzer für die Modellparameter, mit denen das Modell  $M$ , vorausgesetzt es war sinnvoll gewählt, die gegebenen Messwerte  $\mathbf{x}$  und  $\mathbf{y}$  gut modelliert.

## 3.6 Hintergrund-Vordergrund Segmentierung

Bei der Hintergrund-Vordergrund Segmentierung geht es darum in einer Bildfolge jeweils die Bilder in Vordergrund und Hintergrund zu unterteilen, wobei als Hintergrund die statischen Objekten in der Szene eingeteilt werden und als Vordergrund die beweglichen Objekte erkannt werden sollen. Eine simple Methode wäre ein über die Zeit gemitteltes Bild als Referenzbild, von dem jedes Einzelbild subtrahiert wird und das Ergebnis mit einem Schwellwert verglichen. Diese Methode birgt einerseits das Problem, einen passenden Schwellwert zu finden, da verschiedene Pixel unterschiedlich stark in ihrem Grauwert schwanken. Außerdem werden während der Lernphase zufällig ruhende Vordergrundobjekte oder später entfernte Hintergrundobjekte chronisch als Vordergrund detektiert. Die einfache Methode kommt auch nicht mit Lichtschwankungen und sich wiederholenden Minimalbewegungen wie Blätterrauschen oder eine wackelnde Kamera zurecht. Für ein besseres Verfahren muss das Hintergrundbild kontinuierlich neu bestimmt werden. Dazu betrachtet man die Pixel als stochastischen Prozess. Dieser wird mit einer Gauß-Mischverteilung modelliert.

$$g(x, y) \sim \sum_{i=1}^K w_{i,t} N(\mu_{i,t}(x, y), \sigma_{i,t}^2(x, y)) \quad (3.20)$$

Verschiedene Gaußverteilungen repräsentieren dabei unterschiedliche Farben. Basierend auf der Verweildauer, repräsentiert durch Gewichte, und der Varianz einer jeden Gaußverteilung in der Mischverteilung wird bestimmt welche der Verteilungen zur Hintergrundfarbe gehören könnten. Pixelwerte, die nicht zur Hintergrundverteilung passen, werden als Vordergrund betrachtet, bis eine Verteilung genügen Grund zur Annahme liefert, der Pixel gehöre zum stochastischen ‘Hintergrundprozess’.

Jeder Pixel wird durch eine Gauß-Mischverteilung mit  $K$  Verteilungen modelliert. Die Wahrscheinlichkeit, dass ein Pixel zum Zeitpunkt  $N$  einen bestimmten Wert  $x_N$  annimmt ist:

$$p(x_N) = \sum_{j=1}^K w_j \eta(x_N; \theta_j) \quad (3.21)$$

mit  $w_k$  Gewicht zur  $k$ -ten Komponente der Mischverteilung und  $\eta(x; \theta_k)$  Normalverteilung der  $k$ -ten Komponente, die mit dem Erwartungswert  $\mu_k$  und der Kovarianz der  $k$ -ten Komponente  $\Sigma_k = \sigma_k^2 \mathbf{I}$ , wie folgt lautet:

$$\eta(x; \theta_k) = \eta(x; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)}. \quad (3.22)$$

Die Komponenten der Gauß-Mischverteilung werden absteigend nach  $\frac{w_k}{\sigma_k}$  sortiert und die ersten  $B$  Komponenten als Hintergrundkomponenten betrachtet, wobei:

$$B = \arg \min_b \left( \sum_{j=1}^b w_j > T \right). \quad (3.23)$$

Jeder Pixel, der mehr als die zweieinhalbfache Standardabweichung von jeder der  $B$  Verteilungen entfernt ist, wird als Vordergrundpixel angesehen. Teste  $|g(x, y, N) - \mu_k^N(x, y)| < 2, 5\sigma_k^N(x, y)$ . Die Komponente, die als erste mit dem Testpixel  $g(x, y, N)$  übereinstimmt, wird wie folgt aktualisiert:

$$\begin{aligned} \hat{w}_k^{N+1} &= (1 - \alpha) \hat{w}_k^N + \alpha \hat{p}(\omega_k | \mathbf{x}_{N+1}) \\ \hat{\boldsymbol{\mu}}_k^{N+1} &= (1 - \alpha) \hat{\boldsymbol{\mu}}_k^N + \rho \mathbf{x}_{N+1} \\ \hat{\boldsymbol{\Sigma}}_k^{N+1} &= (1 - \alpha) \hat{\boldsymbol{\Sigma}}_k^N + \rho (\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}}_k^{N+1})(\mathbf{x}_{N+1} - \hat{\boldsymbol{\mu}}_k^{N+1})^T \\ \rho &= \alpha \eta(\mathbf{x}_{N+1}; \hat{\boldsymbol{\mu}}_k^N, \hat{\boldsymbol{\Sigma}}_k^N) \\ \hat{p}(\omega_k | \mathbf{x}_{N+1}) &= \begin{cases} 1, & \text{falls } \omega_k \text{ die erste passende Verteilung} \\ 0, & \text{sonst} \end{cases} \end{aligned}$$

mit einer Zeitkonstante  $\frac{1}{\alpha}$  und  $\omega_k$  der  $k$ -ten Komponente der Gauß-Mischverteilung. Falls keine der  $K$  Verteilungen zu dem Testpixel passt, so wird die mit dem kleinsten Gewicht ersetzt mit dem Wert des Testpixels als Mittelwert, einer hohen Anfangsvarianz und einem niedrigen Gewicht.

Dieses Verfahren basiert auf der Arbeit (Staufer und Grimson, 1999). Es gibt diverse Varianten, unter anderem die Weiterentwicklung von (KaewTraKulPong und Bowden, 2001), bei der Abwandlungen der Aktualisierungsgleichung und der Initialisierung verwendet werden und eine Schattendetektion hinzukam, und die Weiterentwicklung von (Zivkovic, 2004; Zivkovic und Heijden, 2006), in der neben der kontinuierlichen Anpassung der Parameter auch die Anzahl der Komponenten der Gaußmischverteilung für jeden Pixel kontinuierlich angepasst werden.



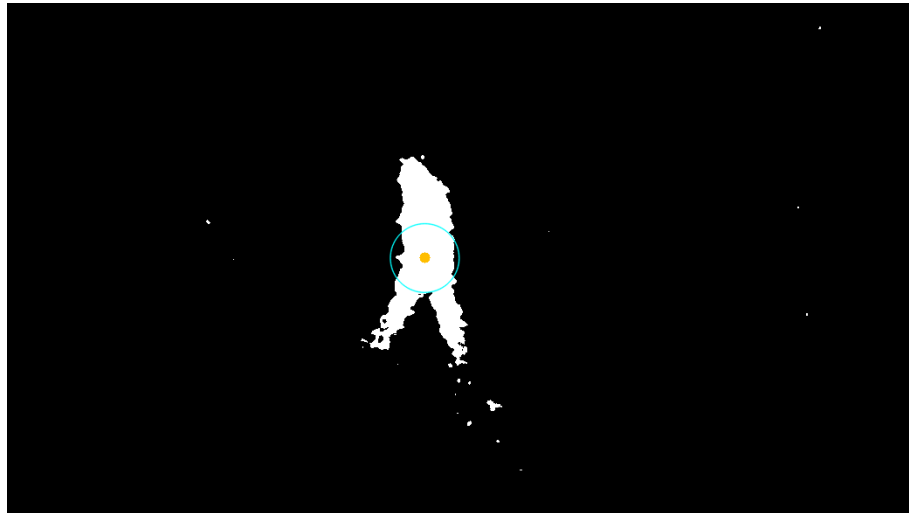


Abbildung 3.9: Ergebnis einer Hintergrundsubtraktion mit einfachem Blobdetektor. Weiße Pixel entsprechen hierbei dem Vordergrund und der hellblaue Kreis einem erkannten Blob.

## 3.7 Blobdetektor

Der verwendete einfache Blobdetektor `SimpleBlobDetector` aus der OpenCV Bibliothek ([Bradski, 2000](#)) geht folgendermaßen vor: Zuerst wird das Bild durch Anwenden diverser Schwellwerte in Binärbilder umgewandelt. Dabei werden Werte von inklusive 'minThreshold' bis 'maxThreshold' (exclusive) mit einer Schrittweite 'thresholdStep' angewandt. Als nächstes werden in jedem dieser Binärbilder Zusammenhangskomponenten mit dem Algorithmus von ([Suzuki und Abe, 1985](#)) bestimmt und ihr jeweiliges Zentrum berechnet. Diese zentralen Punkte werden über alle Binärbilder hinweg anhand ihrer Lage im Bild gruppiert. Dabei bestimmt der Parameter 'minDistBetweenBlobs' welche Punkte noch zusammengehören. Zu jeder gefundenen Gruppe von Mittelpunkten wird ein Gesamtmittelpunkt bestimmt und der Radius. Diese Werte werden als Größe und Lage eines Keypoints von dem Verfahren als Ergebnis geliefert.



# Kapitel 4

## Design

Im Folgenden wird die Hard- und Software des verwendeten Systems vorgestellt. Die Hardware wurde vom Fraunhofer Institut IOSB zur Verfügung gestellt, Grundlegende Teile der Software entstand im Vorfeld der Diplomarbeit am selben Institut.

### 4.1 Hardware

Der Hardwareaufbau besteht aus einem zwei bis drei Meter hohen Stativ, auf dem drei Kameras angebracht wurden: Eine Übersichtskamera vom Typ Axis Q1755, daneben eine Wärmebildkamera vom Typ Axis Q1922 und darunter eine PTZ-Kamera vom Typ Axis P5534, wie in Abbildung 4.1 zu sehen. Die Kameras sind so angebracht, dass ihre Sichtfelder annähernd die selbe Blickrichtung haben.

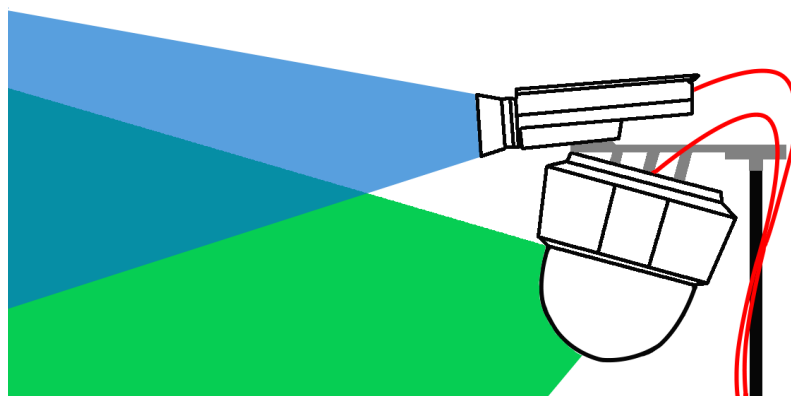


Abbildung 4.1: Aufbau und Blickrichtung von Übersichts- und PTZ-Kamera.

Die Übersichtskamera liefert Bilder der Größe  $1920 \times 1080$  Pixel und die Wärmebildkamera  $640 \times 480$  Pixel. Die PTZ-Kamera lässt sich von  $-180$  bis  $+180$  Grad schwenken und von  $0$  Grad bis  $-180$  Grad neigen. Sie liefert Bilder der Größe  $1280 \times 720$  Pixel. Außerdem können die Übersichts- sowie die PTZ-Kamera zoomen und einen IR-Filter für nahes Infrarot zuschalten. Die Kameras besitzen Autofokus und automatische Blende (Autoiris). Im Anhang B befinden sich Auszüge aus den Datenblättern der Kameras, denen man weitere Details entnehmen kann. Eine Eigenschaft aller verwendeten Kameras ist die Ansteuerung über Gigabit Ethernet. Das HTTP-Protokoll zum Senden und Empfangen von Befehlen und Daten trägt dazu bei, dass die Kameras leicht verzögert auf Befehle reagieren.

## 4.2 Software

Die verwendete Software *CameraSystem* für das System ist, unter Zuhilfenahme der QT Klassenbibliothek, in C++ realisiert. In die Software wurden Teile der vorhandenen Programmibliotheken OpenCV und VCA eingebunden. Durch Multithreading auf hoher Ebene ist das Programm gut skalierbar, beispielsweise in der Anzahl der eingebundenen Kameras, als auch der z.B. pro Kamera eingesetzten Bildbearbeitungsalgorithmen. Bei der Entwicklung der hier verwendeten Software wurde GIT zur Versionsverwaltung eingesetzt und zum effizienten Verwalten von Fehlern ein Bugtracker verwendet. Um die Entwicklung noch weiter möglichst dynamisch zu gestalten und reaktionsfreudig auf kurzfristige Änderungen zu sein, wurden agile Methoden zur Programmierung eingesetzt: z.B. kurze Meilensteine, Extreme Programming und eine ständige Schleife der sich ändernden Anforderungen und der tatsächlichen verfügbaren Funktionalität.

OpenCV (**Open** Source Computer Vision) (Bradski, 2000) ist eine Programmbibliothek in der neueste Forschungsergebnisse in den Bereichen Bildverarbeitung, Photogrammetrie und maschinelles Sehen in möglichst echtzeitfähigen Algorithmen umgesetzt werden. Das OpenCV Projekt wurde 1999 bei Intel von Gary Bradski initiiert und später als quell-offenes Projekt unter BSD-Lizenz von Willow Garage fortgeführt. Die plattformunabhängige Bibliothek enthält über 2500 Algorithmen in den Programmiersprachen C, C++ und Python. Die Software ist in Pakete für die unterschiedlichen Aufgabenbereiche aufgeteilt, beispielsweise Bildverarbeitung, Videoanalyse, Kamerakalibrierung, 2D-Bildmerkmale, Objekterkennung, Programmierung mit Graphikprozessor-Unterstützung, Mosaik- und Panoramabildgenerierung, maschinelles Lernen oder Cluster-Analyse. Neben der OpenCV Bibliothek wurde ebenso die VCA Bibliothek (Video Content Analysis) der Arbeitsgruppe 'Inhaltsbezogene Videoanalyse' am Fraunhofer Institut IOSB verwendet. Die VCA Bibliothek ist eine Templatebibliothek mit grundlegenden extrem performanten Bildverarbeitungsalgorithmen und Methoden des maschinellen Lernens.

QT (<http://qt-project.org/>) ist eine C++-Klassenbibliothek zur Erstellung plattformübergreifender Anwendungen mit graphischer Benutzeroberfläche. QT ist in verschiedenen Betriebssystemen einsetzbar und unterstützt unterschiedliche Graphikplattformen. Eine offizielle Einführung in die Programmierung mit QT kann man in (Blanchette und Summerfield, 2006) finden. Neben einem Toolkit, das vordefinierte Widgets bereitstellt, wie Buttons, Schieberegler oder Textboxen, bietet QT auch Unterstützung für mehrsprachige Anwendungen. QT enthält Module für Datenbanken, Netzwerkanbindung und Nebenläufigkeit mehrerer Threads für parallele Tasks. Außerdem erweitert es die C++ Funktionalität mit Hilfe eines Präprozessors um das ‘Signal-Slot’-Konzept für die Kommunikation zwischen Objekten, welches das Entwurfsmuster ‘Beobachter’ (Gamma u. a., 1995) umsetzt und einen ereignisgesteuerten Programmfluss ermöglicht. Eine kleiner Ausschnitt der verwendeten Signale und Slots ist in dem Diagramm 4.2 dargestellt, Ausschnitte zu dem zugehörigen Programmcode sind im Folgenden aufgeführt.

Ausschnitt aus *CameraPtzControl*, der die Signale und Slots für die Schwenkbewegung der Kamera nach Rechts und Links (Pan) enthält:

```
void CameraPtzControl::on_pushButtonPtzControlRight_clicked()
{
    int stepsize = ui.spinBoxPtzControlStepsize->value();
    emit camSetRelRightLeft(stepsize);
}
void CameraPtzControl::on_pushButtonPtzControlLeft_clicked()
{
    int stepsize = ui.spinBoxPtzControlStepsize->value();
    emit camSetRelRightLeft(-1*stepsize);
}
void CameraPtzControl::camSetRelRightLeftSlot( int value )
{
    emit camSetRelRightLeft(value);
}
```

Wird der Button für ‘Kamera nach rechts bewegen’ `pushButtonPtzControlRight` gedrückt, so sendet das Button-Widget ein `clicked` Signal.

Der Slot `on_pushButtonPtzControlRight_clicked` ist durch Anwendung des ‘Konvention vor Konfiguration’ Paradigmas automatisch mit dem oben erwähnten Signal verbunden. Empfängt nun dieser Slot ein Signal, so wird anhand der Schrittweite, die in dem Widget `spinBoxPtzControlStepsize` eingestellt ist, ermittelt welcher relative Versatz des *pan*-Wertes der Kamera gewünscht ist. Anschließend wird ein Signal zum passenden Ändern des *pan*-Wertes per `emit camSetRelRightLeft(stepsize)` gesendet. Selbiges, nur mit anderem Vorzeichen geschieht beim Button für die Bewegung nach

links. Neben Signalen von Buttons kann die Bewegung auch durch Senden eines beliebigen Signals geschehen. Dieses Signal muss allerdings die passenden Parameter zum Slot `camSetRelRightLeftSlot( int value )`, in unserem Fall ein Parameter vom Typ `int`, enthalten und mit diesem Slot verbunden sein.

Ausschnitt aus *BaseCameraPtzControl*, der den Slot für die Schwenkbewegung der Kamera nach Rechts und Links enthält:

```
void BaseCameraPtzControl::camSetRelRightLeftSlot( int value )
{
    axisCamera.set_rel_right_left(value);
}
```

Der Slot `camSetRelRightLeftSlot( int value )` empfängt Signale mit denen er Verbunden wurde, wie beispielsweise `emit camSetRelRightLeft(value)` aus dem Widget *CameraPtzControl*. Von den Signalen wird ein Parameter vom Typ `int` erwartet. Dieser Parameter wird verwendet, um die Kamera seinen Wert in Grad nach rechts, bzw. bei negativem Vorzeichen nach links zu bewegen.

Ausschnitt aus *CameraPtzControl*, der eine Instanz der Widgetbasisklasse in einem eigenen Thread startet. Die `connect`-Aufrufe wurden auf den, der zu obigem Beispiel einer Signale und Slot Verbindung gehört, eingeschränkt:

```
void CameraPtzControl::startCamera()
{
    BaseCameraPtzControl* camControl =
        new BaseCameraPtzControl(camConfigFile,camName);
    camControl->moveToThread(&camThread);
    camControl->doThreadSetup(camThread);
    ...
    connect(this, SIGNAL(camSetRelRightLeft(int)),
            camControl, SLOT(camSetRelRightLeftSlot(int)));
    ...
    camThread.start();
}
```

In der Methode `startCamera` des Widgets *BaseCameraPtzControl* wird eine neue Instanz der Klasse *BaseCameraPtzControl* erstellt und in einen eigenen Thread verschoben. Die Methode `doThreadSetup` verbindet die nötigen Signale und Slots zwischen dem Thread und der Klasse, wie beispielsweise das Signal `started()` des Threads mit dem Slot, der Klasse in dem die Arbeit aufgenommen wird. Im Aufruf `camThread.start()` wird dieses Signal ausgelöst.

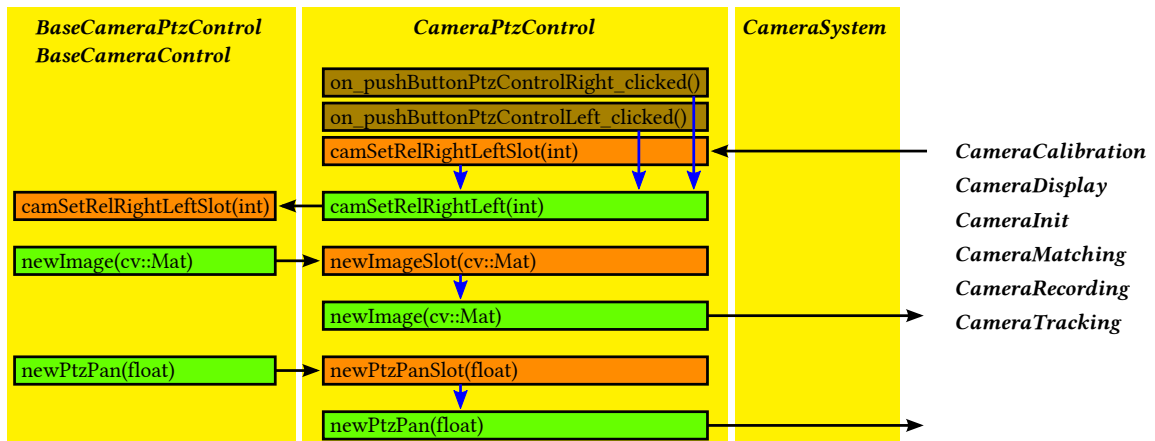


Abbildung 4.2: Ausschnitt des Diagramms der Signal-Slot Verbindungen. Signale in Grün, public Slots in Orange und in Braun private Slots für Signale von Gui-Elementen, wie beispielsweise das Anklicken eines bestimmten Buttons. Blaue Pfeile stellen den Ereignisfluss innerhalb der Widgets dar, schwarze Pfeile verbundene Signale und Slots. Die einzelnen Widgets sind alle über *CameraSystem* verbunden.

Soweit möglich wurde zur besseren Wiederverwendbarkeit einzelner Programmkomponenten das Entwurfsmuster 'Modell-Präsentation-Steuerung' (Gamma u. a., 1995) umgesetzt, um die tatsächliche Programmlogik von der konkreten graphischen Oberfläche zu trennen. Dies unterteilt in Datenmodell, Repräsentation und Programmsteuerung. Die graphische Oberfläche für die Repräsentation wird als QT-Gui-Datei gestaltet und mit Signalen und Slots an die verschiedenen Programmteile angebunden. Voreingestellte Daten können über XML-Konfigurationsdateien eingestellt werden, die beim Starten des Programms automatisch geladen werden. Das Hauptfenster *CameraSystem* verbindet einzelne Widgets durch Verknüpfen von Signalen und Slots zur Kommunikation zwischen den einzelnen Widgets. Auf diese Weise sind die Widgets *CameraControl*, *CameraPtzControl* für die Ansteuerung einer statischen bzw. Pan-Tilt-Zoom-Kamera, *CameraDisplay* für die Anzeige eines Kamerabildes, *CameraRecording* für das Abspeichern von Bildfolgen, *CameraInit* für das Austauschen der Konfigurationsdatei der zu verwendenden Netzwerkkamera, *CameraCalibration* für die Bestimmung der Linsenverzerrung einer einzelnen Kamera, *CameraTracking* für automatische Objektverfolgung und *CameraMatching* für die Durchführung der Systemkonfiguration, angebunden. Die komplette graphische Benutzeroberfläche mit den integrierten Widgets ist in Abbildung 4.3 dargestellt.

Ein Widget besteht aus einer Widgetbasisklasse und einer Widgetguiklasse, beispielsweise *BaseCameraControl* und *CameraControl*, einer Gui-Designdatei und einer Konfigurationsdatei in XML-Format. Die Gui-Designdatei, im Beispielfall ist `cameracontrol.ui`,

enthält die relative Lage der verwendeten Widgets zueinander, ihre Größenskalierung und deren optische Gestaltung wie beispielsweise Rahmen- und Schriftartwahl. Diese Datei kann mit der QT-Creator Software erstellt werden und wird im XML-Format abgespeichert. Die Modellbasisklasse sorgt für die Durchführung der eigentlichen Aufgabe des Widgets. Im Beispiel führt sie in einer permanenten Schleife die Kommunikation mit einer Kamera durch. Diese Klasse wird von der Widgetguiklasse (*CameraControl* im Beispiel) instanziiert und in einen eigenen Thread verschoben. Die Widgetguiklasse regelt die Kommunikation zu *CameraSystem* und zur Widgetbasisklasse, verknüpft die Gui-Elemente mit der gewünschten Funktionalität, lädt Standartwerte über die Konfigurationsdatei und startet bei Bedarf Instanzen der Widgetbasisklasse in jeweils einem neuen Thread. Klassen, die Gui-Elemente verwalten, können nicht in einen eigenen Thread verschoben werden. Die Kommunikation zwischen Widgetguiklasse und Widgetbasisklasse ist wie die Kommunikation der Widgets untereinander über Signale und Slots realisiert, mit einer Ausnahme: Eine mit Mutex gesicherte Variable zum Beenden der Widgetbasisklasse, z.B. für das Stoppen einer Kameraübertragung oder beim Schließen des Programms, unabhängig von den noch gepufferten Signalen. An dieser Stelle war die explizite Behandlung der Synchronität mit einem Mutex erforderlich, da auf diese Variable von unterschiedlichen Threads aus zugegriffen wird und solche Zugriffe nicht von sich aus geordnet ablaufen. Von jedem Widget können im Programm mehrere Instanzen

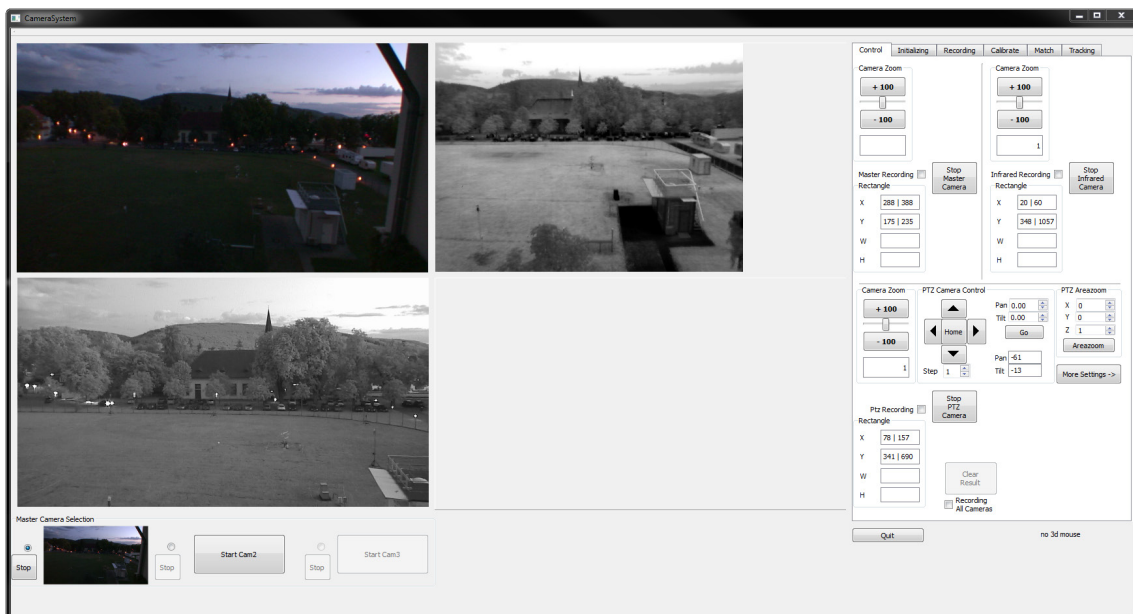


Abbildung 4.3: Gui der verwendeten CameraSystem Software bestehend aus drei *CameraDisplay* Teilen, einem Label für Ergebnisbilder, einem Auswahlfeld für drei Master-Kameras und den Gui-Elementen aller anderen Widgets jeweils als Registerkarte.



verwendet werden, bei *CameraControl* sind dies beispielsweise fünf im Programm, drei für eine Auswahl an Master-Kameras, eine Instanz für eine Pan-Tilt-Zoom-Kamera und eine Instanz für eine Wärmebildkamera.

