

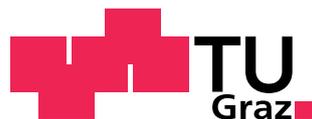
Diplomarbeit

# Videobasierte, monokulare Positionsbestimmung von Objekten auf einer Geländefläche

Gerd Hübscher

---

Institut für Technische Informatik  
Technische Universität Graz  
Vorstand: O. Univ.-Prof. Dipl.-Ing. Dr. techn. Reinhold Weiß



Begutachter: O. A.o.Univ.-Prof. Dipl.-Ing. Dr. techn. Eugen Brenner  
Betreuer: O. A.o.Univ.-Prof. Dipl.-Ing. Dr. techn. Eugen Brenner

Graz, im März 2010

## **Kurzfassung**

Die gegenständliche Diplomarbeit setzt sich mit dem Entwurf eines Systems auseinander, das eine videogestützte monokulare Positionsbestimmung von Objekten ermöglicht, sofern deren Bewegung auf einer Geländefläche angenommen werden kann. Es wurden die mathematischen Grundlagen zum Aufbau und zur Umwandlung von Kameramodellen erarbeitet, um eine virtuelle Ansicht auf ein Modell der beobachteten Geländefläche so wählen zu können, dass die virtuelle Ansicht mit dem durch eine Kamera aufgenommenen Bild der Geländefläche übereinstimmt. Dies ermöglicht die Zuordnung von zweidimensionalen Bildpunkten zu korrespondierenden dreidimensionalen Weltpunkten auf der Geländefläche, sodass der Fußpunkt eines mittels Bildverarbeitung verfolgten Objektes in Weltkoordinaten bestimmt werden kann. Schließlich wurde ein entsprechendes System aufgebaut und seine Funktionalität in sportspezifischen Anwendungen, allen voran dem Skirennlauf, evaluiert.

## **Abstract**

The given diploma thesis deals with the design as well as the implementation of a system for videobased, monoculare position determination of objects, provided that their movement can be assumed to be on an area surface. Mathematical principles to transform and to build up camera models were developed, in order to choose a virtual viewport relative to the model in such a way, that the virtual view onto the model corresponds to the picture taken by the real camera. This allows the assignment between two-dimensional points in image space and three-dimensional points on the area surface in world space, so that the world space coordinates of the bottom of an object can be detected by means of image processing. In conclusion such a system was built up and evaluated in sports applications, mainly ski races.

## EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am .....

.....  
(Unterschrift)

## Danksagung

Diese Diplomarbeit wurde im Studienjahr 2009/10 am Institut für Technische Informatik an der Technischen Universität Graz in Kooperation mit der SELTEC GmbH durchgeführt.

Besonders möchte ich mich bei Herrn Dr. Brenner für die Betreuung dieser Arbeit und die zahlreichen anregenden Diskussionen, Ratschläge und Hilfestellungen bedanken.

Außerdem gilt mein Dank den Mitarbeitern von SELTEC, die dieses Projekt ermöglicht haben, allen voran Herrn Dipl.-Ing. Reinhard Buzek und Herrn Peter Gridling.

Schließlich bedanke ich mich bei Herrn Dipl. Phys. Peter Butzhammer für die wertvollen Anregungen in Bezug auf Objektverfolgung und Kalmanfilterung.

Graz, im März 2010

Gerd Hübscher

# Inhaltsverzeichnis

<b>Symbolverzeichnis</b>	<b>7</b>
<b>1 Einleitung</b>	<b>9</b>
1.1 Einführung . . . . .	9
1.2 Aufgabenstellung . . . . .	10
1.3 Entwicklungsumfeld . . . . .	10
1.4 Stand der Technik . . . . .	10
1.5 Neuerungen . . . . .	11
1.6 Aufbau dieser Diplomarbeit . . . . .	11
<b>2 Kamerakalibrierung</b>	<b>13</b>
2.1 Kameramodelle . . . . .	13
2.1.1 Orientierung der Kamera . . . . .	14
2.1.2 Photogrammetrie . . . . .	15
2.1.3 Virtuelle Modellierung . . . . .	17
2.1.4 Überführung der Modelle . . . . .	19
2.2 Berechnung der Kameraparameter . . . . .	22
2.2.1 Direkte lineare Transformation (DLT) . . . . .	23
2.3 Dekomposition der Transformationsmatrix . . . . .	24
2.3.1 RQ-Dekomposition . . . . .	25
2.4 Nichtlineare Optimierung . . . . .	26
2.4.1 Optimierungsverfahren . . . . .	27
2.4.2 Optimierung des Lochkameramodells . . . . .	28
2.4.3 Optimierung des computergraphischen Modells . . . . .	28
2.5 Numerische Probleme . . . . .	28
2.5.1 Normalisierung der Kalibrierpunkte . . . . .	29
2.6 Ergebnisse . . . . .	29
2.6.1 Messungen . . . . .	29
2.6.2 Auswirkungen der Lage der Kalibrierpunkte . . . . .	30
2.6.3 Linsenverzerrungen . . . . .	34
<b>3 Tiefenrekonstruktion</b>	<b>35</b>
3.1 Grundlegende Überlegungen . . . . .	35
3.2 Modellierung . . . . .	37
3.3 Modellbildung aus Luftbildaufnahmen . . . . .	37
3.4 Modellbildung aus geodätisch erfassten Geländepunkten . . . . .	37