Im Widget zur Objektverfolgung wurde Graphikprozessor-Unterstützung verwendet (Pham u. a., 2010). Dies ermöglicht es zur Beschleunigung die Berechnung der Hintergrund-Vordergrund Segmentierung auf dem über CUDA angesprochenen Graphikprozessor durchzuführen.



# Kapitel 5

## Umsetzung

Für die Zusammenarbeit von Übersichtskamera und Pan-Tilt-Zoom-Kamera (in unserem Fall ist der Zoom konstant) muss der Zusammenhang zwischen Koordinaten im Bild der Übersichtskamera und Motorkoordinaten, d.h. Pan-Tilt-Werte, der Pan-Tilt-Zoom-Kamera bekannt sein. Um diesen Zusammenhang automatisch zu bestimmen fährt die Pan-Tilt-Zoom-Kamera eine Spirale ab, wobei an jeder der angefahrenen Stellen versucht wird, ein zentrierter Teil des Bildes der Pan-Tilt-Zoom-Kamera in dem Bild der Übersichtskamera einzuordnen. Die so gewonnenen Werte-Quadrupel  $(pan, tilt, x, y)$  bestehen aus Motorkoordinaten der Pan-Tilt-Zoom-Kamera und Bildkoordinaten an denen das Pan-Tilt-Zoom-Bild in das Übersichtsbild eingeordnet wurde. Diese Werte dienen danach der Berechnung einer flächendeckenden Tabelle von Motorkoordinaten zu allen Bildpunkten der Übersichtskamera. Der Gesamttablauf ist in Abbildung 5.2 dargestellt.



Abbildung 5.1: Verwendeter zentrierter Ausschnitt der spiralförmig bewegten Pan-Tilt-Zoom-Kamera.

## 5.1 Spiralförmige Bewegung der Kamera

Damit Werte mit unterschiedlichen Pan-Tilt-Werten ermittelt werden können, muss die Pan-Tilt-Zoom-Kamera bewegt werden. Es gibt zwei verschiedene Methoden, die Kamera zu bewegen: Die Angabe von absoluten Pan-Tilt-Werten und *Areazoom*, eine Funktionalität der Kamera, bei der eine Koordinate im aktuellen Kamerabild als neuer Mittelpunkt des Kamerabildes angegeben wird. Die Methode über Absolutwerte hat den Nachteil, dass dabei nur ganzzahlige Werte angegeben werden können. Beim Verwenden von *Areazoom* können kleinere Bewegungsschritte realisiert werden. Der Nachteil dieser Methode liegt darin, dass bei Tilt-Werten nahe Null Grad, d.h. bei so weit wie möglich nach oben geschwenkter Kamera, das Bild der Kamera einen Bereich von weniger Grad in Pan-Richtung abdeckt, als bei weit nach unten geschwenkter Kamera. Bewegt man die Kamera jeweils ausgehend vom aktuellen Bild um eine Bildbreite nach rechts, eine Bildhöhe nach unten, eine Bildbreite nach links und eine Bildhöhe nach oben, so gelangt man nicht zum Ausgangspunkt. Aus diesem Grund wurde ein erster Ansatz, der wegen der feineren Schrittweite *Areazoom* verwendete, verworfen.

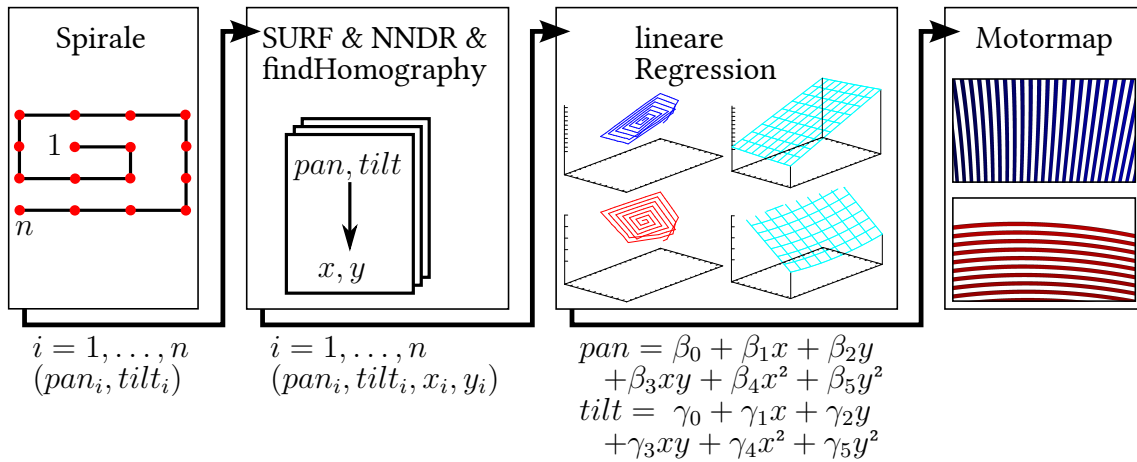


Abbildung 5.2: Übersicht der vier Schritte zur Konfiguration des Systems: Im ersten Schritt wird die aktive Kamera spiralförmig bewegt, um Messwerte mit unterschiedlichen Pan-Tilt-Wert sammeln zu können. Im zweiten Schritt, siehe auch Abbildung 5.6, werden jedem dieser Pan-Tilt-Werte die Korrespondierenden Master-Bild-Koordinaten zugeordnet. Dies geschieht mit Hilfe von SURF-Deskriptoren und Bestimmung einer Homographie zwischen beiden Kamerabildern. Der dritte Schritt ermittelt mit linearer Regression je eine Funktion für  $pan$  und  $tilt$  in Abhängigkeit von  $x, y$  passend zu den Messwerten. Im letzten Schritt wird mit diesen Funktionen eine flächendeckende Lookup-Tabelle als Pan-Tilt-Motorkarte bestimmt.

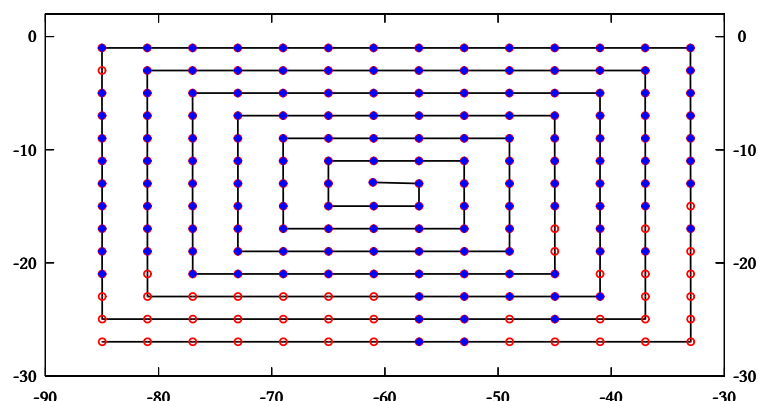


Abbildung 5.3: Pan-Tilt-Werte, die die Pan-Tilt-Zoom-Kamera als Spirale abfährt. Motorkoordinaten, an denen zugehörige Koordinaten im Übersichtsbild gefunden wurden, sind als blaue Rauten, die andern als rote Kreise, markiert.

Die Home-Position der Pan-Tilt-Zoom-Kamera wird so gewählt, dass sie im Sichtfeld der Master-Kamera liegt. Ausgehend von dieser Position wird die Pan-Tilt-Zoom-Kamera mit absoluten Pan-Tilt-Werten spiralförmig bewegt. Um die Bewegung dem Seitenverhältnis des Master-Bildes anzupassen, werden zu Schritten von  $\delta$  Grad nach unten oder oben, Schritte von  $\delta * \text{Bildbreite} / \text{Bildhöhe}$  Grad nach rechts oder links verwendet.

## 5.2 SURF-Merkmale, deren Zuordnung und Ermitteln der Projektivität

In jedem Schritt wird nun ein zentrierter Bildausschnitt der Pan-Tilt-Zoom-Kamera, dessen Größe angegeben werden kann, verwendet, wie in Abbildung 5.1 zu sehen ist. Zusätzlich wird das Übersichtsbild hergenommen und in beiden Bildern jeweils SURF Merkmale nach dem in der Arbeit (Bay u. a., 2008) beschriebenen Verfahren, das in Kapitel 3.1 erläutert wird, ermittelt. Diese lokalen Bildmerkmale der beiden Bilder werden mit Hilfe der FLANN Bibliothek paarweise zugeordnet, wie in Abbildung 5.4 veranschaulicht.

Die gefundenen Zuordnungen von Merkmals-Vektoren werden noch mit Hilfe von 'Nearest Neighbor Distance Ratio' (NNDR) eingeschränkt auf möglichst eindeutige Zuordnungen, wie in (Szeliski, 2011) und in Kapitel 3.2 beschrieben: Eine Zuordnung von Merkmals-Vektor A im Pan-Tilt-Zoom-Bild und B im Übersichtsbild wird weiterverwendet, falls für den zu A im Merkmalsraum zweit nächst gelegenen Vektor C im Master-Bild

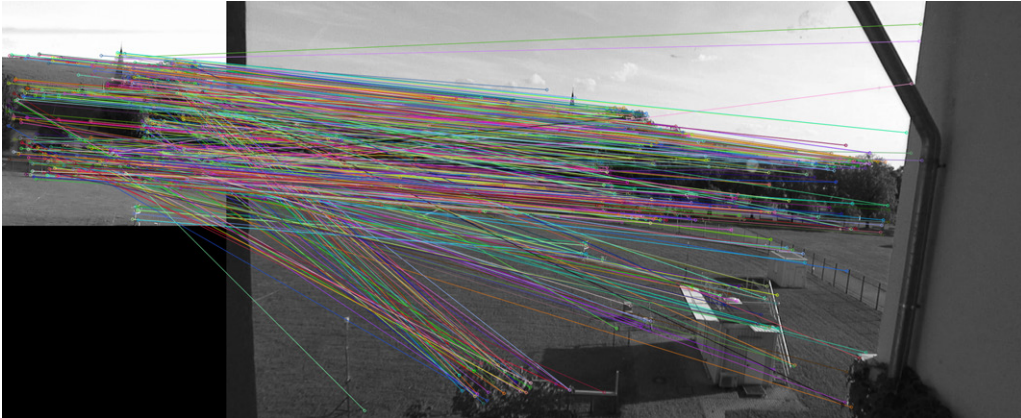


Abbildung 5.4: Korrespondierende SURF Merkmale.

gilt:  $d(A, B) < d(A, C) * ratio$ . Zuordnungen von Merkmals-Vektoren, zu denen es allzu ähnliche Vektoren im Merkmalsraum gibt, so dass es zu Verwechslungen kommen kann, werden damit verworfen. Das erzielte Ergebnis kann man in Abbildung 5.5 erkennen.

Für die verbleibenden Korrespondenzen von Merkmalen wird eine projektive Abbildung bestimmt, welche die Bildpunkte in der Pan-Tilt-Zoom-Kamera auf Master-Bildpunkte abbildet. Die Grundlagen hierzu, kann man Kapitel 3.4 entnehmen. Das grüne Rechteck in Abbildung 5.5 entspricht dem Abbild des Randes des verwendeten Pan-Tilt-Zoom-Teilbildes mit der ermittelten Projektivität.

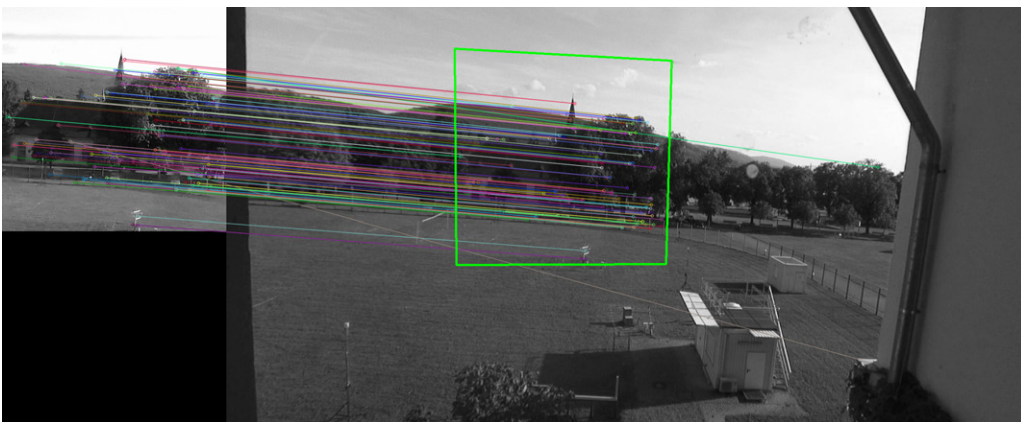


Abbildung 5.5: Korrespondenzen von SURF Merkmalen mit passendem NNDR und Lage des Pan-Tilt-Zoom-Bildausschnittes gemäß der bestimmten Projektivität als grünes Viereck.

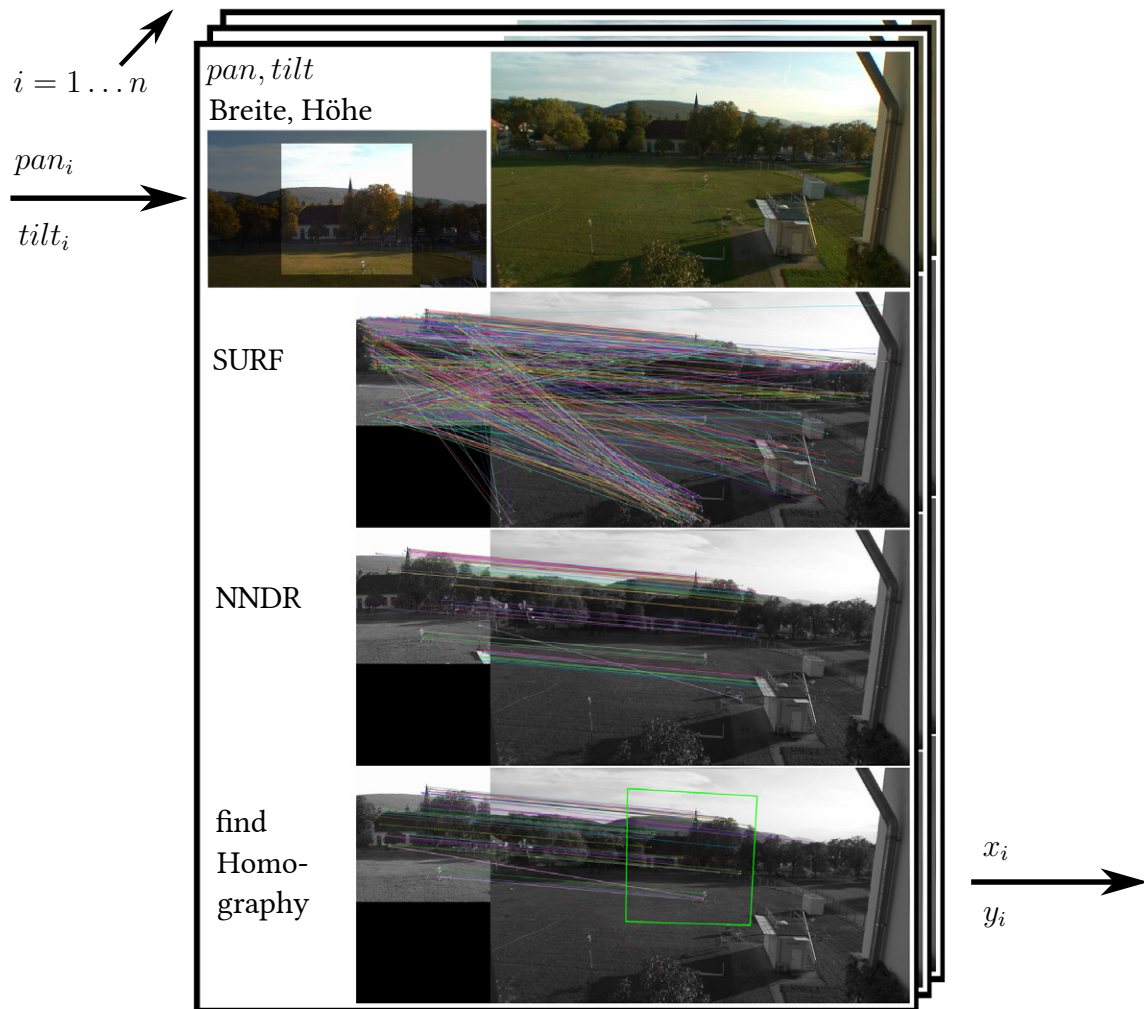


Abbildung 5.6: Übersicht der konsekutiven Schritte zur Erstellung einer flächendeckenden Zuordnungstabelle von Pixeln im Übersichtsbild zu Motorkoordinaten der Pan-Tilt-Zoom-Kamera. Oben: Spirale der angefahrenen Pan-Tilt-Werte. Mitte: An jeder angefahrenen Stelle wird eine Korrespondenz von SURF Merkmalen ermittelt, die gefundenen Merkmale mittels NNDR aussortiert, eine Projektivität daraus bestimmt, mit dieser der Mittelpunkt des Pan-Tilt-Zoom-Bildausschnittes transformiert und das Ergebnis als  $x$ - $y$ -Wert abgespeichert. Unten: Aus den ermittelten Quadrupeln (Pan,Tilt, $x$ , $y$ ) wird je eine lineare Regression für  $x$ , $y$ ,Pan und  $x$ , $y$ ,Tilt durchgeführt.





Abbildung 5.7: Übersichtsbild und Pan-Tilt-Zoom Teilbild, dazu gefundene Korrespondenzen von SURF-Merkmalen und bestimmte Homographie.

Die hierfür genutzte Methode `findHomography` der OpenCV Bibliothek (Bradski, 2000) bestimmt eine projektive Transformation  $H_{sm}$  zwischen der Lage der SURF-Merkmale im Pan-Tilt-Zoom-Bild  $(x_s, y_s)$  und der Lage der zugeordneten Merkmale im Übersichtsbild  $(x_m, y_m)$ , so dass der Rückprojektionsfehler aus Gleichung 3.13 minimiert wird. Um den Einfluss von Ausreißern zu minimieren wird hierbei ein ‘Random Sample Consensus’ Verfahren (RANSAC), wie in Kapitel 3.2 beschrieben, verwendet. Des Weiteren wird das Ergebnis noch mit der Levenberg-Marquardt Methode verbessert. Die resultierende Matrix  $H_{sm}$  ist bereits so normiert, dass in Gleichung 5.1  $h_{33} = 1$  ist.

$$\begin{bmatrix} x'_m \\ y'_m \\ z'_m \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} \Leftrightarrow \mathbf{p}'_m = \mathbf{H}_{sm} \mathbf{p}_s \quad (5.1)$$

Mit den ermittelten korrespondierenden Bildkoordinaten  $(x_{mi}, y_{mi})$  und  $(x_{si}, y_{si})$  der Merkmale für  $i = 1 \dots n$  kann das lineare Gleichungssystem  $\mathbf{A}\mathbf{h} = \mathbf{0}$ , siehe Gleichung 3.12, aufgestellt werden. Für mehr als vier zusammengehörende Messwerte, d.h.  $n > 4$  ist dies ein überbestimmtes Gleichungssystem. Es wird daher mittels ‘Random Sample Consensus’ Verfahren für jeweils vier zufällig ausgewählte Messwert-Quartupel gelöst, wobei die Lösung anhand des Rückprojektionsfehlers aus Gleichung 3.13 bewertet wird.

Durch die Verwendung Homogener Koordinaten ist die Transformationsmatrix  $H_{sm}$  nur bis auf einen Skalierungsfaktor eindeutig bestimmt. Aus diesem Grund sind solche Ma-



trizen üblicherweise normalisiert, das heißt auf eine bestimmte Skalierung festgelegt. Die Methode `findHomography` normalisiert beispielsweise die Matrix durch festhalten eines Elementes so, dass  $h_{33}$  immer gleich 1 ist. Dies kann allerdings zu ungünstigen Werten nahe Null führen. In der Arbeit (Begelfor und Werman, 2005) wird für eine informative Transformationsmatrix eine Normalisierung, die  $\det(H'_{sm}) = 1$  erreicht, nahegelegt.

$$\det H'_{sm} = 1 \quad H'_{sm} = H_{sm} * \frac{\text{sgn}(\det H_{sm})}{\sqrt[3]{|\det H_{sm}|}} \quad (5.2)$$

Damit liegt die Matrix  $H'_{sm}$  in der speziellen linearen Gruppe  $SL_3(\mathbb{R})$ . Dies ermöglicht es, extreme Transformationen über Schwellwerte, wie beispielsweise die in den Ungleichungen 5.3 genannten, zu identifizieren und als Ausreißer zu klassifizieren. Siehe Abbildung 5.8. Die hier verwendeten Werte sind:

$$-5,0 < h'_{11} < 5,0, \quad -5,0 < h'_{22} < 5,0, \quad 0,55 < h'_{33} < 1,5 \quad (5.3)$$

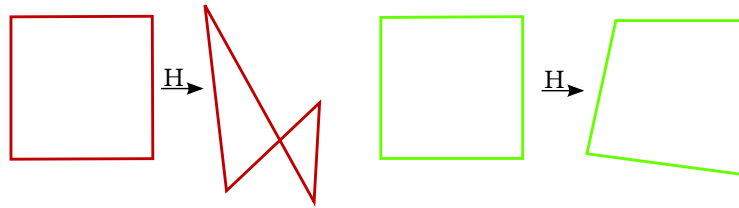


Abbildung 5.8: Zur Weiterverarbeitung abgelehnte und akzeptierte Transformation des Randes des verwendeten Teilbildes der Pan-Tilt-Zoom-Kamera.

Mit der ermittelten Projektivität wird der Mittelpunkt des Pan-Tilt-Zoom-Bildes abgebildet und der so ermittelte  $x$ - $y$ -Koordinaten-Wert des Übersichtsbildes zusammen mit dem Pan-Tilt-Wert der Pan-Tilt-Zoom-Kamera abgespeichert.

## 5.3 Interpolation der Motor- und Bildkoordinaten

Somit erhält man eine Reihe von Pan-Tilt-Werten 5.8a, 5.8b und dazu zugeordnete  $x$ - $y$ -Koordinaten 5.8c, 5.8d im Übersichtsbild. Da jedoch für jedes Pixel des Übersichtsbildes die zugehörigen Pan- und Tilt-Werte benötigt werden, ist noch ein Interpolationsschritt nötig. Als erster Ansatz wurde hier eine Delaunay-Triangulation der Messpunkte im Übersichtsbild durchgeführt. Bei einer Delaunay-Triangulation ist stets die Umkreisbedingung erfüllt, d.h. jeder Kreis durch die Eckpunkte eines beliebigen Dreiecks der Triangulation enthält keinen weiteren Eckpunkt.

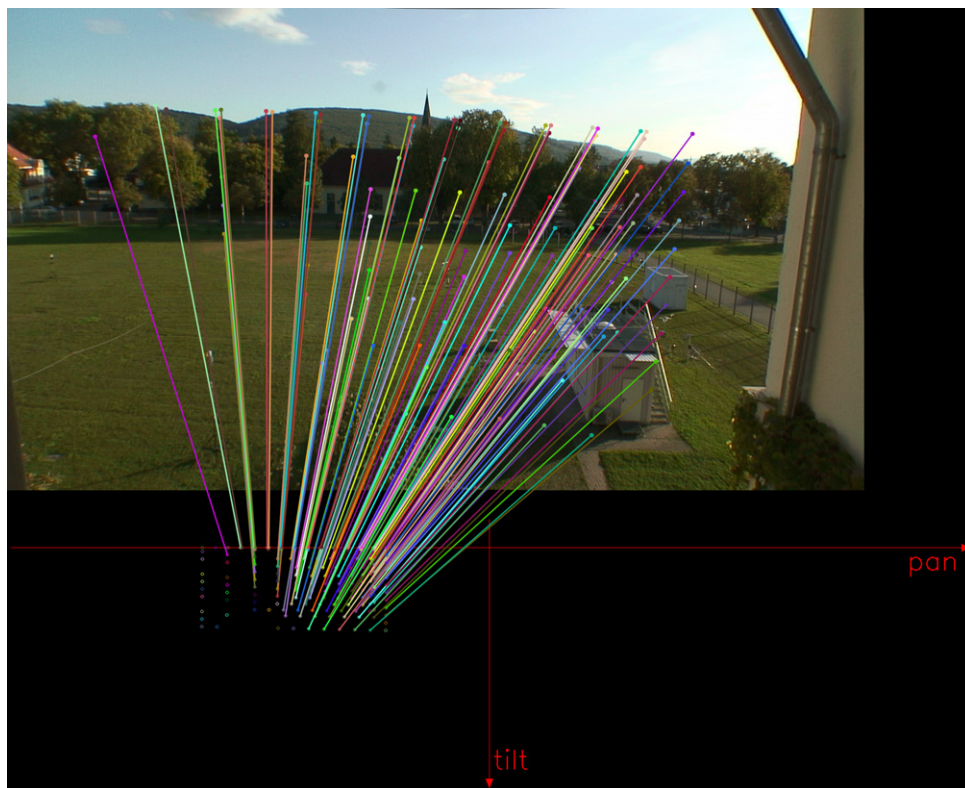


Abbildung 5.9: Gesammelte Werte aus einem spiralförmigen Durchlauf der Pan-Tilt-Zoom-Kamera. Mit jedem der angefahrenen Pan-Tilt-Werte in der unteren Bildhälfte ist die korrespondierende Koordinate, sofern sie ermittelt werden konnte, im Übersichtsbild verbunden.

Damit kann der jeweils zu interpolierende Punkt eindeutig in baryzentrischen Koordinaten bezüglich der drei nächstgelegenen Messpunkte angegeben werden. Hat beispielsweise der Punkt  $D(x_D, y_D)$  die baryzentrischen Koordinaten  $(a, b, c)$  bezüglich der Punkte  $A(x_A, y_A)$ ,  $B(x_B, y_B)$  und  $C(x_C, y_C)$ , so bedeutet dies:

$$x_D = ax_A + bx_B + cx_C \quad (5.4a)$$

$$y_D = ay_A + by_B + cy_C \quad (5.4b)$$

$$a + b + c = 1 \quad (5.4c)$$

Wobei die Bedingung 5.4c dafür sorgt, dass es sich um normierte baryzentrische Koordinaten handelt. Die Pan-Tilt-Werte zu einem beliebigen Master-Bildpunkt werden anhand des Dreiecks aus bekannten Eckpunkten, das ihn umgibt und der baryzentrischen Koor-

dinaten der drei beteiligten Eckpunkte mittels baryzentrischer Interpolation bestimmt.

$$pan(D) = a * pan(A) + b * pan(B) + c * pan(C) \quad (5.5a)$$

$$tilt(D) = a * tilt(A) + b * tilt(B) + c * tilt(C) \quad (5.5b)$$

Dieses Verfahren funktioniert gut für alle Punkte innerhalb des Polygons, das von den Messpunkten aufgespannt wird. Die extrapolierten Werte außerhalb dagegen, sind nicht zufriedenstellend, was man optisch gut an der ermittelten flächendeckenden Motorkarte in Schaubild 5.10 für das Master-Bild erkennen kann.

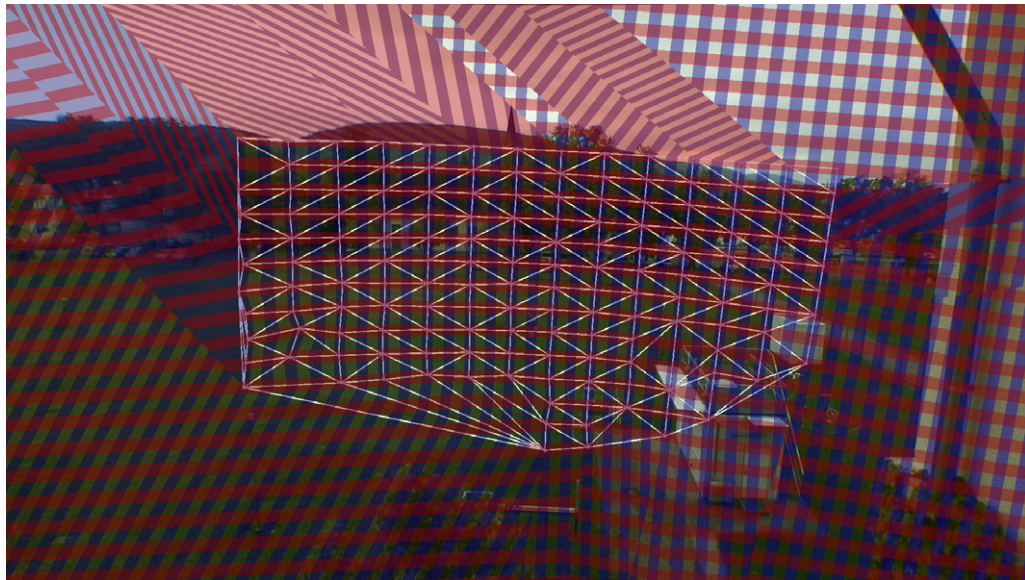


Abbildung 5.10: Delaunay-Triangulation und daraus durch baryzentrische Interpolation entstandene Motorkarte. Ganzzahlige Pan-Werte in Blau und ganzzahlige Tilt-Werte in Rot, jeweils jeden ungeraden Wert transparent gesetzt zur besseren Übersicht.

Als zweites Verfahren wurde jeweils eine lineare Regression für Pan- und Tilt-Werte durchgeführt. Dabei wird je eine quadratische Gleichung für

$$pan = a_{p1} + a_{p2} * x + a_{p3} * y + a_{p4} * x * y + a_{p5} * x^2 + a_{p6} * y^2 \quad (5.6a)$$

und

$$tilt = a_{t1} + a_{t2} * x + a_{t3} * y + a_{t4} * x * y + a_{t5} * x^2 + a_{t6} * y^2 \quad (5.6b)$$

bestimmt.

Einige Tests mit einem Function-Finder-Programm ergaben, dass die Funktion

$$a_{p1} + a_{p2} * x + a_{p3} * y + a_{p4} * x * y + a_{p5} * x^2 + a_{p6} * y * x^2 \quad (5.7a)$$

für Pan und

$$a_{t1} + a_{t2} * x + a_{t3} * y + a_{t4} * x * y + a_{t5} * y^2 + a_{t6} * x * y^2 \quad (5.7b)$$

für Tilt geringfügig besser die gemessenen Werte 5.8 repräsentieren. Da jedoch nur ganzzahlige Werte später an die Pan-Tilt-Zoom-Kamera zur Steuerung geschickt werden können, wirkt sich diese Verbesserung praktisch kaum aus und es wurde darauf verzichtet zwei unterschiedliche Funktionen zu verwenden. Die Grundlagen zur linearen Regression sind in Kapitel 3.5 zu finden.

$$\mathbf{p} = (p_1 \cdots p_n)^T \quad (5.8a)$$

$$\mathbf{t} = (t_1 \cdots t_n)^T \quad (5.8b)$$

$$\mathbf{x} = (x_1 \cdots x_n)^T \quad (5.8c)$$

$$\mathbf{y} = (y_1 \cdots y_n)^T \quad (5.8d)$$

Die passend zu den Gleichungen 5.6a und 5.6b verwendete Designmatrix  $X$  für den Modellraum  $M = \langle X \rangle$  kann man der Formel 5.9 und die zugehörigen Regressionskoeffizienten den Gleichungen 5.10 entnehmen.

$$\mathbf{X} = \begin{pmatrix} 1 & x_1 & y_1 & x_1 y_1 & x_1^2 & y_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n y_n & x_n^2 & y_n^2 \end{pmatrix} \quad (5.9)$$

$$\mathbf{a}_p = \begin{pmatrix} a_{p0} & a_{p1} & a_{p2} & a_{p3} & a_{p4} & a_{p5} \end{pmatrix}^T \quad (5.10a)$$

$$\mathbf{a}_t = \begin{pmatrix} a_{t0} & a_{t1} & a_{t2} & a_{t3} & a_{t4} & a_{t5} \end{pmatrix}^T \quad (5.10b)$$

Damit ergibt sich die Strukturgleichung des linearen Modells zu:

$$(p_1 \cdots p_n)^T = \mathbf{X} \mathbf{a}_p + \boldsymbol{\epsilon}_p \quad (5.11a)$$

$$(t_1 \cdots t_n)^T = \mathbf{X} \mathbf{a}_t + \boldsymbol{\epsilon}_t \quad (5.11b)$$

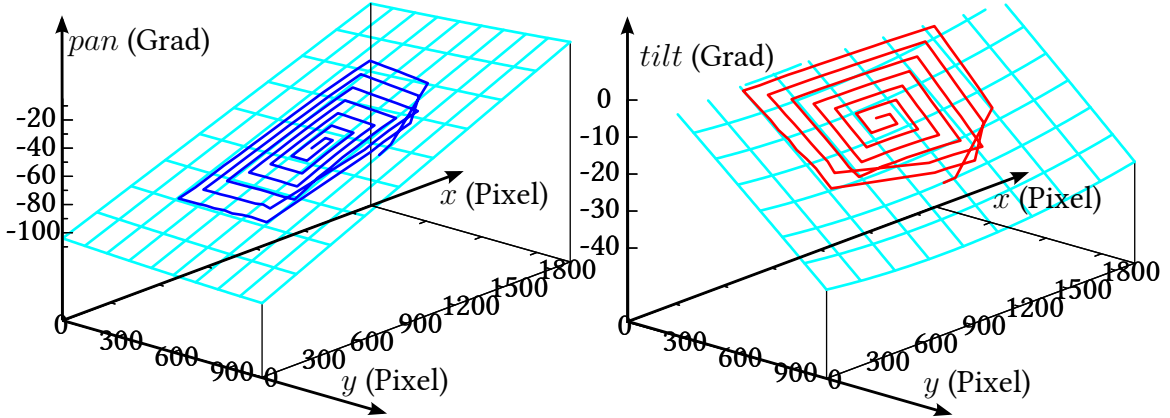


Abbildung 5.11: Quadratische Funktionen als Annäherung an gewonnene Pan- und Tilt-Messwerte in Abhängigkeit der  $x$ - und  $y$ -Koordinaten im Übersichtsbild. Pan-Messwerte in Blau, Tilt-Messwerte in Rot und die jeweiligen Funktionsplots in Hellblau.

Und durch Verwendung der Methode der kleinsten Quadrate ergibt sich die geschätzte Struktur zu:

$$\hat{\mathbf{a}}_p = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T (p_1 \cdots p_n)^T = \mathbf{X}^+ (p_1 \cdots p_n)^T \quad (5.12a)$$

$$\hat{\mathbf{a}}_t = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T (t_1 \cdots t_n)^T = \mathbf{X}^+ (t_1 \cdots t_n)^T \quad (5.12b)$$

Wobei  $\mathbf{X}^+ = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$  die Moore Penrose Pseudoinverse ist.

Damit auch hier Ausreißer in den Werten keinen zu großen Einfluss haben, wird wiederum RANSAC eingesetzt, das heißt das obige Verfahren wird mit einer zufällig gewählten Hälfte der Messwerte angewandt und falls das Ergebnis für einen ausreichenden Anteil aller Messwerte innerhalb einer bestimmten Fehlerschranke liegt verwendet, ansonsten mit neuer zufällig bestimmter Auswahl einer Hälfte der gemessenen Werte wiederholt. Ergebnisse der linearen Regression werden in Abbildung 5.11 veranschaulicht.

## 5.4 Erstellen einer flächendeckenden Motorkarte

Für einen schnellen Zugriff auf die nötigen Motorkoordinaten zum Fokussieren auf einen bestimmten Pixel im Master-Bild wird eine flächendeckenden Lookup-Tabelle für alle Pixel des Master-Bildes anhand der durch lineare Regression bestimmten Gleichungen für Pan 5.6a und Tilt 5.6b vorausberechnet. Eine visuelle Repräsentation dieser Werte kann man in Abbildung 5.12 sehen. Die blauen Streifen stehen für ganzzahlige Pan-Werte, die roten für ganzzahlige Tilt-Werte. Es wurde ein Bildformat mit einem Farbraum von mehr



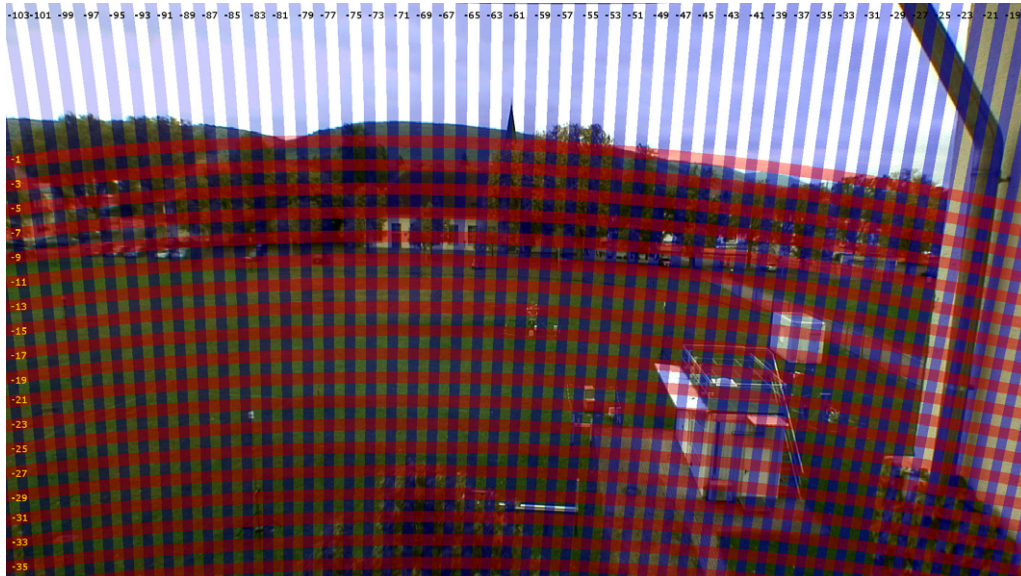


Abbildung 5.12: Flächendeckend berechnete Pan-Tilt-Werte für das Übersichtsbild. Tilt-Werte größer Null sind nicht dargestellt, da die Bewegung der Kamera mechanisch nach Oben beschränkt ist.

als 360 Farben gewählt und zu allen Werten 180 hinzuaddiert, um negative Farbwerte zu vermeiden. Ausserdem wurde, da Farbwertänderungen von  $+1$  nicht mit dem bloßen Auge sichtbar sind, jeder zweite Wert auf transparent gesetzt.

# Kapitel 6

## Experimente und Auswertung

Für Experimente in unterschiedlichen Diskursbereichen wurde sowohl die in Kapitel 4 beschriebene Hardware an diversen Orten aufgestellt, als auch Außen- und Innenkameras des Kameranetzwerkes am Fraunhofer IOSB in Ettlingen verwendet. Im Folgenden wird auf sechs der erfolgreich durchgeführten Experimente, siehe A bis F in Abbildung 6.1 und ein Experiment, das erfolgreich die Grenzen des verwendeten Ansatzes beleuchtet, näher eingegangen.

Bei den Versuchen mit den fest montierten Kameras in den Fluren des Fraunhofer Instituts in Ettlingen fiel einerseits auf, dass die relativ kargen Flure ziemlich wenig lokale



Abbildung 6.1: Für die Experimente verwendete Diskursbereiche. A: Wiese vor dem Institut mit Blick in Richtung Kulisse. B: Selbe Blickrichtung aus dem zweiten Stockwerk. C: Selbe Blickrichtung vom Turm aus. D: Wiese vor dem Institut mit Blick parallel zum Gebäude. E: Eingang des Instituts bei Dunkelheit. F: Flur im Gebäude.

Bildmerkmale liefern. Da zusätzlich diese Kameras mit jeweils mehreren Metern Versatz montiert sind und somit recht unterschiedliche Sichtbereiche abdecken, kam keine hinreichend gute Zuordnung zustande. Mit dem eigenen Kamerastativ konnte im Flur allerdings eine erfolgreiche Konfiguration durchgeführt werden, siehe Experiment F, und vom Kameranetzwerk konnte eine Kameragruppe, die von außen auf den Institutseingang gerichtet ist, in Experiment E sinnvoll verwendet werden. Die Experimente A, B und C entstanden jeweils mit Blick vom Institut in Richtung Süden, wobei einmal das Stativ direkt am Boden stand (A), beim nächsten Mal im Fenster des zweiten Stockwerkes (B) und das dritte Experiment (C) vom Turm auf dem Institut hinab durchgeführt wurde. Experiment D fand ebenfalls mit Stativ am Boden statt, allerdings mit einer anderen Blickrichtung.



Abbildung 6.2: Markierungen zur Fehlermessung bei Experiment A . Links aus Sicht der Master-Kamera, Rechts aus Sicht der Pan-Tilt-Zoom-Kamera nach automatischem Fokussieren auf die Markierung ganz Rechts im Master-Bild.

Bei diversen Experimenten wurde die ermittelte Konfiguration anschließend mit Hilfe von Markierungen evaluiert. Als Markierung wurden Halbkugeln aus Styropor, wie in Abbildung 6.2 zu sehen, verwendet, da man diese einerseits noch gut im Bild der Übersichtskamera erkennen kann und andererseits in der Nahaufnahme der Mittelpunkt leicht zu bestimmen ist. Ein Mausklick auf die Markierung im Master-Bild führt dazu, dass diese Stelle automatisch mit der Pan-Tilt-Zoom-Kamera fokussiert wird. Anschließend wird zuerst gemessen wie viele Pixel die Markierung daraufhin im Bild der Pan-Tilt-Zoom-Kamera vom Mittelpunkt entfernt ist und danach der angefahrene Pan-Tilt-Wert der Kamera mit dem Pan-Tilt-Wert, der sich durch zentrieren der Markierung im Bild ergibt, verglichen.



## 6.1 Experiment A: Am Boden stehendes Stativ mit Blick nach Süden

Für Experiment A stand das Stativ mit dem Kamerasystem an der Südseite des Fraunhofer Instituts auf dem Boden mit Blick vom Gebäude weg. Dort befindet sich eine umzäunte Rasenfläche umgeben von urbanem Hintergrund. Durch das Stativ befanden sich die Kameras auf Augenhöhe eines größeren Menschen. Durch mechanische Einschränkung konnte die Pan-Tilt-Zoom-Kamera nicht höher als Firsthöhe der Gebäude im Hintergrund bewegt werden. Der Rasen im Vordergrund bietet nur wenige lokale Bildmerkmale.

Für die Konfiguration des Systems wurde bei diesem Experiment ein  $700 \times 700$  Pixel großes Teilbild der komplett rausgezoomten Pan-Tilt-Zoom-Kamera gewählt. Es wurde mit  $\delta = 2$  eine Spirale aus 272 Pan-Tilt-Werten abgefahren, von denen zu 125 Werten je eine  $x$ - $y$ -Koordinate im Master-Bild bestimmt und somit ein Daten-Quartupel gebildet werden konnte. Zu den restlichen Werten wurden entweder zu wenige lokale Bildmerkmale gefunden oder die aus den lokalen Bildmerkmalen bestimmte Homographie wurde anhand von Schwellwerten als zu extreme Transformation, wie die in Abbildung 5.8 links, verworfen. Die Spirale ist in Abbildung 6.7 ganz oben dargestellt mit roten Kreisen für die 272 angefahrenen Motorkoordinaten, von denen alle blau gefüllt sind, denen eine  $x$ - $y$ -Bildkoordinate zugeordnet werden konnte. Diese Zuordnung ist in Abbildung

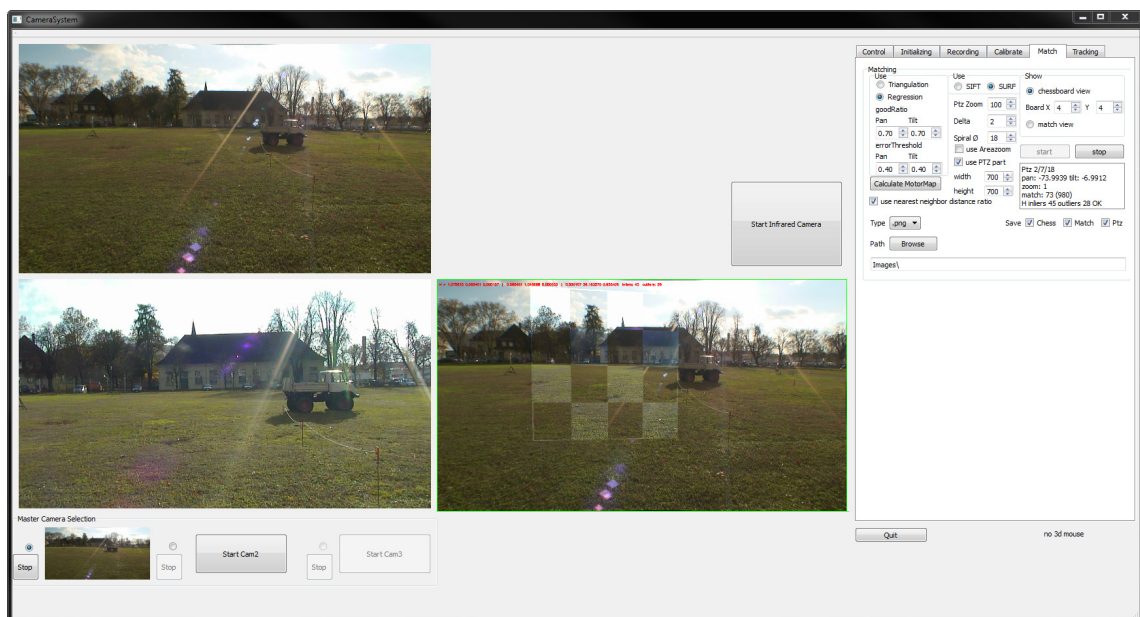


Abbildung 6.3: Gui-Screenshot des Konfigurationsdurchlaufs von Experiment A.

6.7 in der Mitte Links bildlich verdeutlicht. Ein Screenshot der Benutzeroberfläche des Programms während der Konfiguration ist in Abbildung 6.3 zu sehen. Für die lineare Regression waren ein Fehlerschwellwert von 0,4 und ein Mindestanteil von Inliern von 0,7 eingestellt. Die ermittelten Funktionen für Pan und Tilt kann man den Gleichungen 6.1 entnehmen, bzw. den Graphen in Abbildung 6.7 in der Mitte Rechts. Gleichung 6.1a wurde mit 89 Inlier und 36 Outlier nach 771 Iterationsschritten bestimmt, Gleichung 6.1b mit 105 Inlier und 20 Outliern im ersten Schritt.

$$\begin{aligned} pan = & -108,482468 + 0,043942 * x - 0,008725 * y \\ & + 0,000010 * xy - 0,000000 * x^2 + 0,000002 * y^2 \end{aligned} \quad (6.1a)$$

$$\begin{aligned} tilt = & 11,184613 - 0,001869 * x - 0,046444 * y \\ & + 0,000001 * xy + 0,000002 * x^2 + 0,000003 * y^2 \end{aligned} \quad (6.1b)$$

Die daraus berechnete Motorkarte ist in der Abbildung 6.7 unten Rechts veranschaulicht. Unten Links in der Abbildung ist die Triangulation der Bildkoordinaten, zu denen passende Motorkoordinaten bekannt sind, abgebildet. Nach der Konfiguration wurde eine Evaluation mit 72 Markierungen durchgeführt.

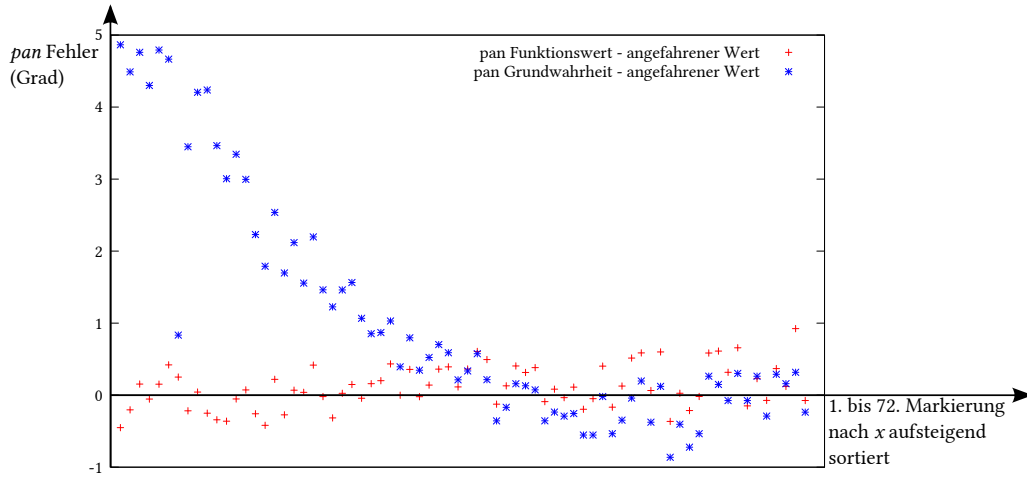


Abbildung 6.4: Pan-Wert Abweichung der 72 Markierungen bei der Evaluation von Experiment A, Dargestellt in Grad zwischen berechnetem Wert und angefahrenem Wert als rote Kreuze sowie zwischen Grundwahrheit und angefahrenem Wert als blaue Sterne. Geordnet nach aufsteigendem  $x$ -Wert.