3.4.1	Differenzielle GNSS Messungen . . . . .	39
3.4.2	Vergleich der Messmethoden . . . . .	40
3.4.3	Überführung in kartesische Koordinaten mit Referenzpunkt . . . . .	42
3.4.4	Aufbau des Modells . . . . .	44
3.5	Wiedergewinnung des Weltpunktes . . . . .	44
3.5.1	Schnittpunktgewinnung . . . . .	45
3.6	Ergebnisse . . . . .	47
<b>4</b>	<b>Objekterkennung</b>	<b>50</b>
4.1	Ansätze zur Bildererkennung für Sportanwendungen . . . . .	50
4.2	Bildverarbeitung . . . . .	51
4.2.1	Laufzeitoptimierung . . . . .	52
4.2.2	Hintergrundschatzung . . . . .	52
4.2.3	Keying . . . . .	53
4.2.4	Morphologische Operationen . . . . .	53
4.2.5	Objektextraktion . . . . .	54
4.3	Objektverarbeitung . . . . .	54
4.3.1	Bildmomente . . . . .	54
4.3.2	Gewinnung des Fußpunktes . . . . .	55
4.3.3	Objektverfolgung . . . . .	55
4.4	Ergebnisse . . . . .	55
<b>5</b>	<b>Integration</b>	<b>58</b>
5.1	Anforderungen . . . . .	58
5.2	Datenabhängigkeiten . . . . .	59
5.3	Logische Verarbeitungseinheiten . . . . .	60
5.4	Entwicklungsumgebung . . . . .	62
5.5	Physikalische Verarbeitungseinheiten . . . . .	63
5.6	Umsetzung und Anwendung . . . . .	63
5.6.1	Zielhardware . . . . .	64
5.6.2	Kommunikation . . . . .	64
5.6.3	Vermessung und Modell . . . . .	65
5.6.4	Ergebnisse . . . . .	67
<b>6</b>	<b>Zusammenfassung</b>	<b>70</b>
6.1	Ausblick . . . . .	70
6.2	Schlusswort . . . . .	71
<b>A</b>	<b>Ergänzungen</b>	<b>72</b>
A.1	Matlab . . . . .	72
A.2	Datenblätter . . . . .	74
	<b>Literaturverzeichnis</b>	<b>75</b>
	<b>Abbildungsverzeichnis</b>	<b>79</b>
	<b>Index</b>	<b>82</b>

# Symbolverzeichnis

$\alpha$	Seitenverhältnis des Lochkameramodells.
$\tilde{\mathbf{C}}$	Der Positionsvektor des Kameramodells.
$\mathbf{C}$	Der nicht-normierte Positionsvektor des Kameramodells.
$\eta$	Geographische Länge in ellipsoidischen Koordinaten (WGS 84 Referenzellipsoid).
$f$	Brennweite des Lochkameramodells.
$G$	Die gesamte lineare Abbildung des computergraphischen Kameramodells.
$G_{\text{Ansicht}}$	Die Ansichtsmatrix des computergraphischen Modells.
$G_{\text{Projektion}}$	Die Projektionsmatrix des computergraphischen Modells.
$h$	Geographische Höhe in ellipsoidischen Koordinaten (WGS 84 Referenzellipsoid).
$P$	Die gesamte lineare Abbildung des realen Kameramodells.
$\varphi$	Geographische Breite in ellipsoidischen Koordinaten (WGS 84 Referenzellipsoid).
$R$	Die Rotationsmatrix des Lochkameramodells.
$K$	Innere Orientierung des Lochkameramodells.
$\mathbf{t}$	Der Translationsvektor des Kameramodells.
$\mathbf{X}_c$	Ein beliebiger Punkt im Bildraum, gegeben durch $x$ , $y$ und die homogenen Koordinate $w$ .

$\mathbf{X}_f$	Der Schnittpunkt zwischen dem bei Rückprojektion eines Bildpunktes entstehenden Sichtstrahles und der Farplane.
$\mathbf{X}_n$	Der Schnittpunkt zwischen dem bei Rückprojektion eines Bildpunktes entstehenden Sichtstrahles und der Nearplane.
$\mathbf{X}_w$	Ein beliebiger Punkt im Objektraum (Welt), gegeben durch $X, Y, Z$ und die homogenen Koordinate $W$ .

# Kapitel 1

## Einleitung

### 1.1 Einführung

Solange es in der Geschichte sportliche Wettkämpfe gibt, solange gibt es auch den Wunsch, athletische Leistungen objektiv bewerten und insbesondere vergleichen zu können. Ein möglicher, sehr verbreiteter Parameter zur objektiven Bewertung von Leistungen ist die dazu benötigte Zeit. Sie ist ein Unterscheidungsmerkmal, eine strukturelle Komponente, die es ermöglicht, Ereignisse voneinander abzugrenzen. Doch trotz aller Entwicklungen auf diesem Gebiet, angefangen von frühen rein mechanischen Systemen, die noch gespannte Drähte als Auslöser verwendeten, über halbelektronische bis hin zu vollautomatisierten Zeitmesssystemen, wie wir sie heute kennen, ist die Zeit für sich ein Parameter geblieben, der zwar eine Bewertung ermöglicht, die erbrachte Leistung aber im Allgemeinen nicht beschreibt. Dies hat hauptsächlich zwei Gründe: Zum einen stellt jede Zeitmessung nur eine Momentaufnahme dar und jede Steigerung der Auflösung auf diesem Wege ist mit relativ hohem Aufwand und damit auch mit Kosten verbunden. Zum anderen liefert die reine Zeitmessung keinerlei Auskunft über das, was ich als qualitatives Verhalten der Athletin oder des Athleten bezeichnen möchte. Im Ursprung ist das die Bewegung.

*Jede Bewegung verläuft in der Zeit und hat ein Ziel.*  
[Aristoteles, *Nikomachische Ethik*]

Die Position einer Sportlerin oder eines Sportlers zu jedem Zeitpunkt zu kennen, bedeutet nicht nur den Gewinn der zeitlichen Ableitungen dieser Größe, also Geschwindigkeit und Beschleunigung. Es bedeutet auch die Öffnung eines Optionsraumes für eine Fülle von Vergleichsmöglichkeiten. So sind nicht nur abgeleitete Größen zugänglich, sondern auch Rückschlüsse auf Vorgänge möglich, die sich unmittelbar auf den Bewegungsverlauf auswirken und in Kombination mit einem entsprechenden Dynamikmodell direkt ausgewertet werden können, um sie Athleten, Trainern und Zuschauern direkt zur Verfügung zu stellen. Dies ermöglicht es, einen tieferen Einblick in die Bewegungsabläufe einer Sportart und deren Auswirkungen zu gewinnen.

## 1.2 Aufgabenstellung

Ziel dieser Arbeit ist es, ein System zu entwerfen und zu optimieren, mit dem in erster Linie die Lage (der Ort) und weiters auch die Bahn (Spur) eines Athleten bestimmt werden kann. Dies soll keinerlei Beeinflussung für letzteren mit sich bringen und somit sowohl berührungslos als auch ohne den Einsatz von aktiven oder passiven Transpondern erfolgen. Die Erfassung selbst soll dabei dynamisch erfolgen und „weichen“ Echtzeitbedingungen genügen. Für einen ersten Prototypen wird der Anwendungsfall eines Skirennens gewählt.

## 1.3 Entwicklungsumfeld

Das Wesen von Sportveranstaltungen stellt ganz besondere Anforderungen an ein entsprechendes System zur Ortsbestimmung. Es muss flexibel und zuverlässig sein, rasch auf- und wieder abbaubar und soll sich nahtlos in bestehende Installationen integrieren lassen. Da erst ab einer gewissen Bewerbgröße Rahmenbedingungen gewährleistet sind, die den Einsatz ermöglichen, ist in der Entwicklung mit einer geringen Anzahl von Testfällen zu rechnen. Gleichmaßen ist jedoch die Nachbildung realer Bedingungen zu Testzwecken im Labor schwierig beziehungsweise unmöglich, da weder entsprechende externe Datenquellen zu Vergleichszwecken, noch die Vielzahl an Störeinflüssen modelliert werden können. Es bedarf daher großer Erfahrung im Bereich von Sportwettkämpfen, um bereits im Vorfeld ohne eine entsprechende Abdeckung durch Testdaten Probleme erkennen und beheben zu können.

## 1.4 Stand der Technik

Die dreidimensionale Lagebestimmung von Objekten lässt sich als Problemstellung zum Fachbereich der Physik, der Geodäsie, der Navigation oder auch der Astronomie zählen, sowohl in Abhängigkeit der maximalen Reichweite als auch der entsprechenden Messgenauigkeit. Die Anwendung im Sport zählt von den beobachteten Dimensionen her zur Geodäsie. Etablierte Methoden haben allerdings unterschiedliche Nachteile (vgl. Lepetit und Fua 2005b, Seite 4): Mechanische Ansätze bieten zwar hohe Genauigkeit, schränken jedoch die Reichweite sehr stark ein. Magnetische Verfahren sind sehr störanfällig gegenüber Metallen in der Umgebung, die sehr häufig in Messsituationen anzutreffen sind und haben darüber hinaus nur eine geringe Reichweite. Die Verwendung von Ultraschall ist auf große Entfernungen unpräzise, insbesondere da diese Messmethode abhängig von Schwankungen in der Umgebungstemperatur ist. Inertialsensoren leiden unter einer gewissen Drift über längere Zeiträume. Elektromagnetische Systeme sind zwar sehr genau, stellen aber massive Anforderungen an die Infrastruktur und gegebenenfalls an die zur Verfügung stehende Verarbeitungsleistung. Sowohl Satellitenortung als auch bodengestützte Systeme wie LPM<sup>1</sup> erwiesen sich in der praktischen Anwendung bislang als unzureichend, da sie entweder wie im ersten Fall sowohl örtlich als auch zeitlich zu ungenau waren oder ebenfalls zu hohe Anforderungen an die Infrastruktur vor Ort stellten.

Bildgestützte Messverfahren legen das Fachgebiet der Fernerkundung nahe, allerdings ist

---

<sup>1</sup> Local Position Measurement ist ein patentiertes System zur lokalen Positionsmessung in Echtzeit der ABATEC Electronic AG

hierfür im Allgemeinen ein kalibriertes Stereokamerasystem notwendig. Ein ähnliches System wurde bereits von Nachbauer u. a. (1996) vorgeschlagen, es setzt jedoch neben zwei Kameras, die den gleichen Bereich abdecken, fest vermessene und sichtbare Kontrollpunkte auf der Geländefläche voraus. Mauthner und Bischof (2007) bestimmten bereits den Fußpunkt eines Volleyballspielers über die direkte Transformation der Bildebene auf das Spielfeld.

## 1.5 Neuerungen

Die Überlegung und wesentliche Neuerung, die dieser Diplomarbeit zu Grunde liegt, ist, dass die allgemeine Rekonstruktion der Tiefeninformation durch die Annahme einer Bewegung auf einer beliebig modellierbaren Fläche eingeschränkt werden kann. Daher ist es auch bei Verwendung von nur einer Kamera möglich, die dreidimensionale Position des verfolgten Objektes wiederzugewinnen, sofern der Fußpunkt desselben ermittelt werden kann. Dadurch ist eine Positionsbestimmung mit sehr geringem Aufwand möglich, die auch in Echtzeit erfolgen kann, sofern die Punktkorrespondenzen bereits im Vorfeld ermittelt werden.

## 1.6 Aufbau dieser Diplomarbeit

Die Arbeit gliedert sich in vier Kapitel, deren ersten drei die Hauptkomponenten des Systems beschreiben. Der letzte Abschnitt zeigt schließlich die Integration dieser Komponenten sowie die praktische Anwendbarkeit an einem realen Einsatzbeispiel. Aufgrund des breitgefächerten Themas wird der wissenschaftliche Forschungsschwerpunkt auf die ersten beiden Hauptkomponenten gelegt, während die Ausführungen zur Objekterkennung lediglich einen Ausgangspunkt für weiterführende Arbeiten darstellen.

### **Kapitel 2: Kamerakalibrierung**

Integraler Bestandteil dieser Arbeit ist es, Welt- und Bildkoordinatensystem in eine Beziehung zueinander zu setzen. Erst durch die mathematische Formulierung eines entsprechenden Kameramodells ist das Messen aus Bildern überhaupt möglich, wenngleich nur Winkel bestimmt werden können, nicht aber absolute Entfernungen.