Das Ergebnis ist in der Graphik 6.6 verdeutlicht und in den Tabellen 6.1, C.1 und C.2 aufgelistet. Vergleicht man die in den Funktionen 6.1 errechneten Werte jeweils mit dem Wert, an dem die Pan-Tilt-Zoom-Kamera steht, nachdem der berechnete Wert gerundet

und als neue absolute Motorkoordinate an die Kamera übermittelt wurde, so kann man bereits eine Schwankung erkennen. Dies sieht man auch an den roten Kreuzen in den Schaubildern 6.4 und 6.5. Die Abweichungen jeweils zwischen der manuell bestimmten Grundwahrheit und dem Wert, an dem die Kamera zu stehen kommt, ist in den selben Schaubildern als blauer Stern markiert.

$x$	$y$	$pan$ laut Funktion	$pan$ laut LUT	$pan$ angefahren	$pan$ Grundwahrheit
28	421	-110,452955	-110	-110,001801	-105,136703
79	455	-108,207425	-108	-108,003304	-103,514000
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1377	620	-44,077634	-44	-44,003700	-44,239899

Tabelle 6.1: Tabelle der  $pan$ -Daten aus der Evaluation von Experiment A. Erste und zweite Spalte Koordinaten im Masterbild, dritte Spalte daraus errechneter Wert nach Funktion 6.1a, der gerundet sowohl in der Motorkarte als auch in der nächsten Spalte eingetragen wird, fünfte Spalte ist der Wert den die Pan-Tilt-Zoom-Kamera angenommen hat, nachdem ihr der Wert der vorherigen Spalte als Anweisung geschickt wurde und in der letzten Spalte ist der Wert der Grundwahrheit verzeichnet. Die gesamten Tabellen für  $pan$  und  $tilt$  jeweils mit allen 72 Einträgen befinden sich in Anhang C.

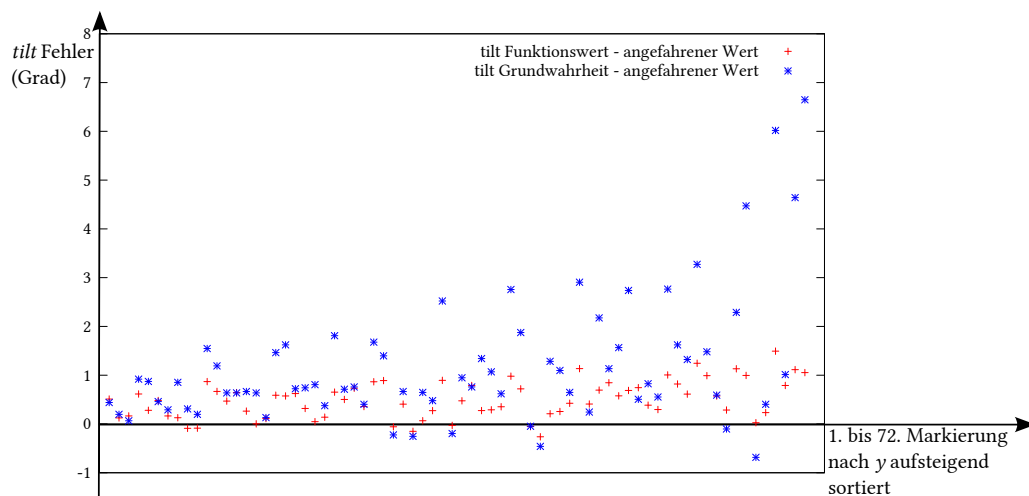


Abbildung 6.5: Tilt-Wert Abweichung der 72 Markierungen bei der Evaluation von Experiment A, Dargestellt in Grad zwischen berechnetem Wert und angefahrenem Wert als rote Kreuze sowie zwischen Grundwahrheit und angefahrenem Wert als blaue Sterne. Geordnet nach aufsteigendem  $y$ -Wert.

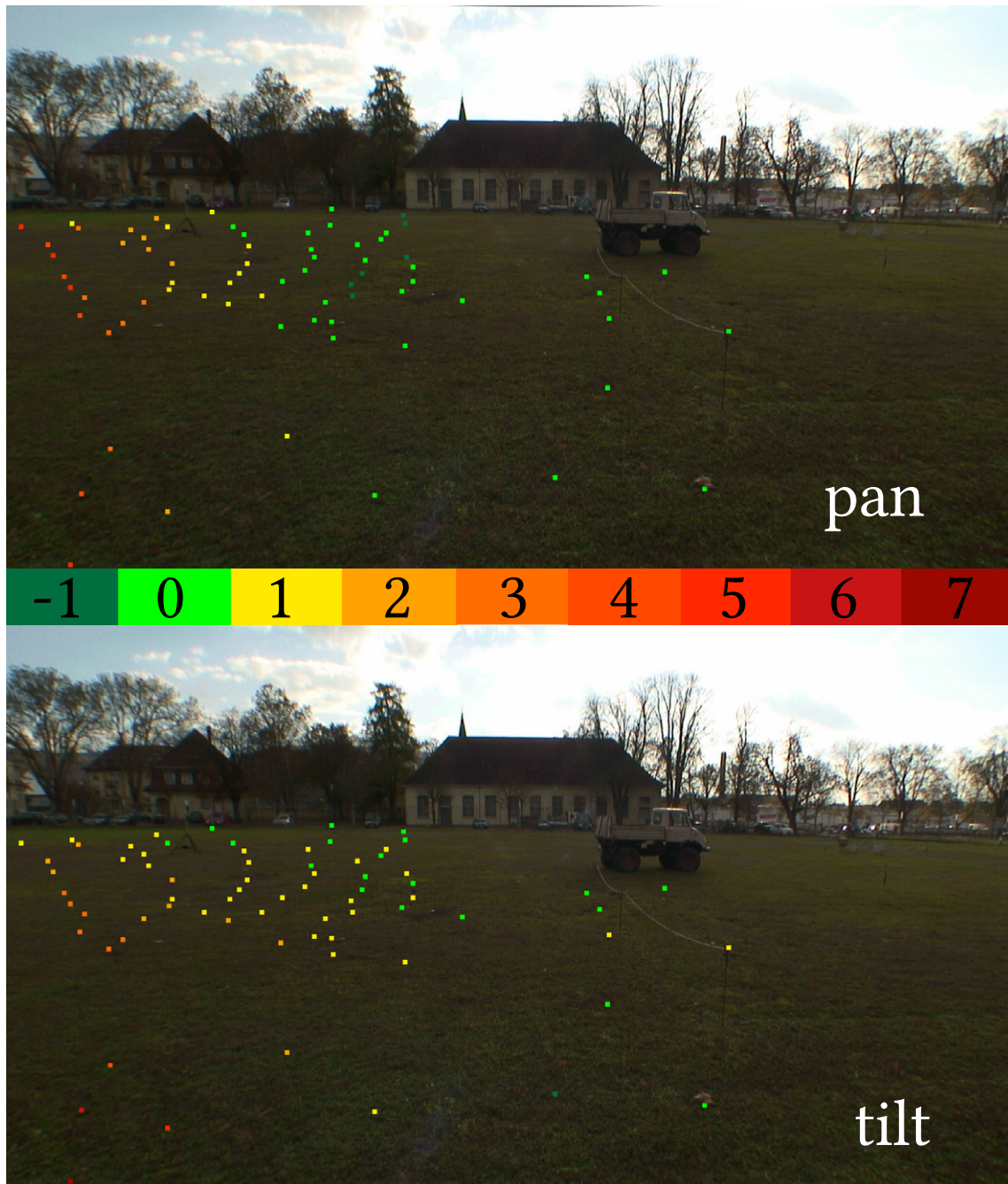


Abbildung 6.6: Fehlermap von Experiment A. Dargestellt sind die Fehler zwischen dem gerundeten Konfigurationsergebnis, wie es in der Motorkarte steht, und der gerundeten Grundwahrheit.



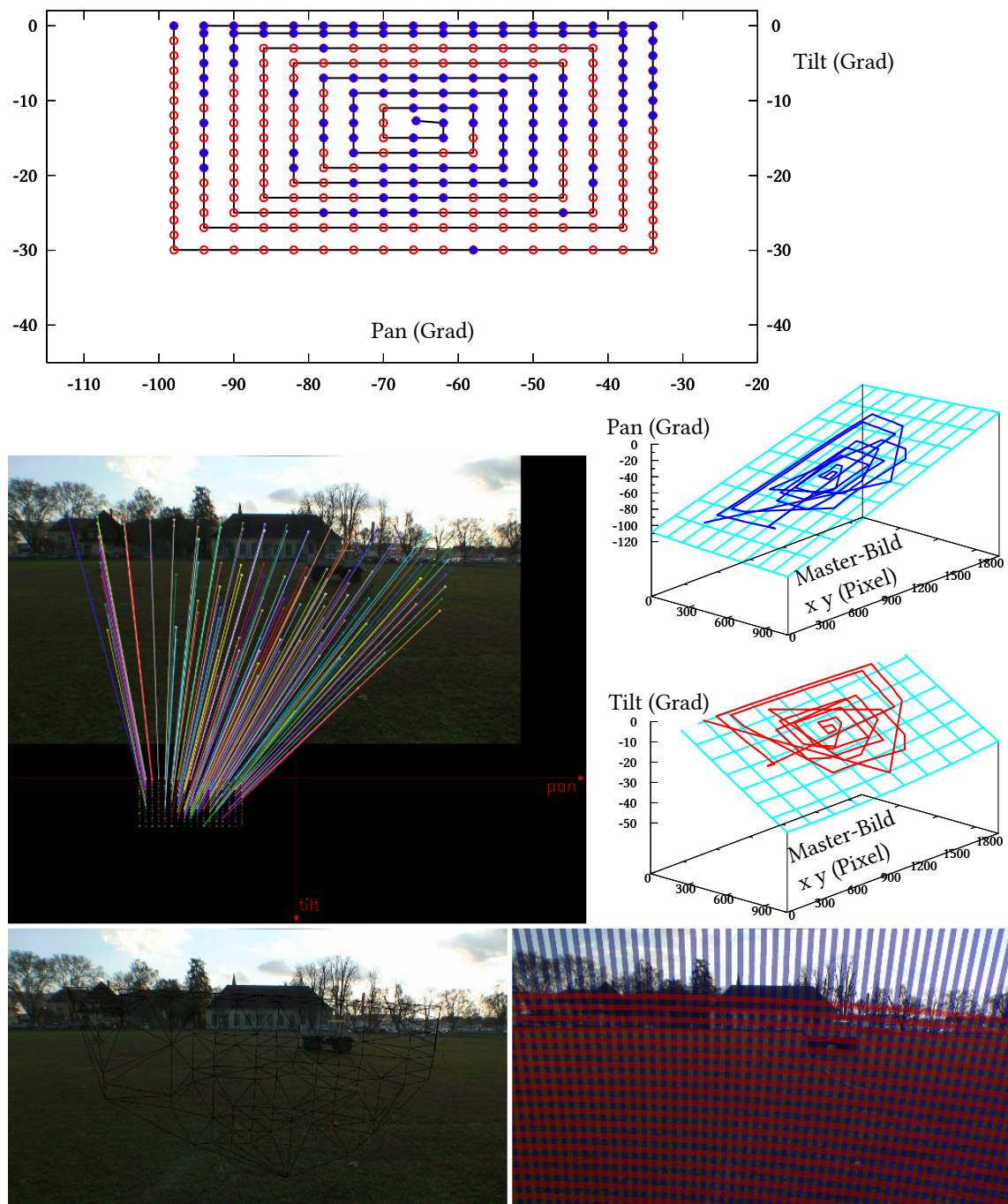


Abbildung 6.7: Ergebnisbilder auf der Wiese in Richtung Kulisse.

## 6.2 Experiment B: Stativ im zweiten Stockwerk mit Blick nach Süden

Bei der Durchführung des Experiments B war das Stativ mit dem Kamerasystem im zweiten Stock in einem der Fenster an der Südseite des Instituts aufgestellt. Die Richtung, in welche die Kameras ausgerichtet waren, ist somit die selbe wie in Experiment A, nur von einem höher gelegenen Blickpunkt ausgehend. Bei diesem Experiment war die Sicht der Pan-Tilt-Zoom-Kamera neben der mechanischen Begrenzung nach oben zusätzlich durch optische Einschränkung beeinträchtigt, die das Fenster zu beiden Seiten verursachte. Hinzu kam, dass eine Fenstersprosse im unteren Bereich im Bild der Pan-Tilt-Zoom-Kamera für lokale Bildmerkmale sorgte, zu denen keine Zuordnung im Master-Bild gefunden werden konnte. Beispielbilder hierzu sind in Abbildung 6.8 zusammengestellt.



Abbildung 6.8: Problemfälle Fenstersprosse und Rand des Fensters in der Pan-Tilt-Zoom-Kamera.

Bei diesem Experiment wurden von der Pan-Tilt-Zoom-Kamera Teilbilder der Größe  $614 \times 614$  verwendet. Der gewählte  $\delta$ -Wert war wiederum 2 und es wurde damit, wie in Abbildung 6.11 oben zu sehen, eine Spirale von 361 Pan-Tilt-Werten abgefahren. Zu 152 dieser Positionen der Pan-Tilt-Zoom-Kamera konnten, wie in Abbildung 6.11 Mitte Links zu sehen, zugehörige Bildkoordinaten im Master-Bild bestimmt werden.

$$\begin{aligned} pan = & -106,403435 + 0,042925 * x - 0,007538 * y \\ & + 0,000010 * xy + 0,000001 * x^2 - 0,000000 * y^2 \end{aligned} \quad (6.2a)$$

$$\begin{aligned} tilt = & 12,291303 - 0,004598 * x - 0,044936 * y \\ & - 0,000001 * xy + 0,000004 * x^2 + 0,000001 * y^2 \end{aligned} \quad (6.2b)$$

Die lineare Regression wurde mit einem Fehlerschwellwert von 0,3 und einem Mindestanteil Inlier von 0,8 durchgeführt. Die so ermittelten Funktionen für  $pan$  und  $tilt$  sind in den Gleichungen 6.2 notiert und in der rechten Mitte der Abbildung 6.11 dargestellt. Bei der Bestimmung der  $pan$ -Funktion, die nach 67 Iterationsschritten abbrach, traten 122 Inlier und 30 Outlier auf. Bei der  $tilt$ -Funktion waren es 135 Inlier und 17 Outlier direkt nach dem ersten Schritt. Es wurde eine Evaluation mit 38 Markierungen durchgeführt.



Abbildung 6.9: Fehlermap zu Experiment B. Links Pan, Rechts Tilt. Dargestellt sind die Fehler zwischen dem gerundeten Konfigurationsergebnis, wie es in der Motorkarte steht, und der gerundeten Grundwahrheit. Es wurde die selbe Farbskala wie in Abbildung 6.6 verwendet, woraus hier jedoch nur die Werte  $-1$  (Gelb),  $0$  (Hellgrün) und  $+1$  (Dunkelgrün) vorkommen.

Dabei wurden die Positionen der Markierungen zu weit vom Bildrand entfernt gewählt, weswegen das Ergebnis recht gut ausfiel, da keine Evaluation der kritischeren Randbereiche stattfand. Eine Tabelle der Daten bezüglich der Pan-Werte der 38 evaluierten Punkte im Master-Bild ist im Anhang unter C.3 zu finden, selbiges zu den Tilt-Werten unter C.4.

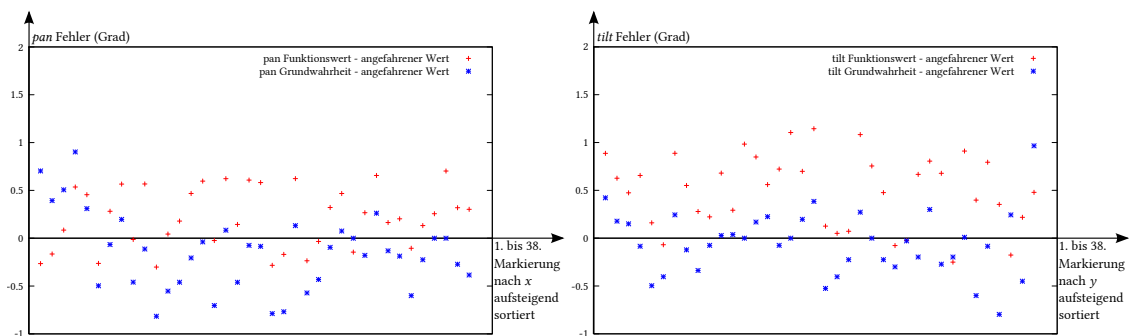


Abbildung 6.10: Abweichung der 38 Markierungen bei der Evaluation von Experiment B, Links für Pan und rechts für Tilt, jeweils dargestellt in Grad zwischen berechnetem Wert und angefahrenem Wert als rote Kreuze sowie zwischen Grundwahrheit und angefahrenem Wert als blaue Sterne. Links geordnet nach aufsteigendem  $x$ -Wert, rechts nach aufsteigendem  $y$ -Wert.

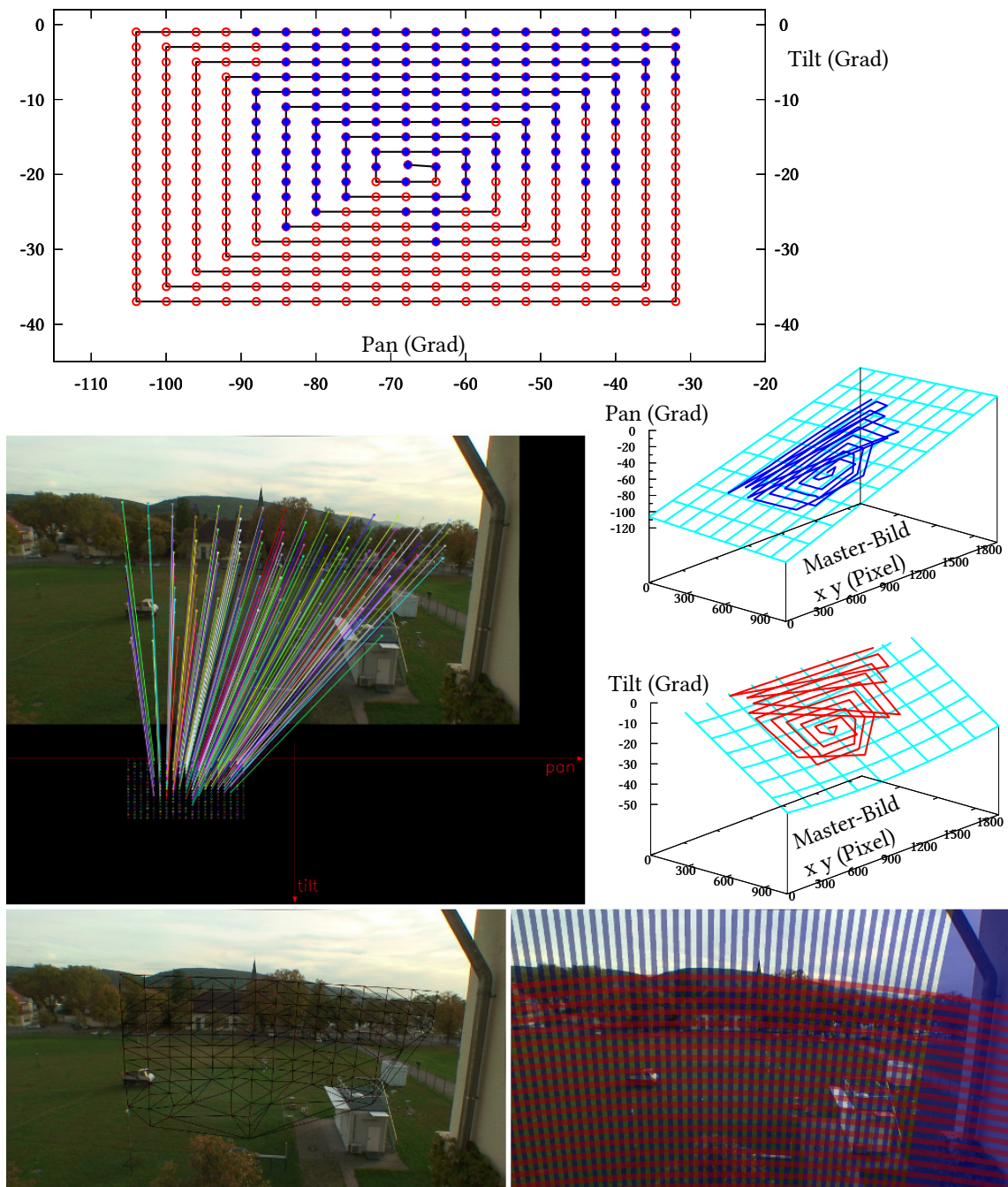


Abbildung 6.11: Ergebnisbilder aus dem zweiten Stock in Richtung Kulisse.



## 6.3 Experiment C: Stativ auf dem Turm mit Blick nach Süden

Für das nächste Experiment wurde ausgenutzt, dass sich auf dem Institutsgebäude ein kleiner Turm mit Balkon befindet. Auf diesem wurde das Stativ mit den Kameras mit Blick Richtung Süden postiert. Die verwendeten Teilbilder der aktiven Kamera haben eine Größe von  $600 \times 600$  Pixeln und entstanden ohne Zoomen der Kamera. Als  $\delta$  wurde abermals der Wert 2 gewählt.

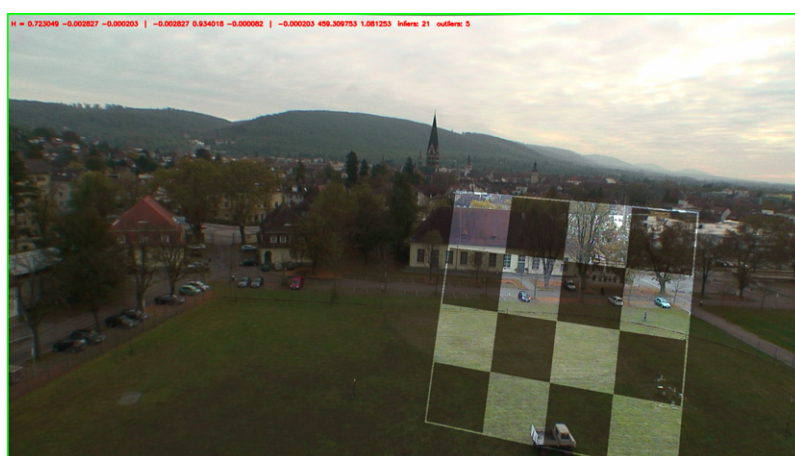


Abbildung 6.12: Blick vom Turm der Übersichtskamera mit eingefügtem Teilbild der Pan-Tilt-Zoom-Kamera.

Die Bewegungsspirale enthält 420 Motorkoordinaten, von denen 313 erfolgreich Bildkoordinaten im Übersichtsbild zugeordnet werden konnten. In Abbildung 6.12 ist eine solche Zuordnung optisch dargestellt, wie man sie auch während des Konfigurationsprozesses angezeigt bekommt. Alle weiteren Graphiken zum Experiment C wurden in der Abbildung 6.13 zusammengestellt. Durch lineare Regression mit den Schwellwerten 0,6 für Pan und 0,45 für Tilt und einem Mindestanteil an Inlier von 0,6 bei Pan und 0,7 bei Tilt wurden die in den Gleichungen 6.3 dargestellten Funktionen ermittelt.

$$\begin{aligned} pan = & -108,669281 + 0,045629 * x - 0,009613 * y \\ & + 0,000012 * xy + 0,000002 * x^2 - 0,000000 * y^2 \end{aligned} \quad (6.3a)$$

$$\begin{aligned} tilt = & 12,842436 - 0,005559 * x - 0,044867 * y \\ & - 0,000000 * xy + 0,000004 * x^2 + 0,000000 * y^2 \end{aligned} \quad (6.3b)$$

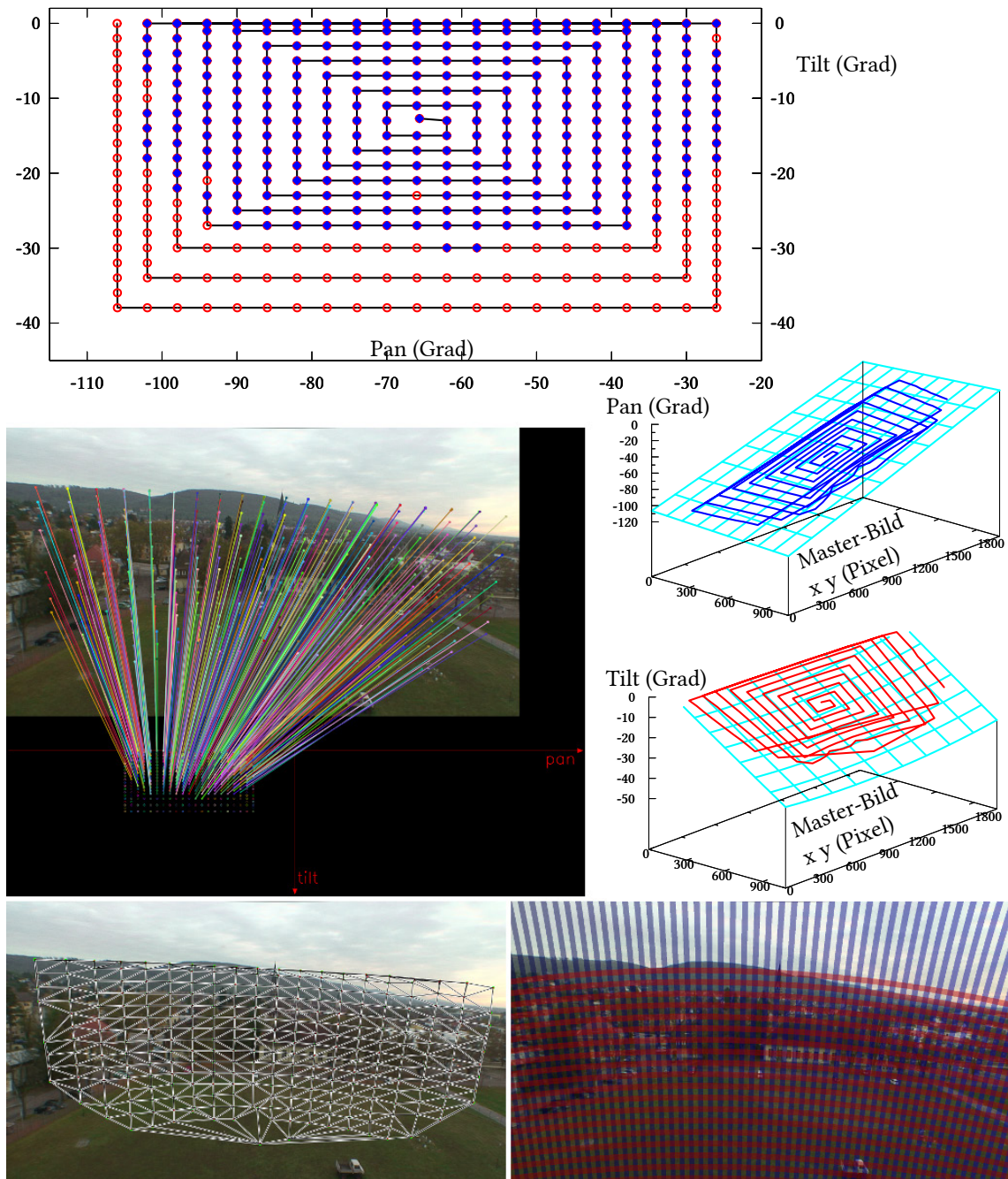


Abbildung 6.13: Ergebnisbilder vom Turm hinunter.