### **Kapitel 3: Tiefenrekonstruktion**

Ausgehend von Kapitel 2 kann sodann die Tiefeninformation rückgewonnen werden. Zu diesem Zweck wird sowohl Kamera als auch Geländefläche virtuell nachgebildet. Die unterschiedlichen Methoden zur Modellierung werden in diesem Abschnitt untersucht, um schließlich aus dem Schnittpunkt zwischen dem Sichtstrahl eines Bildpunktes und dem Modell der Geländefläche die dreidimensionale Position zu bestimmen.

### **Kapitel 4: Objekterkennung**

Ist die Zuordnung zwischen Bild- und Weltpunkten erfolgt, muss das zu verfolgende Objekt im Bild erkannt und seine Position bestimmt werden. Dieser Prozess wird als Objekter-

kennung bezeichnet und stellt ein sehr weites Forschungsgebiet dar, das an dieser Stelle hinsichtlich der Laufzeitoptimierung für diesen speziellen Anwendungsfall beleuchtet wird. Es werden einige einfache und sehr schnelle Methoden vorgestellt, mit denen eine stabile Objekterkennung realisiert werden kann.

### **Kapitel 5 und 6: Integration, Zusammenfassung und Ausblick**

Schließlich wird im letzten Kapitel die Integration der vorgestellten Komponenten behandelt, wobei näher auf die tatsächliche Anwendung eingegangen und dabei auftretende Probleme aufgezeigt werden. Daraus ergeben sich Optimierungsmöglichkeiten und ein Ausblick auf zukünftige Forschungsaktivitäten.

## Kapitel 2

# Kamerakalibrierung

Um aus Bildern messen zu können, benötigt man nähere Informationen über die verwendete Kamera, über deren inneren Aufbau und deren Position und Blickrichtung in der aufgenommenen Szene. Man spricht hierbei auch von der inneren und äußeren Orientierung der Kamera. Es gibt mehrere mathematische Modelle, die eine reale Kamera mit unterschiedlicher Genauigkeit abbilden und sich demnach auch wesentlich in der Anzahl der modellierten Parameter unterscheiden. Für die gestellte Aufgabe ist allerdings nur ein Modell sinnvoll, das auch wieder auf der gleichen Abstraktionsebene im modellbildenden Prozess abgebildet werden kann. Eine Entscheidung darüber zu treffen ist insofern schwierig, als für die Visualisierung entscheidende Vereinfachungen getroffen werden, die nicht mit realen Eigenschaften vereinbar sind. Ist ein solches Modell dennoch gefunden, lässt sich ein Bildpunkt in einen Strahl im Objektraum überführen. Den Weg dorthin beschreibt dieses Kapitel. Für die mathematische Beweisführung der gezeigten Herleitungen sei auf Hartley und Zisserman (2004, Part 0) verwiesen, die Nomenklatur ist an dieses Standardwerk bewusst angelehnt und diesem auch, sofern nicht anders gekennzeichnet, entnommen.

### 2.1 Kameramodelle

Unter dem Terminus Kamera versteht man in mathematischem Sinne eine Zuordnung zwischen einer 3D Szene, dem Objektraum (Welt) und einem zugeordneten 2D Bild, dem Bildraum. Grundsätzlich lässt sich diese Zuordnung als Transformation (Translation, Rotation) gefolgt von einer Projektion darstellen. Im Folgenden werden nun zwei bekannte Kameramodelle entwickelt, die einerseits der Photogrammetrie, also der Rückgewinnung von Welt- aus Bildkoordinaten, und andererseits der Computergraphik, also der Umwandlung von Welt in Bildkoordinaten, entstammen. Die Problematik liegt allerdings in der Überführung der beiden Modelle ineinander, denn es gibt zwar viele Lösungen aus dem Bereich der virtuellen Realität, die sich aber entweder auf Stereokamerasysteme beziehen, oder aber für diese Problemstellung unvollständig sind.

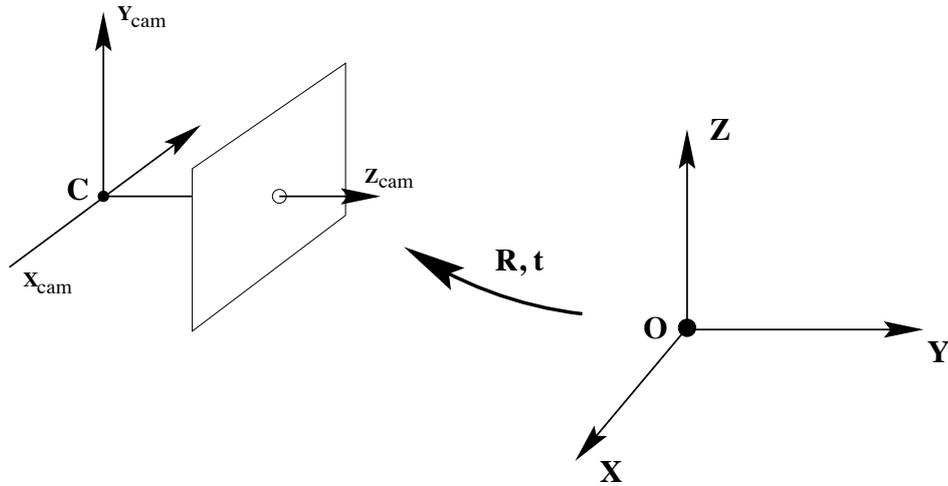


Abbildung 2.1: Äußere Orientierung einer Kamera (Hartley und Zisserman 2004, Fig. 5.3)

### 2.1.1 Orientierung der Kamera

#### Äußere Orientierung

Die äußere Orientierung einer Kamera bezeichnet die Position und Rotation der Kamera im Objektraum. In Matrixschreibweise lässt sich diese affine Transformation der Weltkoordinaten eines Punktes in Kamerakoordinaten wie folgt darstellen:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} \quad (2.1)$$

$$\mathbf{X}_c = [R | \mathbf{t}] \mathbf{X}_w \quad (2.2)$$

#### Innere Orientierung

Die innere Orientierung der Kamera beschreibt die Projektion aus dem Kamerakoordinatensystem auf die Bildebene. Im einfachsten Fall, dem Lochkameramodell, ist dies eine zentrale Projektion in euklidischen Bildkoordinaten mit der Brennweite  $f$ .