## 6.4 Experiment D: Stativ am Boden mit Blick nach Osten

Experiment D beinhaltet zwei Konfigurationsdurchläufe, die beide mit dem Stativ an der selben Stelle durchgeführt wurden. Hierzu wurde das Stativ am Boden hinter dem Institut, mit Blickrichtung längs zum Gebäude nach Osten, aufgestellt. Der erste Durchlauf wurde mit  $600 \times 600$  Pixel großen Bildausschnitten der Pan-Tilt-Zoom-Kamera durchgeführt. Der  $\delta$ -Wert betrug 2 und die sich daraus ergebende Spirale bestand aus 342 Motorkoordinaten. Von diesen konnten zu 175 Wertepaaren korrespondierende Koordinaten im Übersichtsbild ermittelt werden. Graphiken zum Experiment kann man Abbildung 6.16 entnehmen. Die lineare Regression wurde mit dem Schwellwert 0,4 für Pan sowie Tilt durchgeführt und hatte jeweils eine Inlierrate von 0.7 als Abbruchbedingung.

$$\begin{aligned} pan &= -109,974747 + 0,048123 * x - 0,011010 * y \\ &\quad + 0,000011 * xy - 0,000002 * x^2 + 0,000002 * y^2 \end{aligned} \quad (6.4a)$$

$$\begin{aligned} tilt &= 10,632298 - 0,002286 * x - 0,041041 * y \\ &\quad + 0,000002 * xy + 0,000002 * x^2 - 0,000008 * y^2 \end{aligned} \quad (6.4b)$$

Für *pan* konnte innerhalb dieser Parameter nach 3000 Iterationsschritten keine Funktion gefunden werden. Das Beste Verhältnis von 119 Inlier zu 56 Outlier ergab nur einen Wert von 0,68, was unter 0,7 liegt. Für *tilt* dagegen konnte bereits im ersten Schritt eine Funktion mit 144 Inlier zu 31 Outlier gefunden werden. Zur Bestimmung weiterer Inter-

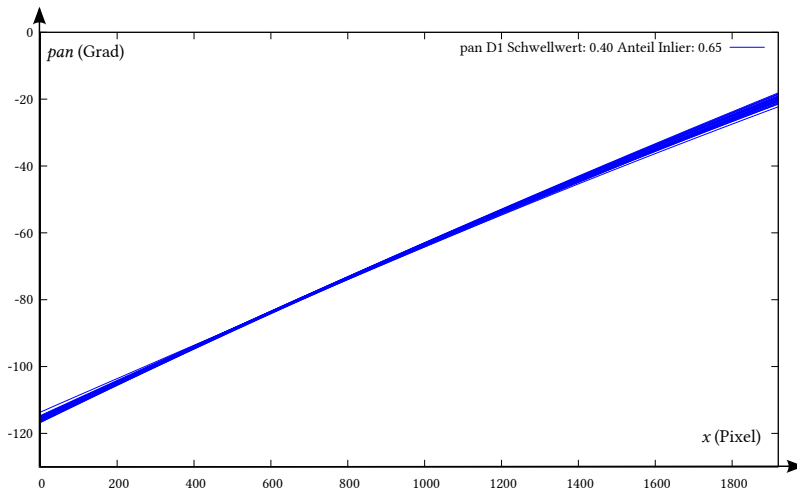


Abbildung 6.14: Pan-Werte gegen  $x$  bei  $y = 540$ , d.h. auf halber Bildhöhe, in Experiment D1 für 30 durch lineare Regression bestimmte Funktionen, jeweils mit Fehler-Schwellwert 0,4 und 0,65 Anteil Inlier.

polationsergebnisse wurde der nötige Anteil Inlier für *pan* auf 0,65 gesenkt und für *tilt* auf 0,8 angehoben. Je eine Funktion für *pan* und *tilt* mit diesen Werten sind in den Gleichungen 6.4 zu finden. Mit diesen Werten wurde 30 mal das Verfahren zur Bestimmung je einer Interpolationsfunktion für jede der beiden Motorkoordinaten durchgeführt. Da ein optischer Vergleich von dreidimensionalen Funktionen schwierig ist, wurde in Abbildung 6.14 der Schnitt mit der Ebene  $\{y = 540\}$ , welche die Bildkoordinaten vertikal halbiert, verwendet, um die erhaltenen Funktionen von 30 Anwendungen des Interpolationsverfahrens darzustellen. In Abbildung 6.15 wurde das selbe mit einem horizontalen Schnitt an  $\{x = 960\}$  für 30 bestimmte *tilt*-Funktionen durchgeführt. Man kann in dieser Abbildung erkennen dass die Tilt-Werte am unteren Bildrand stärker voneinander abweichen, was auf die geringe Dichte von Daten in dem Bereich zurückzuführen ist.

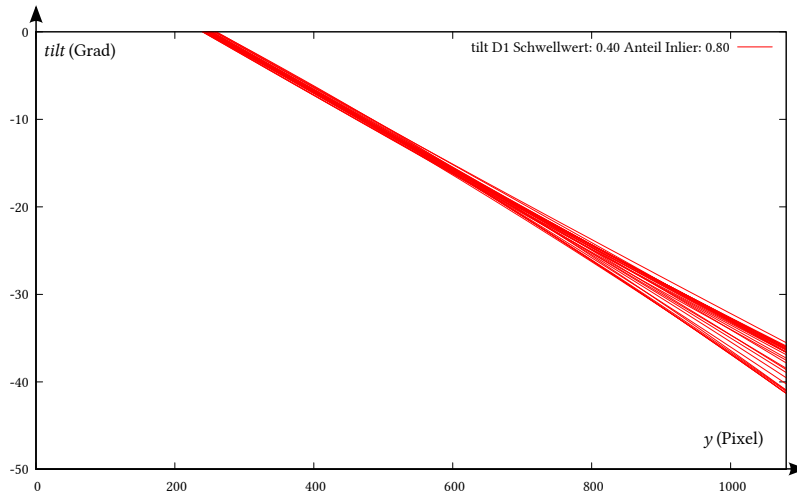


Abbildung 6.15: Tilt-Werte gegen  $y$  bei  $x = 960$ , d.h. auf halber Bildbreite, in Experiment D1 für 30 durch lineare Regression bestimmte Funktionen, jeweils mit Fehlerschwellwert 0,4 und 0,8 Anteil Inlier.

Mit genau dem selben Aufbau des Stativs wurde noch ein weiterer Konfigurationsdurchlauf durchgeführt. Bei diesem wurde ein größeres  $\delta = 6$  gewählt. Außerdem wurde das komplette Bild der Pan-Tilt-Zoom-Kamera verwendet und nicht wie zuvor ein zentriertes Teilbild. Daraus ergab sich eine Spirale von nur 66 Motorkoordinaten, von denen zu 32 Pan-Tilt-Werten zugehörige  $x$ - $y$ -Koordinaten im Master-Bild bestimmt werden konnten. Einige Graphiken zum Experiment sind in Abbildung 6.18 zusammengefasst. Die Parameter für die lineare Regression waren 0,4 für den Fehlerschwellwert und 0,7 für den erforderlichen Anteil Inlier. Für *pan* konnte die Funktion 6.5a nach 60 Iterationsschritten, für *tilt* die Funktion 6.5b nach 2 Iterationsschritten bestimmt werden. Auch hier wurden zum Vergleich untereinander und zum vorherigen Experiment je 30 Funk-



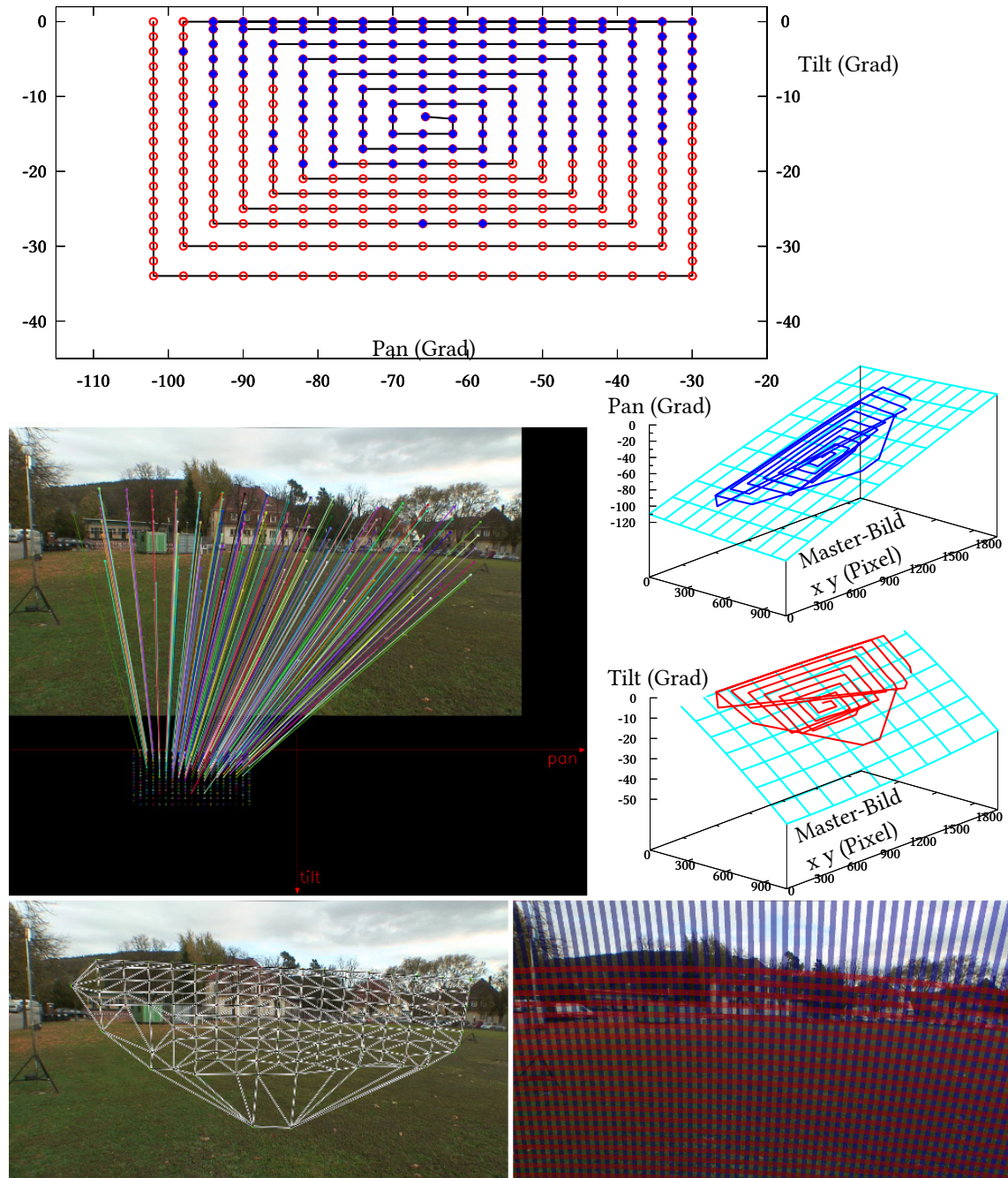


Abbildung 6.16: Ergebnisbilder auf der Wiese in Richtung Pavillon mit kleiner Schrittweite.

tionen für  $pan$  und  $tilt$  bestimmt. Diese sind in Abbildung 6.17 zusätzlich zu jenen in den Abbildungen 6.14 und 6.15 dargestellt. Man kann bei dem Versuch mit weniger Werte-Quartupeln eine breitere Streuung der Funktionen feststellen. Davon sind allerdings die Randbereiche deutlicher betroffen.

$$pan = -116,233200 + 0,057497 * x - 0,006511 * y + 0,000008 * xy - 0,000006 * x^2 + 0,000000 * y^2 \quad (6.5a)$$

$$tilt = 11,622519 - 0,000355 * x - 0,050932 * y + 0,000003 * xy + 0,000001 * x^2 + 0,000004 * y^2 \quad (6.5b)$$

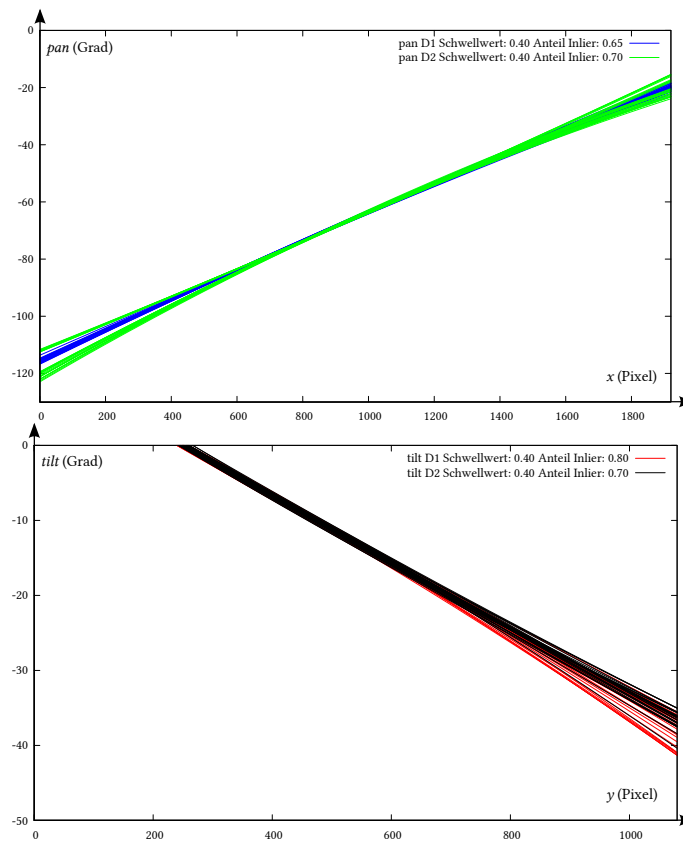


Abbildung 6.17: Pan-Werte gegen  $x$  bei  $y = 540$  und Tilt-Werte gegen  $y$  bei  $x = 960$  von jeweils 30 durch lineare Regression der selben Daten bestimmten Interpolationsfunktionen. Blau/Rot für Experiment D1 mit 175 Daten-Quartupeln, Grün/Schwarz für D2 mit 32 Daten-Quartupeln.

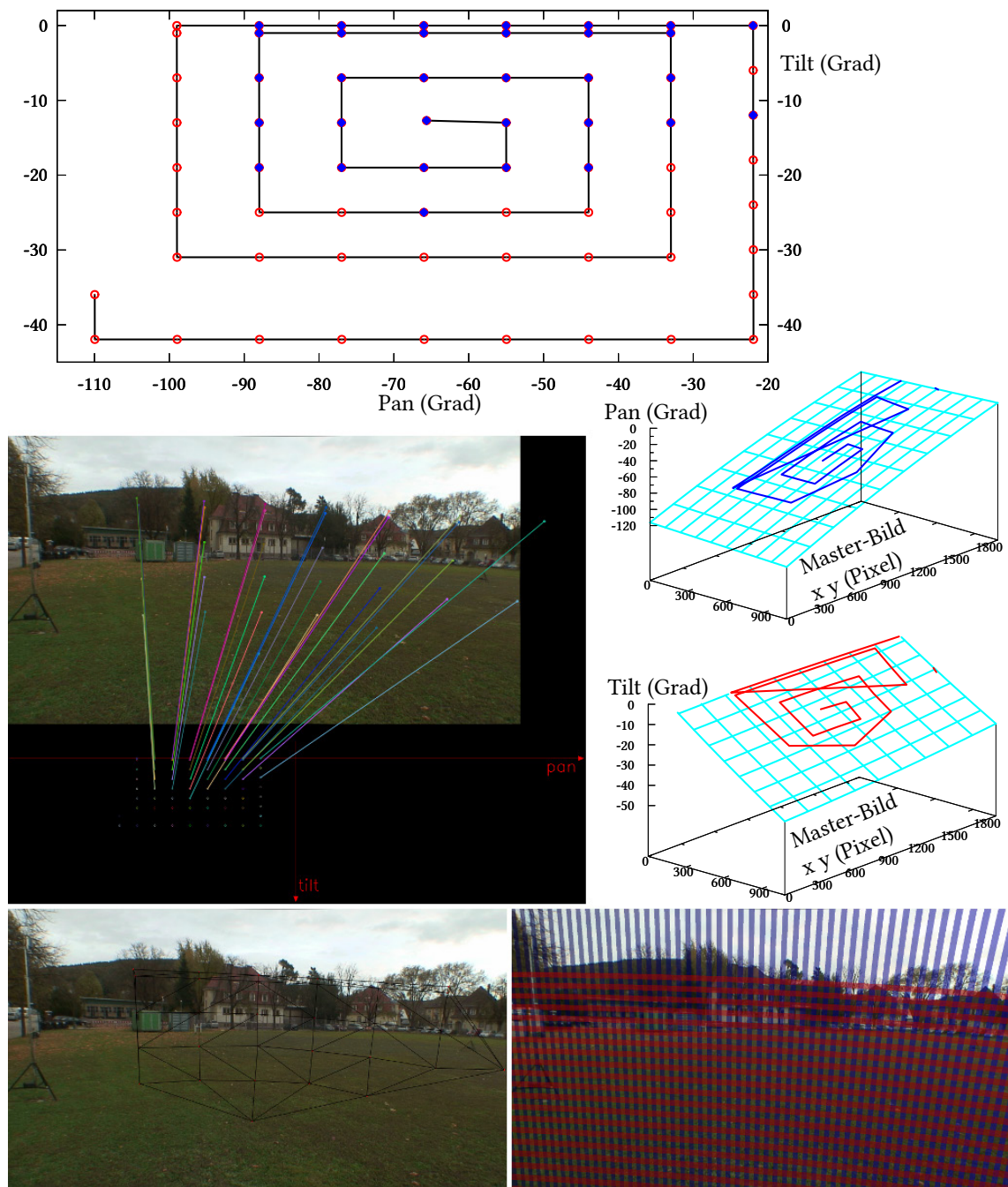


Abbildung 6.18: Ergebnisbilder auf der Wiese in Richtung Pavillon mit großer Schrittweite.

## 6.5 Experiment E: Außenkameras des Kameranetzwerks am Eingang

Zur Durchführung von Experiment E konnten ein Teil des stationären Kameranetzwerkes am Fraunhofer Institut verwendet werden. Dieses enthält eine Kameragruppe, die aus drei nebeneinander montierten statischen Kameras vom Typ Axis Q1755 und eine darunter angebrachten aktiven Kamera vom Typ Axis P5534 besteht. Neben den Kameras sind noch drei Infrarot-Strahler montiert. Von den statischen Kameras wurde diejenige gewählt, die auf den Institutseingang gerichtet ist und zusammen mit der aktiven Kamera verwendet. Aus Gründen des Arbeitsschutzes sind die Kameras so konfiguriert, dass alle Bereiche in denen sich Bürofenster befinden permanent geschwärzt werden. Dies erschwert die Verwendung dieser Kameras ungemein. Es konnte dennoch eine Konfiguration durchgeführt werden und das Ergebnis zur Objektverfolgung, siehe Abbildung 6.19, erfolgreich angewandt.



Abbildung 6.19: Zu Experiment E: Bild aus der automatischen Objektverfolgung mit den Kameras am Institutseingang. Die zugehörigen Aufnahmen wurden bei Dämmerung mit aktivierten Infrarotstrahlern getätigt.

Der Konfigurationsdurchlauf wurde mit leicht reingezoomter Pan-Tilt-Zoom-Kamera und Schrittweite  $\delta = 3$  durchgeführt. Es konnten an 18 von 41 Motorkoordinaten Zuordnungen bestimmt werden, für die selbst mit Fehlerschwellwert 0,1 die Funktionen 6.6 für *pan* und *tilt* ohne Outlier in ein bis zwei Iterationsschritten ermittelt werden konnten.



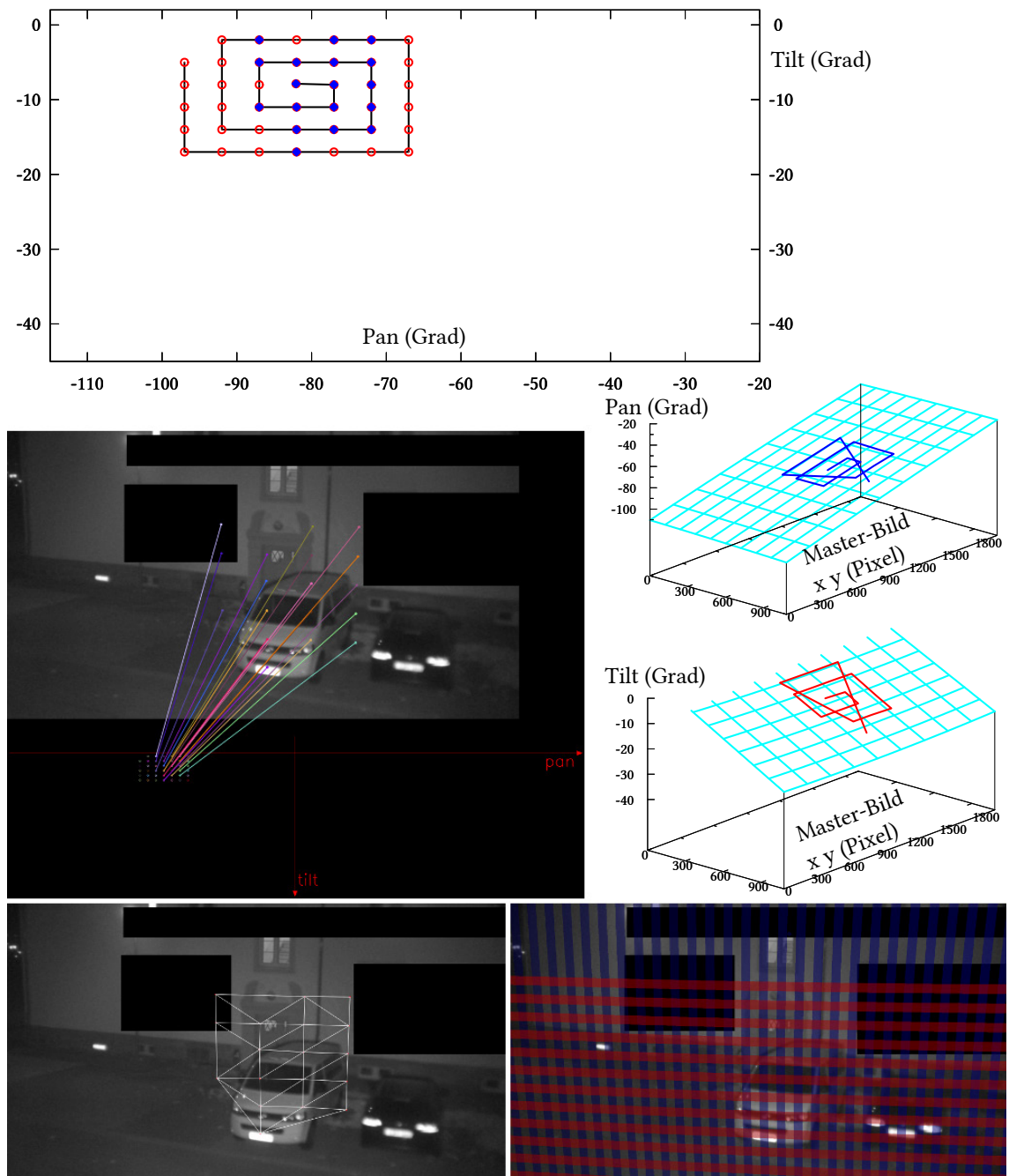


Abbildung 6.20: Ergebnisbilder am Institutseingang.

$$\begin{aligned} pan &= -110,359566 + 0,029822 * x - 0,002170 * y \\ &\quad + 0,000003 * xy - 0,000001 * x^2 - 0,000001 * y^2 \end{aligned} \quad (6.6a)$$

$$\begin{aligned} tilt &= 10,238241 - 0,003767 * x - 0,029601 * y \\ &\quad + 0,000002 * xy + 0,000001 * x^2 - 0,000000 * y^2 \end{aligned} \quad (6.6b)$$

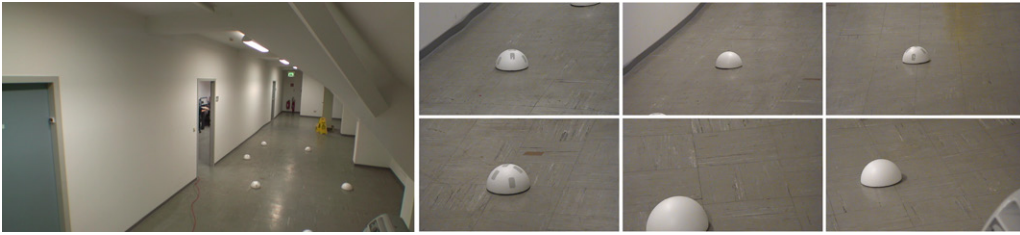


Abbildung 6.21: In Experiment F zur Evaluation platzierte Markierungen im Master-Bild (links) und nach jeweils automatischem fokussieren anhand der durchgeführten Konfiguration im Bild der Pan-Tilt-Zoom-Kamera (rechts).

## 6.6 Experiment F: Stativ im Flur des Gebäudes

Für Experiment F wurde das Stativ in einem der Institutsflure aufgestellt. Die kahlen Wände und der Boden lieferten nur wenige lokale Bildmerkmale. Die Konfiguration wurde mit  $\delta = 4$  und  $600 \times 600$  Pixel großen Teilbildern der Pan-Tilt-Zoom-Kamera durchgeführt. Obwohl nur 8 vom 72 Motorkoordinaten korrespondierenden Bildkoordinaten zugeordnet werden konnten kamen durch die lineare Regression brauchbare Funktionen zustande, siehe Gleichungen 6.7.

$$\begin{aligned} pan &= -112,925377 + 0,053692 * x - 0,017178 * y \\ &\quad + 0,000012 * xy - 0,000004 * x^2 + 0,000008 * y^2 \end{aligned} \quad (6.7a)$$

$$\begin{aligned} tilt &= 18,872549 - 0,007746 * x - 0,063509 * y \\ &\quad + 0,000007 * xy + 0,000003 * x^2 + 0,000011 * y^2 \end{aligned} \quad (6.7b)$$

Zur Evaluation wurden sechs Markierungen zufällig platziert und mit der Pan-Tilt-Zoom-Kamera automatisch angefahren. Das Ergebnis ist in Abbildung 6.21 zu sehen.

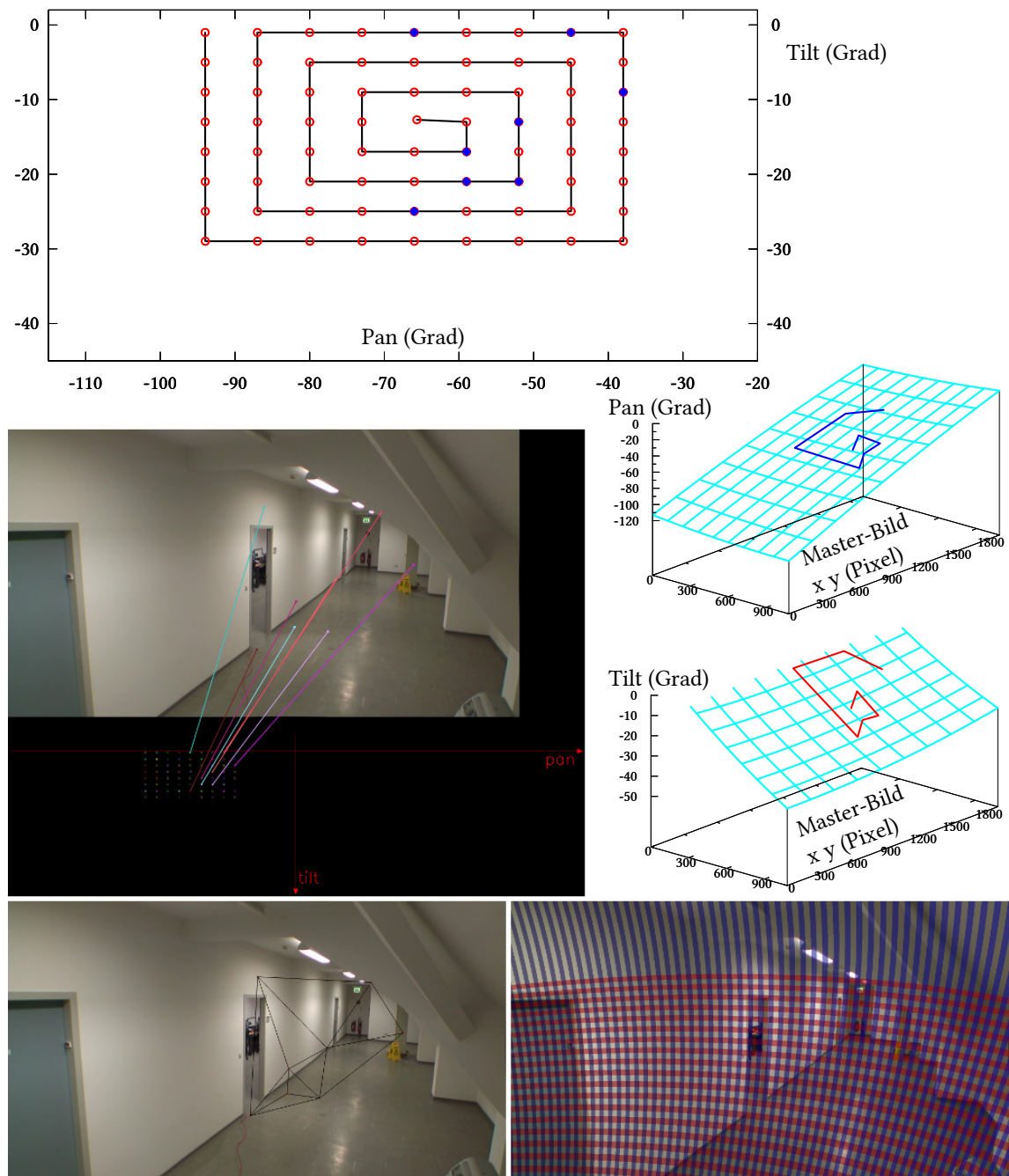


Abbildung 6.22: Ergebnisbilder im Flur.



# Kapitel 7

## Diskussion und Ausblick

Es wurde ein Verfahren zur automatischen Konfiguration des bereitgestellten Mehrkamerasystems entwickelt und implementiert. Das System ist danach in der Lage einen in Form von Koordinaten im Übersichtsbild angegebenen Bereich automatisch mit der aktiven Kamera anzufahren. Die benötigte Angabe der Koordinaten geschieht durch einen Mausklick im Bild oder durch die Lageinformation, die ein Blobdetektor liefert. Die Konfiguration wurde in unterschiedlichen Diskursbereichen durchgeführt und für verschiedene Eingabeparameter evaluiert.

Das Verfahren funktioniert gut für hinreichend nahe beieinander montierte Kameras, wie es bei dem zur Verfügung gestellten System der Fall ist, wenn in den Sichtbereichen genügend lokale Bildmerkmale vorhanden sind. Für karge Bildbereiche wie homogene Bodenflächen oder glatte Wände ist dies nicht gegeben. Hierfür bietet es sich an, ein Lernverfahren zur Onlineverbesserung noch hinzuzufügen, welches, sobald sich bewegliche Objekte in bisher spärlich erfassten Bereichen befinden, diese zur Verbesserung der Konfiguration verwendet. Da sich Kamerasysteme durch permanente Benutzung der Mechanik dejustieren können (Wu und Radke, 2012), ist eine Onlineverbesserung überall und nicht nur für bisher lückenhaft erfasste Bereiche sinnvoll.

Nach der Konfiguration lässt sich momentan das Ergebnis noch durch manuelles Einstellen der optimalen Parameter für die beiden RANSAC-Schwellwerte in der linearen Regression verbessern. Ein automatisches Ausloten der optimalen Parameter an dieser Stelle sollte noch ergänzt werden.

Das aktuelle Verfahren beinhaltet noch nicht die Bestimmung von passenden Zoom-Werten, sondern verwendet bei der Verfolgung der Objekte einen festen Zoom-Wert von 3000. Hier mangelt es noch an einer Zuordnung von Größe und Lage eines Rechtecks im Übersichtsbild zum passenden Zoomfaktor der aktiven Kamera. Eine solche Zuordnung könnte mit der Bestimmung der Größe und Lage des korrespondierenden Rechtecks im Bild der Pan-Tilt-Zoom-Kamera als Zwischenschritt realisiert werden. Der zweite Schritt vom Rechteck im Bild der Pan-Tilt-Zoom-Kamera zum passenden Zoom wäre bereits im-

plementiert, da das System zu einem per Maus gezogenen Rechteck im Pan-Tilt-Zoom-Kamera-Bild in der Lage ist, den passenden Zoom-Wert einzustellen, um das Rechteck formatfüllend ins Bild zu holen.

Der momentan verwendete einfache Blobdetektor ist zur Objektverfolgung nicht ausreichend, da bei mehreren Blobs gleichzeitig im Bild, insbesondere wenn der aktuell verfolgte Blob einen kurzen Aussetzer hat, die Pan-Tilt-Zoom-Kamera so schnell hin und her springt, dass kein scharfes Bild zustande kommt. Dies war auch nicht Teil dieser Arbeit. Aktuell wird an einer netzwerkbasierten Anbindung von verschiedenen Trackern gearbeitet. Dies ist notwendig, weil ein Rechner alleine, aktuell noch nicht die nötige Rechenleistung aufbringen kann, die das *CameraSystem* und zusätzlich das Trackingverfahren benötigen.

Ein System zur inhaltlichen Videoanalyse kann man hierarchisch beschreiben, wie z.B. das 'Cognitive System' in (M. Arens, 2004; Nagel, 2004). Bei diesen Arbeiten wird eine Möglichkeit aufgezeigt, dass und wie semantische Information die darunterliegenden Schichten beeinflussen kann. Das Werkzeug zur Situationsanalyse, der SGT-Editor (M. Arens, 2004), kann auf erkannte Situationen reagieren, was auch in (Bellotto u. a., 2012) gezeigt wurde. Mit dem hier vorgestellten Master-Slave System kann nun gezielt mehr Information über einen bestimmten lokalen Bildausschnitt erfasst werden. Damit können sowohl völlig neue Situationen erkannt werden, die bisher mangels Detailinformation nicht analysiert werden konnten, als auch die bisherige Situationsanalyse robuster gemacht werden. Von der Erweiterung, hier am Fraunhofer IOSB entstandener Arbeiten zur Situationsanalyse (Münch u. a., 2012) um dieses Master-Slave System, verspricht man sich eine völlig neue Detailtiefe der inhaltlichen Videoanalyse.

# Anhang A

## Zeitplan

Zeitraum	Aufgabe
3 Wochen	Literaturrecherche.
4 Wochen	Einarbeitung QT und OpenCV.
4 Wochen	Erweiterung des Kamera Systems für 3 Kameras.
6 Wochen	Umsetzung SURF, NNDR, Hintergrund-Vordergrund Segmentierung, Blobdetektor.
6 Wochen	Experimente und Evaluation.
3 Wochen	Schreiben der Arbeit und Vorbereitung der Präsentation.





**Anhang B**

**Datenblätter**

www.axis.com

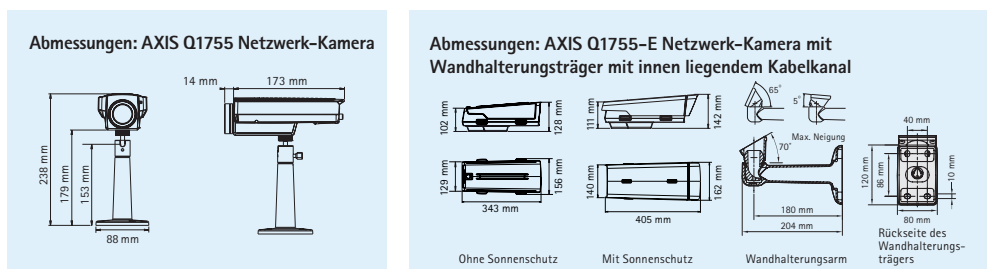
42730/DE/R4/1210

## Technische Daten – AXIS Q1755/-E Netzwerk-Kameras

Kamera		Systemintegration	
Modelle:	AXIS Q1755 60 Hz; AXIS Q1755 50 Hz	Anwendungs- programmier- schnittstelle	AXIS Camera Application Platform von Axis Communications Spezifikationen sind auf <a href="http://www.axis.com">www.axis.com</a> verfügbar Enthält die ONVIF Spezifikation, verfügbar auf <a href="http://www.onvif.org">www.onvif.org</a> AXIS Video Hosting System (AVHS) durch "One Click" Kamera Verbindung
Modell:	AXIS Q1755-E 60 Hz; AXIS Q1755-E 50 Hz	Intelligentes Video	Videobewegungserkennung, aktiver Manipulationsalarm, Audioerkennung, Gatekeeper, AXIS Camera Application Platform
Außenbereich		Alarmauslösung	Videobewegungserkennung, aktiver Manipulationsalarm, Audioerkennung, Temperatur, externe Signale, AXIS Camera Application Platform
Bildsensor	1/3" CMOS, progressive Abtastung, 2 Megapixel	Alarmereignisse	Datei-Upload über FTP, HTTP und E-Mail; Benachrichtigung per E-Mail, HTTP und TCP; Aktivierung externer Ausgänge
Objektiv	f=5,1 – 51 mm, F1,8 – 2,1, Autofokus, Tag/Nacht-Automatik Horizontaler Sichtwinkel: 48,1° – 5,1° M37 x 0,75-Anschlussgewinde für den optionalen Objektivadapter	Videopuffer	96 MB Vor- und Nachalarm
Minimale Lichtstärke	Farbe: 2 Lux bei 30 IRE, F1,8 Schwarzweiß: 0,2 Lux bei 30 IRE, F1,8	Allgemein	
Verschlusszeit	1/10000 s bis 1/2 s	Gehäuse	Kamera: Metall (Zink) AXIS Q1755-E: IP66 und NEMA 4X, IK10 schlagfestes Aluminiumgehäuse
Zoom	10-fach optisch und 12-fach digital, insgesamt 120-fach	Prozessor und Speicher	ARTPEC-3, 256 MB RAM, 128 MB Flash
Video		Stromversorgung	AXIS Q1755: 8–20 V Gleichstrom, max. 11,2 W ; 20–24 V Wechselstrom, max. 17,4 VA; Power over Ethernet (IEEE 802.3af) Klasse 3 AXIS Q1755-E: Power over Ethernet (PoE) IEEE 802.3af Klasse 3 oder High Power over Ethernet (High PoE)
Video- komprimierung	H.264 (MPEG-4 Teil 10/AVC) Motion JPEG	Anschlüsse	RJ-45 10BASE-T/100BASE-TX PoE, Anschlussblock für Netzspannung E/A-Anschlussblock für zwei konfigurierbare Eingänge/Ausgänge 3,5 mm Mikro-/Line-Eingang, 3,5 mm Line-Ausgang RS-422/RS-485 Video-Ausgang: 3x RCA Y/Pb/Pr (HD), 1x RCA Composite (SD)
Auflösungen	HDTV 1080i 1920x1080, HDTV 720p 1280x720 Standard-NTSC/PAL für die Installation	Lokaler Speicher	SD/SDHC-Speicherkartensteckplatz (Speicherkarte nicht im Lieferumfang enthalten)
Bildrate H.264/ Motion JPEG	30/25 Bilder/s bei allen Auflösungen	Betriebs- bedingungen	AXIS Q1755: 0 °C bis 45 °C, relative Luftfeuchtigkeit 20–80 % (nicht kondensierend) AXIS Q1755-E: -30 °C bis 45 °C mit PoE, bis zu -40 °C mit High PoE Arctic Temperature Control ermöglicht die Inbetriebnahme der Kamera bei Temperaturen bis -40 °C über High PoE
Video-Streaming	Mehrere einzeln konfigurierbare H.264- und Motion JPEG-Videoströme Bildrate und Bandbreite steuerbar VBR/CBR H.264	Zulassungen	EN 55022 Klasse B, EN 55024, EN 61000-6-1, EN 61000-6-2, EN 60950-1, FCC Teil 15, Abschnitt B, Klasse B, VCCI, Klasse B ITE, C-tick AS/NZS CISPR 22, ICES-003, Klasse B AXIS Q1755: EN 61000-3-2, EN 61000-3-3 AXIS Q1755-E: IEC 60529 IP66, NEMA 250 Typ 4X, 60950-22, IEC 62262 IK10, IEC 60068-2-6, IEC 60068-2-27
Bildeinstellungen	Komprimierung, Helligkeit, Schärfe, Weißabgleich, Belichtungssteuerung, Gegenlichtausgleich, Rotation – beinhaltet Corridor Format™, Spiegelung von Bildern, Text- und Bild-Overlay, Privatzonenmaske	Gewicht	AXIS Q1755: 1,0 kg; AXIS Q1755-E: 3,5 kg
Audio		Im Lieferumfang enthaltene Zubehör	Montage- und Anschlusskits, Installationsanleitung, CD mit Installationsprogrammen, Aufzeichnungssoftware und Benutzerhandbuch, Windows Decoder-Lizenz für 1 Benutzer AXIS Q1755-E: Wandhalterungsträger, Sonnenschutz, Kabelverschraubung
Audio-Streaming	Zweigeige-Audio	Optionales Zubehör	Wandhalterungszubehör, AXIS PoE Midspan (1 Anschluss), AXIS 18123 High PoE Midspan (1 Anschluss), AXIS T90A Strahler, YP3040 Schwenk-Neige-Motor, AXIS T8414 Installationsmonitor, AXIS Camera Station und Videoverwaltungssoftware von Axis' Application Development Partners
Audio- komprimierung	AAC LC 8/16 kHz, G.711 PCM 8 kHz, G.726 ADPCM 8 kHz Konfigurierbare Bitrate		
Audio- Eingang-/Ausgang	Integriertes Mikrofon, externes Mikrofon oder Audio-Eingang, Audio- Ausgang		
Netzwerk			
Sicherheit	Kennwortschutz, IP-Adressfilter, HTTPS-Verschlüsselung, Netzwerkzugriffskontrolle nach IEEE 802.1X*, Digest-Authentifizierung, Benutzerzugriffsprotokollierung		
Unterstützte Protokolle	IPv4/IPv6, HTTP, HTTPS*, QoS Layer 3 DiffServ, FTP, SMTP, Bonjour, UPnP, SNMPv1/v2c/v3(MIB-II), DNS, DynDNS, NTP, RTSP, RTP, TCP, UDP, IGMP, RTP, ICMP, DHCP, ARP, SOCKS. Unterstützt eine große Auswahl an Schwenkneigeköpfen (Treiber sind zum downloaden auf <a href="http://www.axis.com">www.axis.com</a> verfügbar)		

\* Dieses Produkt enthält Software, die vom OpenSSL Project zur Verwendung im  
OpenSSL Toolkit entwickelt wurde. ([www.openssl.org](http://www.openssl.org))

Weitere Informationen finden Sie unter [www.axis.com](http://www.axis.com)



©2012 Axis Communications AB. AXIS COMMUNICATIONS, AXIS, ETRAX, ARTPEC und VAPIX sind eingetragene sowie in verschiedenen Gerichtsbezirken beantragte  
Marken von Axis. Alle weiteren Firmen und Produktnamen sind Marken oder eingetragene Marken der jeweiligen Firmen. Wir behalten uns das Recht vor,  
Änderungen ohne vorherige Ankündigungen vorzunehmen.



Abbildung B.1: Aus dem Datenblatt der Übersichtskamera Axis Q1755.

www.axis.com

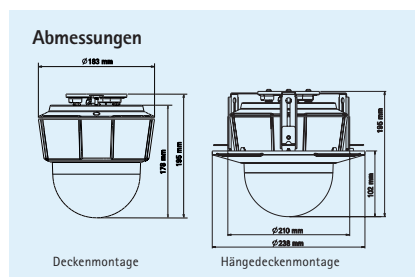
## Technische Daten – AXIS P5534 PTZ Dome Netzwerk-Kamera

40788(DE/RS)/210

Kamera		Systemintegration	
<b>Modelle</b>	AXIS P5534 60 Hz, AXIS P5534 50 Hz	<b>Anwendungs-programmier-schnittstelle</b>	Offene API für Softwareintegration, einschließlich VAPIX® von Axis Communications, verfügbar unter <a href="http://www.axis.com">www.axis.com</a> . Unterstützt das AXIS Video Hosting System (AVHS), das den Kameraanschluss mit einem Mausklick ermöglicht
<b>Bildsensor</b>	1/3" CCD, 1,3 Megapixel, progressive Abtastung	<b>Intelligentes Video</b>	Videobewegungserkennung, Audioerkennung, Advanced Gatekeeper-Funktion AXIS Camera Application Platform, was die Installation zusätzlicher Anwendungen ermöglicht
<b>Objektiv</b>	f=4,7 – 84,6 mm, F1,6 – 2,8, Autofokus, Tag/Nacht-Automatik, horizontaler Sichtwinkel: 55,2° – 3,2°	<b>Alarmauslösung</b>	Videobasierte Bewegungserkennung, Audioerkennung, externer Eingang, PTZ-Position, Temperatur
<b>Minimale Lichtstärke</b>	Farbe: 0,74 Lux bei 30 IRE F1,6 Schwarzweiß: 0,04 Lux bei 30 IRE F1,6	<b>Alarmereignisse</b>	Hochladen von Dateien über FTP, HTTP und E-Mail; Benachrichtigung über E-Mail, HTTP und TCP; Aktivierung externer Ausgänge; Aufzeichnung von Video und Audio auf Edge-Speichergeräten; Videopuffer zum Speichern von Vor- und Nachalarmbildern; Abspielen von Audiclips
<b>Verschlusszeit</b>	1/10 000 s – 1/4 s	<b>Daten-Streaming</b>	Ereignisdaten
<b>Schwenken/Neigen/Zoomen</b>	E-Flip, Auto-Flip, 100 Positionierungsvoreinstellungen Schwenken: 360° (mit Auto-Flip), 0,2° – 300°/s Neigen: 180°, 0,2° – 300°/s 18-fach optischer und 12-fach digitaler Zoom, 216-fach insgesamt	<b>Installationshilfen</b>	Pixelzähler
<b>Schwenk-/Neige-/Zoom-Funktionalität</b>	Limitierte Rundgangsüberwachung Kontrollreihe Richtungsanzeige auf dem Bildschirm	Allgemein	
<b>Video</b>		<b>Gehäuse</b>	Schutz nach IP5, Metallgehäuse (Aluminium), klare Kuppel aus Acrylglas (PMMA)
<b>Video-komprimierung</b>	H.264 (MPEG-4 Part 10/AVC) Motion JPEG	<b>Speicher</b>	256 MB RAM, 128 MB Flash
<b>Auflösungen</b>	HDTV 720p 1280x720 bis 320x180	<b>Stromversorgung</b>	Power over Ethernet Plus (PoE+) IEEE 802.3at, max. 30 W 20 – 24 V Wechselstrom 24 – 34 V Gleichstrom, max. 17 W AXIS T8123 High PoE Midspan (1 Anschluss) enthalten: 100 – 240 V Wechselstrom
<b>Bildrate</b>	H.264: Bis zu 30/25 (60/50 Hz) Bilder/s in allen Auflösungen Motion JPEG: Bis zu 30/25 (60/50 Hz) Bilder/s in allen Auflösungen	<b>Anschlüsse</b>	RJ-45 für 10BASE-T/100BASE-TX PoE Mehrfachanschluss (Kabel nicht enthalten) für Wechsel- oder Gleichstromversorgung, 4 konfigurierbare Alarmeingänge/-ausgänge, Mikrofon-Eingang, Mono-Eingang, Mono-Ausgang für Aktivlautsprecher
<b>Video-Streaming</b>	Mehrere einzeln konfigurierbare H.264- und Motion JPEG-Videostreams Bildrate und Bandbreite steuerbar VBR/CBR H.264	<b>Edge Storage</b>	SD/SDHC-Speicherkartenschacht (Karte nicht im Lieferumfang enthalten) Unterstützt Aufzeichnen direkt auf eine Netzwerkfreigabe, (Network attached Storage oder File-Server)
<b>Bildeinstellungen</b>	Großer Dynamikbereich, Manuelle Verschlusszeiteinstellung, Komprimierung, Farbe, Helligkeit, Schärfe, Weißabgleich, Belichtungssteuerung, Belichtungszonen, Gegenlichtausgleich, Feineinstellung des Verhaltens bei schwachem Licht, Drehung, Text- und Bild-Overlay, 3D Privatzonenmaske, Standbildfunktion bei PTZ	<b>Betriebsbedingungen</b>	0 °C bis 50 °C Relative Luftfeuchtigkeit 15 bis 85 % (nicht kondensierend)
<b>Audio</b>		<b>Zulassungen</b>	EN 55022 Klasse B, EN 61000-3-2, EN 61000-3-3, EN 61000-6-1, EN 61000-6-2, EN 55024, FCC Teil 15 Abschnitt B Klasse B, ICES-003 Klasse B, VCCI Klasse B, C-Tick AS/NZS CISPR 22, EN 60950-1, KCC Klasse B IEC 60721-4-3 Klasse 3K3, 3M3, EN/IEC 60068-2 Midspan: EN 60950-1, GS, UL, cUL, CE, VCCI, CB, CCC
<b>Audio-Streaming</b>	Zwei-Wege	<b>Gewicht</b>	Kamera: 1,9 kg Kamera plus Deckenhalterung: 2,3 kg
<b>Audio-komprimierung</b>	AAC-LC 8/16 kHz, G.711 PCM 8 kHz, G.726 ADPCM 8 kHz Konfigurierbare Bitrate	<b>Im Lieferumfang enthaltenes Zubehör</b>	AXIS T8123 High PoE Midspan (1 Anschluss), Montagesatz für die Befestigung an Decken und in Hängedecken, Rauchgaskuppelabdeckung, Installationsanleitung, CD mit Installationsprogrammen und Aufzeichnungssoftware, Windows-Decoder-Lizenz für einen Benutzer
<b>Eingang/Ausgang</b>	Eingang für externes Mikrofon oder Line-Eingang, Line-Ausgang - Mehrfachanschlusskabel erforderlich (separat erhältlich)		
<b>Netzwerk</b>			
<b>Sicherheit</b>	Kennwortschutz, IP-Adressfilter, HTTPS*-Verschlüsselung, Netzwerkzugriffskontrolle nach IEEE 802.1X*, Digest-Authentifizierung, Benutzerzugriffsprotokollierung		
<b>Unterstützte Protokolle</b>	IPv4/v6, HTTP, HTTPS*, SSL/TLS*, QoS Layer 3 DiffServ, FTP, CIFS/SMB, SMTP, Bonjour, UPnP*, SNMPv1/v2c/v3 (MIB-II), DNS, DynDNS, NTP, RTP, TCP, UDP, IGMP, RTCP, ICMP, DHCP, ARP, SOCKS		

\*Dieses Produkt enthält Software, die vom OpenSSL Project zur Verwendung im OpenSSL Toolkit entwickelt wurde. ([www.openssl.org](http://www.openssl.org))

Weitere Informationen finden Sie unter [www.axis.com](http://www.axis.com)



## Optionales Zubehör

AXIS T91A Montagezubehör



Mehrfachanschlusskabel zum Anschluss von EA- und Audiogeräten sowie für eine Wechsel- oder Gleichstromversorgung

AXIS T90A-Strahler



AXIS T8310 Videoüberwachungs-Steuerungseinheit



AXIS Camera Companion (im Lieferumfang),  
AXIS Camera Station und Video Management  
Software von Axis' Application Development  
Partners (nicht im Lieferumfang).  
Mehr Informationen unter:  
[www.axis.com/products/video/software/](http://www.axis.com/products/video/software/)

©2012 Axis Communications AB. AXIS COMMUNICATIONS, AXIS, ETRAX, ARTPEC und VAPIX sind eingetragene sowie in verschiedenen Gerichtsbezirken beantragte Marken von Axis. Alle weiteren Firmen und Produktnamen sind Marken oder eingetragene Marken der jeweiligen Firmen. Wir behalten uns das Recht vor, Änderungen ohne vorherige Ankündigungen vorzunehmen.