Unter Verwendung von homogenen Koordinaten lässt sie sich anschreiben zu

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} fx_w \\ fy_w \\ z_w \end{bmatrix} \quad (2.3)$$

Verkürzt ergibt sich die Matrix der inneren Orientierung zu

$$K_{cp} = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} \quad (2.4)$$

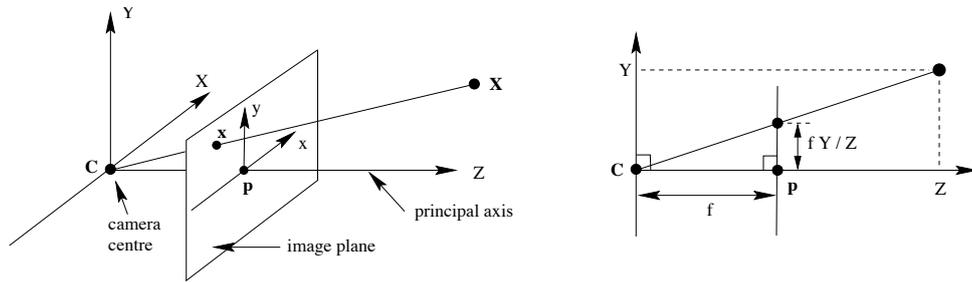


Abbildung 2.2: Innere Orientierung der Lochkamera (Hartley und Zisserman 2004, Fig. 6.1)

Die erste Erweiterung dieses Modells führt den Bildhauptpunkt ein, der den Schnittpunkt der optischen Achse der Kamera mit der Bildebene darstellt. Auf die Bedeutung des Bildhauptpunktes und auf die Probleme bei seiner Berechnung wird in der Diskussion der Ergebnisse am Ende des Kapitels detaillierter eingegangen werden.

$$K_{cp} = \begin{bmatrix} f & x_0 \\ & f & y_0 \\ & & 1 \end{bmatrix} \quad (2.5)$$

Berücksichtigt man nun die unterschiedlichen Dimensionen der Bildebene, also die unterschiedlichen Skalierungen in x- und y-Richtung (repräsentiert durch  $m_x$  und  $m_y$  beziehungsweise das Seitenverhältnis  $\alpha$ ), die durch das Format des Bildsensors gegeben sind, und führt man als zusätzliche Verallgemeinerung noch den Winkel zwischen den beiden Bildachsen ein (durch den Skew-Parameter  $\gamma$ ), so erhält man folgende Matrix für die interne Orientierung der Kamera:

$$K = \begin{bmatrix} f' m_x & \gamma & x_0 \\ & f' m_y & y_0 \\ & & 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{\alpha} & \gamma & x_0 \\ & f & y_0 \\ & & 1 \end{bmatrix} \quad (2.6)$$

Die Einführung dieser Parameter ist zwar notwendig, um eine reale Kamera abzubilden, macht allerdings Probleme bei der Überführung in ein computergraphisches Modell.

### 2.1.2 Photogrammetrie

Photogrammetrie bezeichnet die Rekonstruktion der Lage und Form von Objekten aus Bildern (vgl. Kraus 1998). Zu diesem Zweck muss die innere und äußere Orientierung der Kamera bekannt sein, um Winkel zwischen Sichtstrahlen messen zu können. Die gesamte Abbildung schließt unter Vernachlässigung von Linsenverzeichnungen die beschriebenen Parameter der Translation, Rotation und Projektion ein, womit sich für die Transformation eines Punktes im Objektraum in den Bildbereich unter Berücksichtigung der Kameraposition  $\tilde{\mathbf{C}}$  folgende Gleichung anschreiben lässt.

$$\mathbf{X}_c = K [R | \mathbf{t}] \mathbf{X}_w = KR [I | -\tilde{\mathbf{C}}] \mathbf{X}_w \quad (2.7)$$

### Position und Translation

Die Kameraposition entspricht dem rechten Nullraum, dem Kern der Gesamtprojektionsmatrix  $P = K [R | \mathbf{t}]$ , es muss also gelten  $P\mathbf{C} = \mathbf{0}$ . Dies begründet sich algebraisch daraus, dass die Matrix  $P$  eine  $3 \times 4$  Matrix ist, sie aber obgleich ihrer 4 Spaltenvektoren nur Rang 3 besitzt. Geometrisch entspricht dies der Wahl des Koordinatensystems. Da  $\mathbf{t}$  die Translationskomponente der äußeren Orientierung darstellt und direkt aus der Gesamtprojektion ablesbar ist, kann die um die Rotation bereinigte Kameraposition  $\tilde{\mathbf{C}}$  auch explizit ausgedrückt werden.

$$\tilde{\mathbf{C}} = -R^T \mathbf{t} \quad (2.8)$$

$\tilde{\mathbf{C}}$  bezeichnet die bereits normierte Kameraposition. Algebraisch lässt sich  $\mathbf{C}$  auch als Schnittpunkt der von  $P$  aufgespannten Ebenen ausdrücken,  $\mathbf{p}_i$  bezeichnet hier den  $i$ -ten Spaltenvektor von  $P$  (vgl. Hartley und Zisserman 2004, S. 67, 3.5).

$$\mathbf{C} = \begin{bmatrix} \det([\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4]) \\ -\det([\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_4]) \\ \det([\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4]) \\ -\det([\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]) \end{bmatrix} \quad (2.9)$$

### Rotation

Die linke  $3 \times 3$  Matrix von  $P$  beschreibt die Rotation der Kamera. Es handelt sich um eine Drehmatrix, die damit sowohl orthogonal (das heißt, sie erhält sowohl Länge als auch Winkel zwischen Vektoren, auf die sie angewendet wird) als auch orientierungserhaltend ist. Ihre Determinante ist eins. Für die Modellbildung als auch für nichtlineare Optimierungen ist jedoch eine Parametrisierung zur Reduktion der Variablen notwendig. Hierzu gibt es unterschiedliche Darstellungsformen, deren älteste und bekannteste die in Form von Eulerschen Winkeln aus dem Jahre 1776 ist (vgl. Stuelpnagel 1964).

### Eulersche Winkel

Bei dieser Parametrisierung wird die Drehmatrix als Verkettung dreier Einzeldrehungen um drei aufeinander orthogonal stehende Achsen aufgefasst. Abgesehen von der mehrdeutigen Parametrisierung besteht der wesentliche Nachteil der Darstellung in Eulerschen Winkeln im Phänomen des Gimbal Lock<sup>1</sup>. Es tritt auf, wenn eine Achse um 90 Grad gedreht wird und dann mit einer anderen zusammenfällt, wodurch es zu einer Reduktion der Freiheitsgrade kommt. Dieses Problem ist vor allem in der Computeranimation von großer Bedeutung, erweist sich allerdings auch hier insofern als hinderlich, als es in Grenzfällen die numerische Optimierung beeinträchtigt.

### Exponentialabbildung

Anstatt drei einzelner aufeinanderfolgender Rotationen kann die Gesamtdrehung auch als *eine* Drehung um eine bestimmte Achse beschrieben werden. Diese lässt sich durch einen Vektor  $\mathbf{a}$  darstellen, dessen Orientierung die Drehachse und dessen Länge den Drehwinkel

---

<sup>1</sup> vgl. Grassia 1998, Seite 3.

beschreibt. Gleichung 2.10 zeigt die so genannte Rodrigues<sup>2</sup> Formel zur Erzeugung einer antisymmetrischen  $3 \times 3$  Matrix, die in Matrixdarstellung ein Kreuzprodukt beschreibt.

$$A = \exp([\mathbf{a}]_{\times}) = I + \sin(\|\mathbf{a}\|) \left[ \frac{\mathbf{a}}{\|\mathbf{a}\|} \right]_{\times} + (1 - \cos(\|\mathbf{a}\|)) \left[ \frac{\mathbf{a}}{\|\mathbf{a}\|} \right]_{\times}^2 \quad (2.10)$$

Die Matrix des Kreuzproduktes lautet hierbei für einen dreidimensionalen Vektor  $\mathbf{v}$ :

$$[\mathbf{v}]_{\times} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \quad (2.11)$$

Lässt sich auf diese Weise die Rotationsmatrix aufstellen, kann aus dieser auch wieder Achse und Winkel gewonnen werden. Der Drehwinkel lässt sich über die Spurabbildung<sup>3</sup> der Matrix wieder gewinnen. Für eine  $3 \times 3$  Rotationsmatrix  $V$  gilt:

$$\text{Spur}(A) = 1 + 2 \cos(\phi) \rightarrow \phi = \arccos\left(\frac{\text{Spur}(A) - 1}{2}\right) \quad (2.12)$$

$$\mathbf{a} = \frac{\phi}{2 \sin(\phi)} \begin{bmatrix} A_{3,2} - A_{2,3} \\ A_{1,3} - A_{3,1} \\ A_{2,1} - A_{1,2} \end{bmatrix} \quad (2.13)$$

### 2.1.3 Virtuelle Modellierung

In Anwendungen der Computergraphik wird die Gesamtprojektion in meistens drei Matrizen aufgespaltet: Welt-, Ansichts- (oder Betrachtungs-) und Projektionsmatrix. Erstere transformiert nur das Modell selbst im Weltkoordinatensystem. Da im gegebenen Anwendungsfall aber nur ein Modell existiert und es nur auf die relative Position der Kamera zu diesem ankommt, soll hier die Weltmatrix der Identitätsmatrix entsprechen. Somit ist die relative Transformation zwischen Kamera und Modell ausschließlich in der Betrachtungsmatrix lokalisiert, welche die Transformation der Weltkoordinaten in das Kamerakoordinatensystem angibt. Da das der realen Kamera angenäherte Lochkameramodell keine orthographische, sondern eine perspektivische Projektion auf die Bildebene darstellt, muss zuletzt noch diese Transformation in Form der Projektionsmatrix durchgeführt werden. Geometrisch wird dadurch der Sichtwürfel in einen Pyramidenstumpf umgewandelt, der auch als Frustum (siehe Abbildung 2.3) bezeichnet wird. Die erwähnten Matrizen bilden dreidimensionale, homogene Koordinaten auf eben solche ab; es handelt sich also um  $4 \times 4$  Matrizen. Dies ergibt einen wesentlichen Unterschied zum photogrammetrischen Modell, bildet dies ja dreidimensionale homogene auf zweidimensionale homogene Koordinaten ab und entspricht somit nur einer  $3 \times 4$  Matrix.

#### Die Betrachtungsmatrix

Diese Matrix gibt die Translation und Rotation der Kamera gegenüber dem Weltkoordinatensystems an. Sie wird gewöhnlich aus drei Vektoren gebildet, nämlich der Kameraposition  $\tilde{\mathbf{C}}$  selbst, dem Blickpunkt der Kamera  $\mathbf{\Lambda}$  und dem vertikalen Bildvektor  $\mathbf{\Upsilon}$ . Kennt

<sup>2</sup> vgl. Rodrigues 1840.