**AXIS**  
COMMUNICATIONS

Abbildung B.2: Aus dem Datenblatt der Pan-Tilt-Zoom-Kamera Axis P5534.

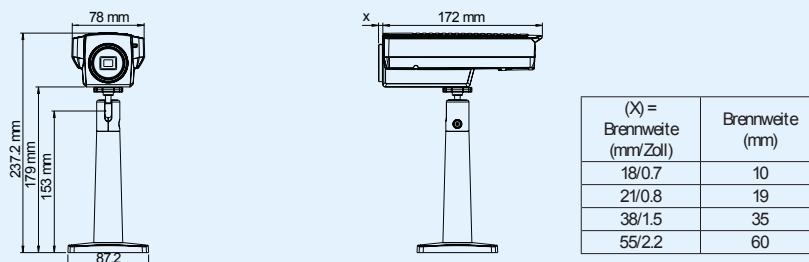
## Technische Daten – AXIS Q1922/- E Wärmebild- Netzwerk- Kameras

Kamera	
Modelle	Innenbereich: AXIS Q1922, 10 mm und 19 mm Außenbereich: AXIS Q1922-E, 10 mm, 19 mm, 35 mm und 60 mm
Bildsensor	Ungekühlter Mikrobolometer 640 x 480, Pixelgröße: 17 µm
Empfindlichkeit	NETD < 100 mK
Video	
Video-komprimierung	H.264 (MPEG-4 Teil 10/AVC) Motion JPEG
Auflösungen	Sensor: 640 x 480. Bild kann bis auf 800 x 600 (DI) skaliert werden
Standard-bildrate	Bis zu 30 Bilder pro Sekunde in Europa, Norwegen, Schweiz, Kanada, USA, Japan, Australien, Neuseeland Bis zu 8,3 Bilder pro Sekunde in anderen Ländern *Eine Bildrate von mehr als 9 Bildern pro Sekunde unterliegt möglicherweise Exportbeschränkungen
Video-Streaming	Mindestens drei H.264- und Motion JPEG-Videoströme mit derselben Palette, simultan oder einzeln konfiguriert für eine Auflösung von max. 30 Bilder pro Sekunde. Bildrate und Bandbreite steuerbar. VBR/CBR/H.264
Bildeinstellungen	Komprimierung, Helligkeit, Belichtungssteuerung, Drehung, Spiegelung von Bildern, Text- und Bild-Overlay, Privatzenenmaske, Farbpalette für Falschfarbendarstellung
Audio	
Audio-Streaming	Zweiwege, Voll duplex
Audio-komprimierung	AAC-LC 8/16 kHz, G.711 PCM 8 kHz, G.726 ADPCM 8 kHz Konfigurierbare Bitrate
Audio-Eingang/-Ausgang	AXIS Q1922: Integriertes Mikrofon, externes Mikrofon oder Audio-Eingang, Audio-Ausgang AXIS Q1922-E: Eingang für externes Mikrofon bzw. Audio-Eingang, Audio-Ausgang
Netzwerk	
Sicherheit	Kennwortschutz, IP-Adressfilter, HTTPS*-Verschlüsselung, Netzwerkzugriffskontrolle nach IEEE 802.1X*, Digest-Authentifizierung, Benutzerzugriffsprotokollierung
Unterstützte Protokolle	IPv4/v6, HTTP, HTTPS*, SSL/TLS*, CoS Layer 3 DiffServ, FTP, SMTP, Bonjour, UPnP, SNMPv1/v2c/v3 (MIB-II), DNS, DynDNS, NTP, RTSP, RTP, TCP, UDP, IGMP, RTCP, ICMP, DHCP, ARP, SOCKS Unterstützung für eine breite Palette an Schwenk-/Neigeköpfen (Treiber stehen unter <a href="http://www.axis.com">www.axis.com</a> zum Download bereit).
Systemintegration	
API (Anwendungs-programmierschnittstelle)	Offene API für Softwareintegration, einschließlich der ONVIF-Spezifikation, die unter <a href="http://www.onvif.org">www.onvif.org</a> verfügbar ist, sowie VAPIX® und AXIS Camera Application Platform von Axis Communications, Spezifikationen verfügbar unter <a href="http://www.axis.com">www.axis.com</a> Unterstützt das AXIS Video Hosting System (AVHS), das den Kameraanschluss mit einem Mausklick ermöglicht
Intelligentes Video	Videobewegungserkennung, aktiver Manipulationsalarm, Audioerkennung, Unterstützung für AXIS Camera Application Platform ermöglicht die Installation zusätzlicher Anwendungen
Alarmauslösung	Intelligentes Video und externe Signale
Alarmereignisse	Datei-Upload über FTP, HTTP und E-Mail; Benachrichtigung per E-Mail, HTTP und TCP; Aktivierung externer Ausgänge, Videopuffer zum Speichern von Vor- und Nachalarmbildern
Allgemein	
Gehäuse	AXIS Q1922: Zinkgehäuse AXIS Q1922-E: Aluminiumgehäuse gemäß IP66 und Germaniumschleife
Arbeitsspeicher	128 MB RAM, 128 MB Flash
Stromversorgung	Power over Ethernet (IEEE 802.3af) Klasse 3 AXIS Q1922: 8-20 V DC, max. 9 W oder 20-24 V AC 50-60 Hz, max. 14 VA, Netzeil nicht im Lieferumfang enthalten AXIS Q1922-E: 8-20 V DC, max. 13 W oder 20-24 V AC 50-60 Hz, max. 20 VA, Netzeil nicht im Lieferumfang enthalten
Anschlüsse	RJ-45 10BASE-T/100BASE-TX PoE, Anschlussblock für Netzspannung, Anschlussblock für zwei konfigurierbare Eingänge/Ausgänge 3,5 mm Mikro/Line-Eingang, 3,5 mm Line-Ausgang RS-422/RS-485 AXIS Q1922/-E: Anschlussblock für Heizung
Lokaler Speicher	SD/SDHC-Speicherkartensteckplatz (Speicherkarte nicht im Lieferumfang enthalten)
Betriebsbedingungen	-40 °C bis +60 °C AXIS Q1922: Relative Luftfeuchtigkeit 20-80 % (nicht kondensierend) AXIS Q1922-E: Relative Luftfeuchtigkeit 10-85%
Zulassungen	EN 55022 Klasse A, EN 61000-3-2, EN 61000-3-3, EN 55024, EN 50121-4, EN 61000-6-1, EN 61000-6-2, EN 60950-1, FCC Teil 15 Abschnitt B Klasse A, VCCI Class A ITE, IEC 60068-2-1, IEC 60068-2-2, IEC 60068-2-6, IEC 60068-2-27, IEC 60068-2-78, KC Class A Q1922-E: EN 60950-22, IEC 60068-2-6, IEC 60068-2-27 (Stöße/erschütterungen), IEC 60529 IP66
Gewicht	AXIS Q1922: 950 g - 970 g AXIS Q1922-E: 3.475 g - 3.650 g
Im Lieferumfang enthaltenes Zubehör	Anschlusszubehör, Installationsanleitung, CD mit Benutzerhandbuch, Aufzeichnungssoftware, Installations- und Verwaltungsprogramme, Windows-Driver-Lizenz für einen Benutzer AXIS Q1922-E: Wandhalterung, 5 m Ethernet-Kabel

\*\*Dieses Produkt enthält Software, die vom OpenSSL-Projekt zur Verwendung im OpenSSL Toolkit entwickelt wurde. ([www.openssl.org](http://www.openssl.org))

Weitere Informationen finden Sie unter [www.axis.com](http://www.axis.com)

Abmessungen: AXIS Q1922 Wärmebild-Netzwerk-Kamera



©2012 Axis Communications AB. AXIS COMMUNICATIONS, AXIS, ETRAX, ARTPEC und VAPIX sind eingetragene sowie in verschiedenen Gerichtsbezirken beantragte Marken von Axis. Alle weiteren Firmen und Produktnamen sind Marken oder eingetragene Marken der jeweiligen Firmen. Wir behalten uns das Recht vor, Änderungen ohne vorherige Ankündigungen vorzunehmen.



Abbildung B.3: Aus dem Datenblatt der Wärmebildkamera Axis Q1922.

# Anhang C

## Daten

$x$	$y$	$pan$ laut Funktion	$pan$ laut LUT	$pan$ angefahren	$pan$ Grundwahrheit
28	421	-110,452955	-110	-110,001801	-105,136703
79	455	-108,207425	-108	-108,003304	-103,514000
89	476	-107,847938	-108	-108,002296	-103,241501
110	516	-107,050836	-107	-106,997002	-102,696602
122	537	-106,574991	-107	-106,995697	-102,330200
122	1065	-108,845919	-109	-108,998199	-104,205399
125	415	-105,747393	-106	-105,998703	-105,164902
137	424	-105,221382	-105	-105,004204	-101,555298
140	590	-105,956138	-106	-105,999100	-101,793503
143	930	-107,253312	-107	-107,002899	-102,765099
149	556	-105,339498	-105	-104,997498	-101,532402
195	623	-103,358345	-103	-102,997101	-99,991600
198	844	-104,050060	-104	-103,997101	-100,652397
222	452	-101,258996	-101	-101,000198	-98,770103
222	605	-101,930819	-102	-102,003799	-99,007896
237	427	-100,417141	-100	-99,997902	-98,208199
262	442	-99,277346	-99	-99,004204	-97,307800
262	565	-99,780539	-100	-100,000603	-97,462799
271	464	-98,934554	-99	-99,003304	-96,885201
286	405	-97,962331	-98	-98,001801	-96,446198

$x$	$y$	$pan$ laut Funktion	$pan$ laut LUT	$pan$ angefahren	$pan$ Grundwahrheit
307	421	-97,018547	-97	-96,998703	-95,536400
307	964	-98,585102	-99	-99,002296	-96,805298
310	541	-97,318211	-97	-97,001999	-95,776100
316	491	-96,847049	-97	-96,995003	-95,431099
316	528	-96,977548	-97	-97,003197	-95,542297
377	553	-94,044831	-94	-94,001999	-92,934799
392	393	-92,836671	-93	-92,996201	-92,143204
423	568	-91,802914	-92	-92,003700	-91,133904
426	528	-91,563128	-92	-91,997002	-90,967003
432	421	-90,999547	-91	-90,998703	-90,605103
444	510	-90,637370	-91	-90,995003	-90,198799
453	436	-90,025570	-90	-90,004204	-89,657799
459	491	-89,861213	-90	-90,001297	-89,477699
462	458	-89,641826	-90	-90,003304	-89,300797
487	553	-88,602911	-89	-88,995697	-88,406502
523	611	-86,889605	-87	-87,003799	-86,790703
526	525	-86,636851	-87	-87,000099	-86,663696
535	820	-86,396198	-87	-87,002998	-86,427696
569	504	-84,501078	-85	-84,995003	-84,780098
575	433	-84,129015	-84	-84,004204	-84,360901
581	464	-83,874134	-84	-84,003304	-84,172401
587	479	-83,597177	-84	-84,002296	-83,843597
587	599	-83,681057	-84	-83,996002	-83,865997
608	565	-82,621707	-83	-83,003700	-82,930298
617	418	-82,088796	-82	-81,998703	-82,355103
620	387	-81,916065	-82	-81,999298	-82,234001
620	602	-82,033670	-82	-81,999100	-82,290604
623	633	-81,884559	-82	-81,995903	-82,251503
657	531	-80,192957	-80	-79,997002	-80,550797
660	553	-80,044255	-80	-79,995697	-80,549698
669	455	-79,597145	-80	-80,000198	-80,019501

$x$	$y$	<i>pan</i> laut Funktion	<i>pan</i> laut LUT	<i>pan</i> angefahren	<i>pan</i> Grundwahrheit
678	510	-79,161542	-79	-78,995003	-79,529900
684	485	-78,869915	-79	-78,996101	-79,343300
702	933	-77,484971	-78	-78,000099	-78,041000
714	445	-77,417155	-78	-78,004204	-77,807404
724	433	-76,936487	-77	-77,000999	-77,376503
754	544	-75,402968	-76	-76,003197	-75,881302
757	399	-75,360817	-75	-74,996201	-75,858803
760	415	-75,208973	-75	-74,995499	-75,717697
760	648	-74,975740	-75	-75,002098	-75,406303
763	479	-75,020345	-75	-75,002296	-75,537399
775	498	-74,416960	-75	-75,002296	-74,739601
775	525	-74,388043	-75	-75,000099	-74,850998
869	562	-69,684852	-70	-70,001999	-70,077301
1046	899	-59,342969	-60	-60,000198	-59,698200
1106	516	-58,145244	-58	-57,996101	-58,070900
1131	547	-56,771653	-57	-57,001301	-56,738201
1146	728	-55,073888	-55	-55,001499	-55,291901
1149	596	-55,634738	-56	-56,003700	-55,713799
1255	507	-50,881885	-51	-51,003300	-50,843700
1331	921	-44,076399	-45	-45,000198	-44,682598
1377	620	-44,077634	-44	-44,003700	-44,239899

Tabelle C.1: Tabelle der *pan*-Daten aus der Evaluation von Experiment A. Erste und zweite Spalte Koordinaten im Master-Bild, dritte Spalte daraus errechneter Wert nach Funktion 6.1a, der gerundet sowohl in der Motorkarte als auch in der nächsten Spalte eingetragen wird, fünfte Spalte ist der Wert den die Pan-Tilt-Zoom-Kamera angenommen hat, nachdem ihr der Wert der vorherigen Spalte als Anweisung geschickt wurde und in der letzten Spalte ist der Wert der Grundwahrheit verzeichnet. Die 72 Einträge der Tabelle sind nach aufsteigendem  $x$ -Wert sortiert.

$x$	$y$	<i>tilt</i> laut Funktion	<i>tilt</i> laut LUT	<i>tilt</i> angefahren	<i>tilt</i> Grundwahrheit
620	387	-6,489948	-7	-7,000300	-6,559700



$x$	$y$	<i>tilt</i> laut Funktion	<i>tilt</i> laut LUT	<i>tilt</i> angefahren	<i>tilt</i> Grundwahrheit
392	393	-6,875796	-7	-7,000300	-6,803500
757	399	-6,835632	-7	-7,000300	-6,934700
286	405	-7,388244	-8	-8,003400	-7,084700
125	415	-7,723472	-8	-8,003400	-7,131600
760	415	-7,522812	-8	-8,003400	-7,544100
617	418	-7,838696	-8	-8,003400	-7,712800
28	421	-7,875564	-8	-8,003400	-7,150300
307	421	-8,092626	-8	-8,003400	-7,694100
432	421	-8,088876	-8	-8,003400	-7,806600
137	424	-8,128742	-9	-8,997200	-7,450300
237	427	-8,329404	-9	-8,997200	-7,806600
575	433	-8,527622	-9	-8,997200	-8,359700
724	433	-8,354484	-9	-8,997200	-8,359700
453	436	-8,733414	-9	-8,997200	-8,331600
262	442	-8,994129	-9	-8,997200	-8,359700
714	445	-8,886036	-9	-8,997200	-8,866000
222	452	-9,411169	-10	-10,000300	-8,537800
79	455	-9,425556	-10	-10,000300	-8,378400
669	455	-9,377176	-10	-10,000300	-9,278400
462	458	-9,682441	-10	-10,000300	-9,259700
271	464	-9,953388	-10	-10,000300	-9,194100
581	464	-9,860698	-10	-10,000300	-9,625300
89	476	-10,351138	-11	-11,003400	-9,194100
587	479	-10,500532	-11	-11,003400	-10,290900
763	479	-10,269972	-11	-11,003400	-10,244000
684	485	-10,645996	-11	-11,003400	-10,600300
316	491	-11,131884	-12	-11,997100	-10,319100
459	491	-11,107288	-12	-11,997100	-10,600300
775	498	-11,061762	-11	-11,003400	-11,228400
569	504	-11,590278	-12	-11,997100	-11,331500
1255	507	-10,150608	-10	-10,000300	-10,253400

$x$	$y$	$tilt$ laut Funktion	$tilt$ laut LUT	$tilt$ angefahren	$tilt$ Grundwahrheit
444	510	-11,930651	-12	-11,997100	-11,350300
678	510	-11,723561	-12	-11,997100	-11,519000
110	516	-12,106353	-13	-13,000300	-10,478400
1106	516	-11,031669	-11	-11,003400	-11,200300
526	525	-12,525204	-13	-13,000300	-12,053400
775	525	-12,211962	-13	-13,000300	-12,240900
316	528	-12,725511	-13	-13,000300	-11,659600
426	528	-12,709781	-13	-13,000300	-11,931500
657	531	-12,647036	-13	-13,000300	-12,381500
122	537	-13,023444	-14	-14,003400	-11,247100
310	541	-13,283028	-14	-14,003400	-12,128400
754	544	-13,055133	-13	-13,000300	-13,047100
1131	547	-12,259488	-12	-11,997100	-12,456500
377	553	-13,793366	-14	-14,003400	-12,719000
487	553	-13,748046	-14	-14,003400	-12,906500
660	553	-13,578852	-14	-14,003400	-13,356500
149	556	-13,862078	-15	-14,997100	-12,090900
869	562	-13,594844	-14	-14,003400	-13,759600
262	565	-14,302932	-15	-14,997100	-12,822100
608	565	-14,152076	-15	-14,997100	-13,862700
423	568	-14,420172	-15	-14,997100	-13,431500
140	590	-15,312907	-16	-16,000200	-13,262700
1149	596	-14,252638	-15	-14,997100	-14,490900
587	599	-15,615292	-16	-16,000200	-15,175200
620	602	-15,704203	-16	-16,000200	-15,447100
222	605	-15,997972	-17	-17,003300	-14,237700
523	611	-16,183584	-17	-17,003300	-15,381500
1377	620	-14,385082	-15	-14,997100	-13,675200
195	623	-16,752532	-18	-17,997101	-14,725300
623	633	-17,006142	-18	-17,997101	-16,515800
760	648	-17,424147	-18	-17,997101	-17,406401

$x$	$y$	<i>tilt</i> laut Funktion	<i>tilt</i> laut LUT	<i>tilt</i> angefahren	<i>tilt</i> Grundwahrheit
1146	728	-19,717621	-20	-20,003300	-20,106400
535	820	-24,871032	-26	-26,003201	-23,715700
198	844	-26,001657	-27	-26,996901	-22,525101
1046	899	-26,970328	-27	-26,996901	-27,681299
1331	921	-26,764254	-27	-26,996901	-26,593800
143	930	-29,506986	-31	-31,000000	-24,981300
702	933	-29,207636	-30	-29,996901	-28,984400
307	964	-30,888852	-32	-32,003101	-27,362499
122	1065	-34,943892	-36	-35,996700	-29,350000

Tabelle C.2: Tabelle der *tilt*-Daten aus der Evaluation von Experiment A. Erste und zweite Spalte Koordinaten im Master-Bild, dritte Spalte daraus errechneter Wert nach Funktion 6.1b, der gerundet sowohl in der Motorkarte als auch in der nächsten Spalte eingetragen wird, fünfte Spalte ist der Wert den die Pan-Tilt-Zoom-Kamera angenommen hat, nachdem ihr der Wert der vorherigen Spalte als Anweisung geschickt wurde und in der letzten Spalte ist der Wert der Grundwahrheit verzeichnet. Die 72 Einträge der Tabelle sind nach aufsteigendem  $y$ -Wert sortiert.

$x$	$y$	<i>pan</i> laut Funktion	<i>pan</i> laut LUT	<i>pan</i> angefahren	<i>pan</i> Grundwahrheit
368	725	-93,268661	-93	-93,003601	-92,300003
423	633	-90,161195	-90	-89,996201	-89,602798
426	605	-89,919099	-90	-90,003998	-89,497803
465	774	-88,462397	-89	-88,998001	-88,095703
511	550	-85,543039	-86	-85,998001	-85,688599
538	577	-84,265507	-84	-84,002502	-84,499702
590	602	-81,715661	-82	-81,997704	-82,063797
593	522	-81,436637	-82	-82,002998	-81,806000
623	565	-80,012051	-80	-79,999397	-80,458504
654	534	-78,435701	-79	-79,002998	-79,115997
657	550	-78,302461	-78	-78,001198	-78,816902
724	519	-74,956221	-75	-74,999901	-75,552803
727	577	-74,823067	-75	-75,002502	-75,462502

$x$	$y$	<i>pan</i> laut Funktion	<i>pan</i> laut LUT	<i>pan</i> angefahren	<i>pan</i> Grundwahrheit
733	605	-74,527961	-75	-74,996201	-75,202003
736	731	-74,399057	-75	-74,996399	-75,034897
763	550	-73,018891	-73	-72,994904	-73,698502
796	531	-71,377437	-72	-71,999901	-71,916100
806	577	-70,855055	-71	-70,999397	-71,459503
836	516	-69,395087	-70	-70,002998	-70,077499
854	593	-68,421983	-69	-69,003998	-69,088699
875	654	-67,285787	-67	-67,002800	-67,790497
900	541	-66,169993	-66	-66,001198	-66,769600
915	565	-65,379055	-66	-66,002502	-65,871002
979	528	-62,232363	-62	-61,996799	-62,568699
982	550	-62,031661	-62	-61,998001	-62,429199
1021	694	-59,680201	-60	-60,001598	-60,096901
1049	590	-58,533029	-59	-59,000900	-58,925499
1058	568	-58,141565	-58	-57,996300	-57,996300
1061	651	-57,734417	-58	-58,000999	-58,180199
1109	626	-55,346177	-56	-56,002399	-55,740398
1125	541	-54,838993	-55	-55,002998	-55,134399
1143	574	-53,799703	-54	-54,002499	-54,189499
1216	568	-50,102683	-50	-49,998001	-50,598099
1240	571	-48,862633	-49	-48,994900	-49,219700
1243	565	-48,738631	-49	-48,994900	-48,994900
1249	593	-48,293573	-49	-48,996300	-48,996300
1295	639	-45,680267	-46	-45,999298	-46,271198
1337	605	-43,696781	-44	-43,999401	-44,383999

Tabelle C.3: Tabelle der *pan*-Daten aus der Evaluation von Experiment B. Erste und zweite Spalte Koordinaten im Master-Bild, dritte Spalte daraus errechneter Wert nach Funktion 6.2a, der gerundet sowohl in der Motorkarte als auch in der nächsten Spalte eingetragen wird, fünfte Spalte ist der Wert den die Pan-Tilt-Zoom-Kamera angenommen hat, nachdem ihr der Wert der vorherigen Spalte als Anweisung geschickt wurde und in der letzten Spalte ist der Wert der Grundwahrheit verzeichnet. Die 38 Einträge der Tabelle sind nach aufsteigendem  $x$ -Wert sortiert.