<sup>3</sup> Die Spur einer quadratischen Matrix ist die Summe ihrer Diagonalelemente

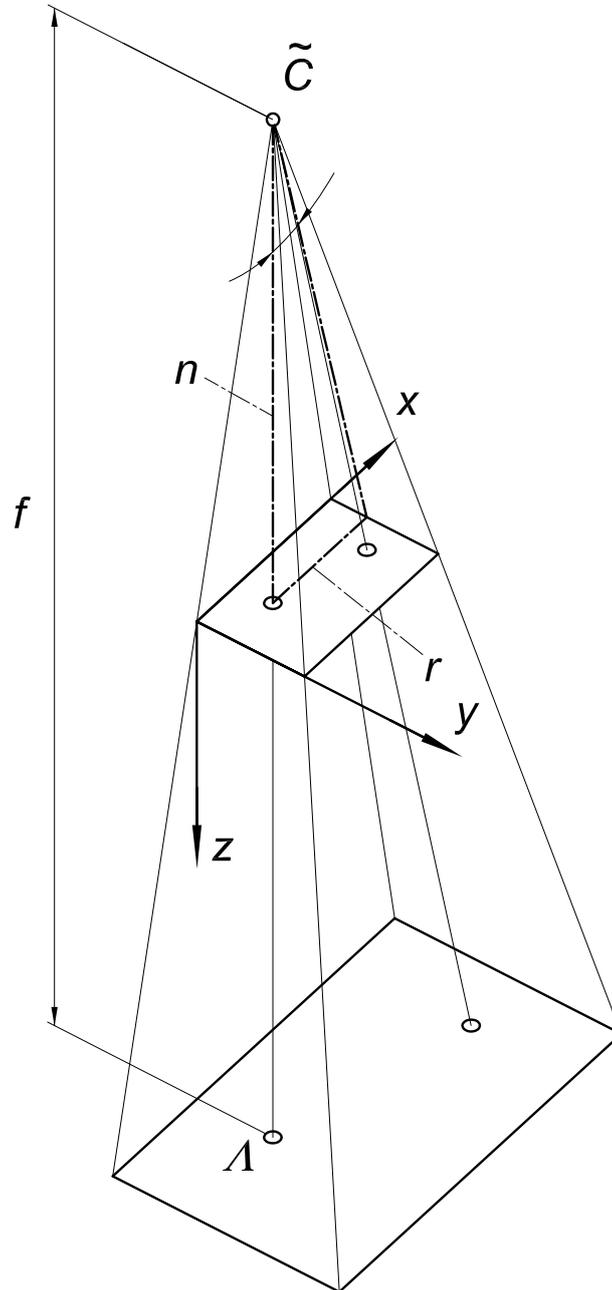


Abbildung 2.3: Abbildungsfrustum

man diese Vektoren, lassen sich zunächst die Achsen des Koordinatensystems definieren, hier für ein rechtshändiges<sup>4</sup> Koordinatensystem:

$$\mathbf{\Gamma}_z = \frac{\tilde{\mathbf{C}} - \mathbf{\Lambda}}{\|\tilde{\mathbf{C}} - \mathbf{\Lambda}\|} \quad (2.14)$$

$$\mathbf{\Gamma}_x = \frac{\mathbf{\Upsilon} \times \mathbf{\Gamma}_z}{\|\mathbf{\Upsilon} \times \mathbf{\Gamma}_z\|} \quad (2.15)$$

$$\mathbf{\Gamma}_y = \mathbf{\Gamma}_z \times \mathbf{\Gamma}_x \quad (2.16)$$

Unter Verwendung der so definierten Vektoren erhält man die Betrachtungsmatrix  $G_{Ansicht}$  wie folgt:

$$G_{Ansicht} = \begin{bmatrix} \Gamma_{x,x} & \Gamma_{x,y} & \Gamma_{x,z} & \langle \mathbf{\Gamma}_x, \tilde{\mathbf{C}} \rangle \\ \Gamma_{y,x} & \Gamma_{y,y} & \Gamma_{y,z} & \langle \mathbf{\Gamma}_y, \tilde{\mathbf{C}} \rangle \\ \Gamma_{z,x} & \Gamma_{z,y} & \Gamma_{z,z} & \langle \mathbf{\Gamma}_z, \tilde{\mathbf{C}} \rangle \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.17)$$

## Die Projektionsmatrix

In Analogie zum photographischen Modell entspricht die Projektionsmatrix der inneren Orientierung der Kamera, weshalb sie für die Überführung besondere Bedeutung hat. Die Einträge dieser Matrix definieren das Sichtvolumen an Hand seiner Begrenzungsflächen. Bei symmetrischen Verhältnissen reichen die vertikalen und horizontalen Ausdehnungen von -1 bis 1, die Ausdehnung in der z-Achse je nach verwendetem System entweder von 0 bis 1 (DirectX) oder ebenfalls von -1 bis 1 (OpenGL). Dies bedeutet, dass das gesamte Sichtfeld der Kamera in normierten Kamerakoordinaten abgebildet und zusätzlich auch die Tiefeninformation in der Projektion als z-Achse berücksichtigt wird, da die Normierung auf die homogene Koordinate erfolgt. Für ein rechtshändiges<sup>5</sup> Koordinatensystem entspricht die Projektionsmatrix Gleichung 2.18.

$$G_{Projektion} = \begin{bmatrix} \frac{2 \text{ near}}{\text{right-left}} & 0 & \frac{\text{left+right}}{\text{right-left}} & 0 \\ 0 & \frac{2 \text{ near}}{\text{top-bottom}} & \frac{\text{top+bottom}}{\text{top-bottom}} & 0 \\ 0 & 0 & \frac{\text{far}}{\text{near-far}} & \frac{\text{near far}}{\text{near-far}} \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad (2.18)$$

### 2.1.4 Überführung der Modelle

Computergraphische Modelle verwenden im Bildraum homogene Koordinaten. Zu den zwei Bildkoordinaten kommt eine dritte hinzu, die die Tiefeninformation im Betrachtungsvolumen enthält. Es erfolgt also eine Abbildung eines vierdimensionalen auf einen, nach der Normierung dreidimensionalen Vektor unter Verwendung einer  $4 \times 4$  Matrix. Diese setzt sich unter Vernachlässigung einer etwaigen Welttransformation aus Gleichung 2.18 und

<sup>4</sup> Für ein linkshändiges Koordinatensystem ergibt sich die z-Achse invertiert zu  $\mathbf{\Gamma}_z = \frac{\mathbf{\Lambda} - \tilde{\mathbf{C}}}{\|\mathbf{\Lambda} - \tilde{\mathbf{C}}\|}$ , der Rest bleibt unverändert.