$x$	$y$	<i>tilt</i> laut Funktion	<i>tilt</i> laut LUT	<i>tilt</i> angefahren	<i>tilt</i> Grundwahrheit
836	516	-12,109137	-13	-12,996300	-12,574400
724	519	-12,369124	-13	-12,996300	-12,818200
593	522	-12,522369	-13	-12,996300	-12,846300
979	528	-12,340711	-13	-12,996300	-13,080600
796	531	-12,835972	-13	-12,996300	-13,493100
654	534	-13,064829	-13	-12,996300	-13,399400
900	541	-13,111492	-14	-13,999400	-13,755600
1125	541	-12,445267	-13	-12,996300	-13,118100
763	550	-13,720245	-14	-13,999400	-14,336900
657	550	-13,776637	-14	-13,999400	-14,074400
982	550	-13,319037	-14	-13,999400	-13,971300
511	550	-13,707141	-14	-13,999400	-13,961900
1243	565	-13,015725	-14	-13,999400	-13,999400
915	565	-14,153557	-15	-15,002500	-14,833800
623	565	-14,442345	-15	-15,002500	-14,777500
1216	568	-13,276953	-14	-13,999400	-14,074400
1058	568	-13,897893	-15	-15,002500	-15,002500
1240	571	-13,300272	-14	-13,999400	-13,802500
1143	574	-13,858285	-15	-15,002500	-14,618100
806	577	-14,876346	-15	-15,002500	-15,527500
727	577	-14,951949	-15	-15,002500	-15,405600
538	577	-14,930214	-15	-15,002500	-15,227500
1049	590	-14,913445	-16	-15,996200	-15,724400
1249	593	-14,247651	-15	-15,002500	-15,002500
854	593	-15,519946	-16	-15,996200	-16,221201
590	602	-16,073365	-16	-15,996200	-16,296200
426	605	-16,019526	-16	-15,996200	-16,024401
1337	605	-14,335087	-15	-15,002500	-15,199400
733	605	-16,193595	-17	-16,999300	-16,699400
1109	626	-16,320649	-17	-16,999300	-17,271200

$x$	$y$	$tilt$ laut Funktion	$tilt$ laut LUT	$tilt$ angefahren	$tilt$ Grundwahrheit
423	633	-17,249493	-17	-16,999300	-17,196199
1295	639	-16,088295	-17	-16,999300	-16,990000
1061	651	-17,604537	-18	-18,002399	-18,602400
875	654	-18,202125	-19	-18,996201	-19,080601
1021	694	-19,646013	-20	-19,999300	-20,796200
368	725	-21,178840	-21	-21,002399	-20,758699
736	731	-21,777912	-22	-21,996201	-22,446100
465	774	-23,523165	-24	-24,002300	-23,036800

Tabelle C.4: Tabelle der  $tilt$ -Daten aus der Evaluation von Experiment B. Erste und zweite Spalte Koordinaten im Master-Bild, dritte Spalte daraus errechneter Wert nach Funktion 6.2b, der gerundet sowohl in der Motorkarte als auch in der nächsten Spalte eingetragen wird, fünfte Spalte ist der Wert den die Pan-Tilt-Zoom-Kamera angenommen hat, nachdem ihr der Wert der vorherigen Spalte als Anweisung geschickt wurde und in der letzten Spalte ist der Wert der Grundwahrheit verzeichnet. Die 38 Einträge der Tabelle sind nach aufsteigendem  $y$ -Wert sortiert.





# Abbildungsverzeichnis

2.1	Pfinder System mit ALIVE, einem Programm zur Interaktion mit virtuellen Agenten. . . . .	4
2.2	Unterschiedliche Kameras, die im System des VSAM Projektes genutzt werden. . . . .	4
2.3	Verschiedene Architekturen für Kamera-Netzwerke aus (Micheloni u. a., 2010). . . . .	5
2.4	Paarweise Zusammenhang zwischen mehreren Pan-Tilt-Zoom Kameras aus (Del Bimbo, Dini, Lisanti u. a., 2010). . . . .	6
3.1	Filtermasken für $L_{xx}$ , $L_{yy}$ und $L_{xy}$ . . . . .	12
3.2	Approximierte und diskretisierte Filtermasken $D_{xx}$ , $D_{yy}$ und $D_{xy}$ . . . . .	12
3.3	Skalierung der Filtermasken $D_{xx}$ , $D_{yy}$ und $D_{xy}$ für den zweiten bis vierten Layer der ersten Oktave. . . . .	13
3.4	Die 26 Nachbarn zu einem Pixel im Skalenraum. . . . .	14
3.5	Haar Wavelets für $x$ - und $y$ -Richtung. . . . .	15
3.6	Bestimmen der Orientierung eines Keypoints. . . . .	15
3.7	SURF Deskriptor Aufbau aus (Bay u. a., 2008). . . . .	16
3.8	Bildliche Darstellung von 'Nearest Neighbor Distance Ratio' (NNDR). . . . .	17
3.9	Ergebnis einer Hintergrundsubtraktion mit einfachem Blobdetektor. . . . .	23
4.1	Aufbau von Übersichts- und PTZ-Kamera. . . . .	25
4.2	Ausschnitt des Diagramms der Signal-Slot Verbindungen. . . . .	29
4.3	Gui der verwendeten CameraSystem Software. . . . .	30
5.1	Verwendeter zentrierter Ausschnitt der spiralförmig bewegten Pan-Tilt-Zoom-Kamera. . . . .	33
5.2	Übersicht der vier Schritte zur Konfiguration des Systems. . . . .	34
5.3	Von der Pan-Tilt-Zoom-Kamera abgefahrenere Spirale von Pan-Tilt-Werten. . . . .	35
5.4	Korrespondierende SURF Merkmale. . . . .	36
5.5	Korrespondierende SURF Merkmale nach Aussortierung mittels NNDR. . . . .	36
5.6	Übersicht der konsekutiven Schritte zur Erstellung einer flächendeckenden Zuordnungstabelle von Pixeln im Übersichtsbild zu Motorkoordinaten der Pan-Tilt-Zoom-Kamera. . . . .	37

5.7	Übersichtsbild und Pan-Tilt-Zoom Teilbild, dazu gefundene Korrespondenzen von SURF Merkmalen und bestimmte Homographie. . . . .	38
5.8	Abgelehnte und akzeptierte Transformation. . . . .	39
5.9	Gesammelte Werte aus einem Messdurchlauf. . . . .	40
5.10	Delaunay-Triangulation und daraus durch baryzentrische Interpolation entstandene Motorkarte. . . . .	41
5.11	Quadratische Funktionen als Annäherung für Pan- und Tilt-Messwerte. . .	43
5.12	Flächendeckend berechnete Pan-Tilt-Werte für das Übersichtsbild. . . . .	44
6.1	Für die Experimente verwendete Diskursbereiche. . . . .	45
6.2	Markierungen zur Fehlermessung bei Experiment A aus Sicht beider Kameras. . . . .	46
6.3	Gui-Screenshot des Konfigurationsdurchlaufs von Experiment A. . . . .	47
6.4	Pan-Wert Abweichung in Grad zwischen berechnetem Wert, sowie Grundwahrheit mit dem angefahrenen Wert in Experiment A. . . . .	48
6.5	Tilt-Wert Abweichung in Grad zwischen berechnetem Wert, sowie Grundwahrheit mit dem angefahrenen Wert in Experiment A. . . . .	49
6.6	Fehlermap von Experiment A. . . . .	50
6.7	Ergebnisbilder auf der Wiese in Richtung Kulisse. . . . .	51
6.8	Problemfälle Fenstersprosse und Rand des Fensters in der Pan-Tilt-Zoom-Kamera. . . . .	52
6.9	Pan- und Tilt-Fehlermap zu Experiment B. . . . .	53
6.10	Pan-Tilt-Wert Abweichung in Grad zwischen berechnetem Wert, sowie Grundwahrheit mit dem angefahrenen Wert in Experiment B. . . . .	53
6.11	Ergebnisbilder aus dem zweiten Stock in Richtung Kulisse. . . . .	54
6.12	Blick vom Turm der Übersichtskamera mit eingefügtem Teilbild der Pan-Tilt-Zoom-Kamera. . . . .	55
6.13	Ergebnisbilder vom Turm hinunter. . . . .	56
6.14	Pan-Werte gegen $x$ bei $y = 540$ in Experiment D1 für jeweils 30 durch lineare Regression bestimmte Funktionen. . . . .	57
6.15	Tilt-Werte gegen $y$ bei $x = 960$ in Experiment D1 für jeweils 30 durch lineare Regression bestimmte Funktionen. . . . .	58
6.16	Ergebnisbilder auf der Wiese in Richtung Pavillon. . . . .	59
6.17	Pan-Werte gegen $x$ bei $y = 540$ und Tilt-Werte gegen $y$ bei $x = 960$ von jeweils 30 ermittelten Funktionen in Experiment D1 und D2. . . . .	60
6.18	Ergebnisbilder auf der Wiese in Richtung Pavillon. . . . .	61
6.19	Bild aus der automatischen Objektverfolgung mit den Kameras am Instituteingang. . . . .	62
6.20	Ergebnisbilder am Instituteingang. . . . .	63
6.21	Automatisch fokussierte Markierungen in Experiment F . . . . .	64
6.22	Ergebnisbilder im Flur. . . . .	65

---

B.1	Aus dem Datenblatt der Übersichtskamera Axis Q1755. . . . .	72
B.2	Aus dem Datenblatt der Pan-Tilt-Zoom-Kamera Axis P5534. . . . .	73
B.3	Aus dem Datenblatt der Übersichtskamera Axis Q1755. . . . .	74



# Literaturverzeichnis

Arens, Michael

- 2004 *Repräsentation und Nutzung von Verhaltenswissen in der Bildfolgenauswertung*, Bd. 287, Dissertationen zur Künstlichen Intelligenz (DISKI), Zugl.: Karlsruhe, Univ., Diss., 2004, Akad. Verl.-Ges. Aka, Berlin, ISBN: 3898382877. (Siehe S. 68.)

Arens, Tilo, Frank Hettlich, Christian Karpfinger, Ulrich Kockelkorn, Klaus Lichtenegger und Hellmuth Stachel

- 2008 *Mathematik*, 1. Aufl. 2008, Spektrum Akademischer Verlag. (Siehe S. 20.)

Badri, Julie, Christophe Tilmant, Jean-Marc Lavest, Quonc-Cong Pham und Patrick Sayd

- 2007 „Camera-to-Camera Mapping for Hybrid Pan-Tilt-Zoom Sensors Calibration“, in *Image Analysis*, hrsg. von Bjarne Ersbøll und Kim Pedersen, Bd. 4522, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, S. 132–141, ISBN: 978-3-540-73039-2. (Siehe S. 5, 8, 9.)

Bay, Herbert, Andreas Ess, Tinne Tuytelaars und Luc Van Gool

- 2008 „Speeded-Up Robust Features (SURF)“, *Computer Vision and Image Understanding*, 110, 3, Similarity Matching in Computer Vision and Multimedia, S. 346–359, ISSN: 1077-3142. (Siehe S. 8, 11, 13, 16, 35.)

Begelfor, E. und M. Werman

- 2005 „How to put probabilities on homographies“, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27, 10 (Okt. 2005), S. 1666–1670, ISSN: 0162-8828. (Siehe S. 39.)

Bellotto, Nicola, Ben Benfold, Hanno Harland, Hans-Hellmut Nagel, Nicola Pirlo, Ian Reid, Eric Sommerlade und Chuan Zhao

- 2012 „Cognitive visual tracking and camera control“, *Computer Vision and Image Understanding*, 116, 3, Special issue on Semantic Understanding of Human Behaviors in Image Sequences, S. 457–471, ISSN: 1077-3142. (Siehe S. 5–7, 68.)

Blanchette, Jasmin und Mark Summerfield

- 2006 *C++ GUI Programming with Qt 4*, Prentice Hall PTR, Upper Saddle River, NJ, USA, ISBN: 0131872494. (Siehe S. 27.)

Bodor, R., R. Morlok und N. Papanikolopoulos

- 2004 „Dual-camera system for multi-level activity recognition“, in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, Bd. 1, S. 643–648. (Siehe S. 7.)

Bradski, G.

- 2000 „The OpenCV Library“, *Dr. Dobb's Journal of Software Tools*. (Siehe S. 23, 26, 38.)

Brémond, François, Monique Thonnat und Marcos Zúñiga

- 2006 „Video-understanding framework for automatic behavior recognition“, English, *Behavior Research Methods*, 38 (32006), S. 416–426, ISSN: 1554-351X. (Siehe S. 7.)

Brown, Matthew und D. Lowe

- 2002 „Invariant Features from Interest Point Groups“, in *BMVC 2002: 13th British Machine Vision Conference*, S. 253–262. (Siehe S. 14.)

Collins, Robert T., Alan J. Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt und Lambert Wixson

- 2000 *A System for Video Surveillance and Monitoring*, Technical report, Carnegie Mellon University, the Robotics Institute. (Siehe S. 4.)

Davis, J. und X. Chen

- 2003 „Calibrating pan-tilt cameras in wide-area surveillance networks“, in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, S. 144–149. (Siehe S. 7.)

Del Bimbo, A., F. Dini, A. Grifoni und F. Pernici

- 2008 „Exploiting Single View Geometry in Pan-Tilt-Zoom Camera Networks“, in *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications – M2SFA2 2008*, Andrea Cavallaro und Hamid Aghajan, Marseille, France. (Siehe S. 8.)

Del Bimbo, A., F. Dini, G. Lisanti und F. Pernici

- 2010 „Exploiting distinctive visual landmark maps in pan-tilt-zoom camera networks“, *Computer Vision and Image Understanding*, 114, 6, Special Issue on Multi-Camera and Multi-Modal Sensor Fusion, S. 611–623, ISSN: 1077-3142. (Siehe S. 5, 6, 8.)

Everts, I., N. Sebe und G. Jones

- 2007 „Cooperative Object Tracking with Multiple PTZ Cameras“, in *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, S. 323–330. (Siehe S. 5.)

Fischler, Martin A. und Robert C. Bolles

- 1981 „Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography“, *Commun. ACM*, 24, 6 (Juni 1981), S. 381–395, ISSN: 0001-0782. (Siehe S. 18.)

Gamma, Erich, Richard Helm, Ralph E. Johnson und John Vlissides

- 1995 *Design Patterns: Elements of Reusable Object-Oriented Software*, Professional Computing Series, Addison Wesley, Reading, MA, USA, ISBN: 978-0201633610. (Siehe S. 27, 29.)

Hampapur, A., S. Pankanti, A. Senior, Ying-Li Tian, L. Brown und R. Bolle

- 2003 „Face cataloger: multi-scale imaging for relating identity to location“, in *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2003*. S. 13–20. (Siehe S. 6, 7.)

Haritaoglu, I., D. Harwood und L.S. Davis

- 2000 „W4: real-time surveillance of people and their activities“, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22, 8 (Aug. 2000), S. 809–830, ISSN: 0162-8828. (Siehe S. 4.)

Horaud, Radu, David Knossow und Markus Michaelis

- 2006 „Camera cooperation for achieving visual attention“, *Machine Vision and Applications*, 16, 6, S. 1–2, ISSN: 0932-8092. (Siehe S. 7.)

Hu, Weiming, Tieniu Tan, Liang Wang und S. Maybank

- 2004 „A survey on visual surveillance of object motion and behaviors“, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34, 3 (Aug. 2004), S. 334–352, ISSN: 1094-6977. (Siehe S. 4.)

IJsselmuiden, Joris und Rainer Stiefelhagen

- 2010 „Towards High-Level Human Activity Recognition through Computer Vision and Temporal Logic“, in *KI 2010: Advances in Artificial Intelligence*, hrsg. von Rüdiger Dillmann, Jürgen Beyerer, UweD. Hanebeck und Tanja Schultz, Bd. 6359, Lecture Notes in Computer Science, Springer Berlin Heidelberg, S. 426–435, ISBN: 978-3-642-16110-0. (Siehe S. 5.)



Jain, Ankur, D. Kopell, K. Kakligian und Yuan-Fang Wang

- 2006 „Using Stationary-Dynamic Camera Assemblies for Wide-area Video Surveillance and Selective Attention“, in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, Bd. 1, S. 537–544. (Siehe S. 7.)

KaewTraKulPong, P. und R. Bowden

- 2001 „An improved adaptive background mixture model for real-time tracking with shadow detection“, in *Proceedings of 2nd European Workshop on Advanced Video Based Surveillance Systems*, Bd. 5308. (Siehe S. 22.)

Khan, S. und M. Shah

- 2003 „Consistent labeling of tracked objects in multiple cameras with overlapping fields of view“, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25, 10 (Okt. 2003), S. 1355–1360, ISSN: 0162-8828. (Siehe S. 8.)

Liao, Hsien-Chou, Ming-Ho Pan, Hung-Wei Hwang, Min-Chih Chang und Chen Po-Cheng

- 2011 „An automatic calibration method based on feature point matching for the cooperation of wide-angle and pan-tilt-zoom cameras“, *Information Technology And Control*, 40, 1, S. 41–47. (Siehe S. 5, 8.)

Low, Yi-Qian, Sze-Wei Lee, Bok-Min Goi und Mow-Song Ng

- 2011 „A New SIFT-Based Camera Calibration Method for Hybrid Dual-Camera“, in *Informatics Engineering and Information Science*, hrsg. von Azizah Abd Manaf, Akram Zeki, Mazdak Zamani, Suriayati Chuprat und Eyas El-Qawasmeh, Bd. 252, Communications in Computer and Information Science, Springer Berlin Heidelberg, S. 96–103, ISBN: 978-3-642-25453-6. (Siehe S. 5, 8.)

Lowe, David G.

- 1999 „Object recognition from local scale-invariant features“, in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, Bd. 2, S. 1150–1157. (Siehe S. 8, 11.)

Lv, Fengjun, Tao Zhao und R. Nevatia

- 2006 „Camera calibration from video of a walking human“, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28, 9 (Sep. 2006), S. 1513–1518, ISSN: 0162-8828. (Siehe S. 8.)

Marcenaro, L., L. Marchesotti und C.S. Regazzoni

- 2002 „A multi-resolution outdoor dual camera system for robust video-event metadata extraction“, in *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, Bd. 2, S. 1184–1189. (Siehe S. 7.)

Marchesotti, L., L. Marcenaro und C. Regazzoni

- 2003 „Dual camera system for face detection in unconstrained environments“, in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, Bd. 1, (siehe S. 6, 7.)

Michelsoni, C., B. Rinner und G.L. Foresti

- 2010 „Video Analysis in Pan-Tilt-Zoom Camera Networks“, *Signal Processing Magazine, IEEE*, 27, 5, S. 78–90, ISSN: 1053-5888. (Siehe S. 4, 5.)

Mohanty, K.K. und M.K. Gellaboina

- 2011 „A semi-automatic relative calibration of a fixed and PTZ camera pair for master-slave control“, in *Visual Information Processing (EUVIP), 2011 3rd European Workshop on*, S. 229–234. (Siehe S. 5, 8.)

Münch, David, Joris IJsselmuiden, Ann-Kristin Grosselfinger, Michael Arens und Rainer Stiefelhagen

- 2012 „Rule-Based High-Level Situation Recognition from Incomplete Tracking Data“, in *Rules on the Web: Research and Applications*, hrsg. von Antonis Bikakis und Adrian Giurca, Bd. 7438, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, S. 317–324, ISBN: 978-3-642-32688-2. (Siehe S. 68.)

Nagel, Hans-Hellmut

- 2004 „Steps Towards a Cognitive Vision System“, *AI Magazine*, 25, S. 31–50. (Siehe S. 68.)

Neubeck, A. und L. Van Gool

- 2006 „Efficient Non-Maximum Suppression“, in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, Bd. 3, S. 850–855. (Siehe S. 14.)

Pham, Vu, Phong Vo, Hung Thanh Vu und Hoai Bac Le

- 2010 „GPU Implementation of Extended Gaussian Mixture Model for Background Subtraction“, in *RIVF*, S. 1–4. (Siehe S. 31.)

Raguram, Rahul, Jan-Michael Frahm und Marc Pollefeys

- 2008 „A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus“, in *Computer Vision – ECCV 2008*, hrsg. von David Forsyth, Philip Torr und Andrew Zisserman, Bd. 5303, Lecture Notes in Computer Science, Springer Berlin Heidelberg, S. 500–513, ISBN: 978-3-540-88685-3. (Siehe S. 18.)

Senior, A.W., A. Hampapur und M. Lu

- 2005 „Acquiring Multi-Scale Images by Pan-Tilt-Zoom Control and Automatic Multi-Camera Calibration“, in *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, Bd. 1, S. 433–438. (Siehe S. 5, 8.)

Sinha, Sudipta N. und Marc Pollefeys

- 2006 „Pan-tilt-zoom camera calibration and high-resolution mosaic generation“, *Computer Vision and Image Understanding*, 103, 3, Special issue on Omnidirectional Vision and Camera Networks, S. 170–183, ISSN: 1077-3142. (Siehe S. 7, 8.)

Sinha, Sudipta und Marc Pollefeys

- 2010 „Camera Network Calibration and Synchronization from Silhouettes in Archived Video“, *International Journal of Computer Vision*, 87 (32010), S. 266–283, ISSN: 0920-5691. (Siehe S. 8.)

Song, Kai-Tai und Jen-Chao Tai

- 2006 „Dynamic Calibration of Pan-Tilt-Zoom Cameras for Traffic Monitoring“, *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 36, 5 (Okt. 2006), S. 1091–1103, ISSN: 1083-4419. (Siehe S. 8.)

Stauffer, C. und W. E. L. Grimson

- 1999 „Adaptive background mixture models for real-time tracking“, in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Bd. 2. (Siehe S. 22.)

Suzuki, Satoshi und Keiichi Abe

- 1985 „Topological structural analysis of digitized binary images by border following“, *Computer Vision, Graphics, and Image Processing*, 30, 1, S. 32–46, ISSN: 0734-189X. (Siehe S. 23.)

Szeliski, Richard

- 2011 „Feature detection and matching“, in *Computer Vision, Texts in Computer Science*, Springer London, S. 181–234, ISBN: 978-1-84882-935-0. (Siehe S. 17, 35.)

Szwoch, Grzegorz, Piotr Dalka, Andrzej Ciarkowski, Piotr Szczuko und Andrzej Czyzewski

- 2011 „Visual object tracking system employing fixed and PTZ cameras“, *Int. Dec. Tech.*, 5, 2 (Juni 2011), S. 177–188, ISSN: 1872-4981. (Siehe S. 5, 7.)

- Tian, Ying-li, Lisa Brown, Arun Hampapur, Max Lu, Andrew Senior und Chiao-fe Shu  
2008 „IBM smart surveillance system (S3): event based video surveillance system with an open and extensible framework“, *Mach. Vision Appl.*, 19, 5-6 (Sep. 2008), S. 315–327, ISSN: 0932-8092. (Siehe S. 5.)
- Tresadern, Philip A. und Ian D. Reid  
2008 „Camera calibration from human motion“, *Image and Vision Computing*, 26, 6, S. 851–862, ISSN: 0262-8856. (Siehe S. 8.)
- Tsai, R.  
1987 „A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses“, *Robotics and Automation, IEEE Journal of*, 3, 4 (Aug. 1987), S. 323–344, ISSN: 0882-4967. (Siehe S. 7.)
- Tsuruoka, S., T. Yamaguchi, K. Kato, T. Yoshikawa und T. Shinogi  
2001 „A camera control based on fuzzy behavior recognition of lecturer for distance lecture“, in *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, Bd. 2, S. 940–943. (Siehe S. 6, 7.)
- Utsumi, Y., E. Sommerlade, N. Bellotto und I. Reid  
2012 „Cognitive active vision for human identification“, in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, S. 1238–1245. (Siehe S. 7.)
- Wren, C.R., A. Azarbayejani, T. Darrell und A.P. Pentland  
1997 „Pfinder: real-time tracking of the human body“, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19, 7 (Juli 1997), S. 780–785, ISSN: 0162-8828. (Siehe S. 3.)
- Wu, Ziyang und R.J. Radke  
2012 „Using scene features to improve wide-area video surveillance“, in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, S. 50–57. (Siehe S. 9, 67.)
- Xiong, Bin und Xiaoqing Ding  
2012 „A Master-Slave System for Intelligent Visual Surveillance“, in *Measuring Technology and Mechatronics Automation in Electrical Engineering*, hrsg. von Zhixiang Hou, Bd. 135, Lecture Notes in Electrical Engineering, Springer US, S. 37–42, ISBN: 978-1-4614-2185-6. (Siehe S. 5, 9.)
- Yi, Richard Da Xu, Junbin Gao und Michael Antolovich  
2008 „Novel methods for high-resolution facial image capture using calibrated PTZ and static cameras“, in *Multimedia and Expo, 2008 IEEE International Conference on*, S. 45–48. (Siehe S. 5–7.)

You, Li, Li Song und Jia Wang

- 2010 „Automatic Weak Calibration of Master–Slave Surveillance System Based on Mosaic Image“, in *Pattern Recognition (ICPR), 2010 20th International Conference on*, S. 1824–1827. (Siehe S. 5, 8.)

Zhou, Xuhui, Robert T. Collins, Takeo Kanade und Peter Metes

- 2003 „A master–slave system to acquire biometric imagery of humans at distance“, in *First ACM SIGMM international workshop on Video surveillance, IWVS '03*, ACM, Berkeley, California, S. 113–120, ISBN: 1-58113-780-X. (Siehe S. 8.)

Zivkovic, Zoran

- 2004 „Improved adaptive Gaussian mixture model for background subtraction“, in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, Bd. 2, S. 28–31. (Siehe S. 22.)

Zivkovic, Zoran und Ferdinand van der Heijden

- 2006 „Efficient adaptive density estimation per image pixel for the task of background subtraction“, *Pattern Recognition Letters*, 27, 7, S. 773–780, ISSN: 0167-8655. (Siehe S. 22.)

Zouba, Nadia, François Brémond und Monique Thonnat

- 2010 „An Activity Monitoring System for Real Elderly at Home: Validation Study“, in *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, S. 278–285. (Siehe S. 7.)