<sup>5</sup> Für ein linkshändiges Koordinatensystem ist die z-Achse invertiert und somit der Eintrag  $G_{4,3} = 1$

2.17 zu  $G = G_{\text{Projektion}} \cdot G_{\text{Ansicht}}$  zusammen und ist zum besseren Vergleich noch einmal in Gleichung 2.19<sup>6</sup> dargestellt.

Im Bildraum einer realen Kamera ist allerdings diese Tiefeninformation nicht mehr enthalten, womit es in der linearen Abbildung in homogenen Koordinaten zu einer Reduktion auf eine  $4 \times 3$  Matrix kommt, die sich aus Translation, Rotation und Projektion zusammensetzt. Unter Verwendung der gleichen Nomenklatur für die Rotation und Translation und nach Multiplikation mit Gleichung 2.6 ergibt sich die Gesamtabbildung des realen Kameramodells zu Gleichung 2.20.

$$G = \begin{bmatrix} \frac{\Gamma_{z,x}(l+r)}{-l+r} + \frac{2n\Gamma_{x,x}}{-l+r} & \frac{\Gamma_{z,y}(l+r)}{-l+r} + \frac{2n\Gamma_{x,y}}{-l+r} & \frac{\Gamma_{z,z}(l+r)}{-l+r} + \frac{2n\Gamma_{x,z}}{-l+r} & \frac{2\langle \mathbf{\Gamma}_x, \tilde{\mathbf{C}} \rangle n}{-l+r} + \frac{\langle \mathbf{\Gamma}_z, \tilde{\mathbf{C}} \rangle (l+r)}{-l+r} \\ \frac{\Gamma_{z,x}(b+t)}{-b+t} + \frac{2n\Gamma_{y,x}}{-b+t} & \frac{\Gamma_{z,y}(b+t)}{-b+t} + \frac{2n\Gamma_{y,y}}{-b+t} & \frac{\Gamma_{z,z}(b+t)}{-b+t} + \frac{2n\Gamma_{y,z}}{-b+t} & \frac{2\langle \mathbf{\Gamma}_y, \tilde{\mathbf{C}} \rangle n}{-b+t} + \frac{\langle \mathbf{\Gamma}_z, \tilde{\mathbf{C}} \rangle (b+t)}{-b+t} \\ \frac{\Gamma_{z,x}(-f-n)}{f-n} & \frac{\Gamma_{z,y}(-f-n)}{f-n} & \frac{\Gamma_{z,z}(-f-n)}{f-n} & \frac{\langle \mathbf{\Gamma}_z, \tilde{\mathbf{C}} \rangle (-f-n)}{f-n} - \frac{2fn}{f-n} \\ -\Gamma_{z,x} & -\Gamma_{z,y} & -\Gamma_{z,z} & -\langle \mathbf{\Gamma}_z, \tilde{\mathbf{C}} \rangle \end{bmatrix} \quad (2.19)$$

$$P = \begin{bmatrix} \Gamma_{z,x} x_0 + \frac{f\Gamma_{x,x}}{\alpha} & \Gamma_{z,y} x_0 + \frac{f\Gamma_{x,y}}{\alpha} & \Gamma_{z,z} x_0 + \frac{f\Gamma_{x,z}}{\alpha} & \frac{\langle \mathbf{\Gamma}_x, \tilde{\mathbf{C}} \rangle f}{\alpha} - \langle \mathbf{\Gamma}_z, \tilde{\mathbf{C}} \rangle x_0 \\ \Gamma_{z,x} y_0 + f\Gamma_{y,x} & \Gamma_{z,y} y_0 + f\Gamma_{y,y} & \Gamma_{z,z} y_0 + f\Gamma_{y,z} & \langle \mathbf{\Gamma}_y, \tilde{\mathbf{C}} \rangle f - \langle \mathbf{\Gamma}_z, \tilde{\mathbf{C}} \rangle y_0 \\ \Gamma_{z,x} & \Gamma_{z,y} & \Gamma_{z,z} & \langle \mathbf{\Gamma}_z, \tilde{\mathbf{C}} \rangle \end{bmatrix} \quad (2.20)$$

Durch Koeffizientenvergleich sind folgende Zusammenhänge ersichtlich:

$$\begin{aligned} x_0 &= \frac{r+l}{r-l} \\ y_0 &= \frac{t+b}{t-b} \\ \frac{2n}{r-l} &= \frac{f}{\alpha} \hat{=} f' m_x \\ \frac{2n}{t-b} &= f \hat{=} f' m_y \end{aligned} \quad (2.21)$$

Des Weiteren lässt sich die Umkehrung der z-Achse an der jeweils homogenen Koordinate erkennen. Im Bereich der Computergraphik entsteht das virtuelle Bild der Kamera vor dem Brennpunkt, bei einer realen Kamera physikalisch bedingt hinter dem Brennpunkt. Diese Inversion spiegelt sich auch in den Projektionsmatrizen der zwei Modelle wieder: Gleichung 2.6 zeigt einen positiven Eintrag für die z-Komponente, Gleichung 2.18 hingegen einen negativen. Damit geht natürlich nach der Normierung auch eine punktsymmetrische Spiegelung des Bildes um den Bildhauptpunkt, das heißt um den Schnittpunkt mit der optischen Achse einher. Abbildung 2.4 zeigt die geometrische Entsprechung der beiden Projektionen im Vergleich.

<sup>6</sup> Zur übersichtlicheren Darstellung werden folgende Abkürzungen vorgenommen: n = near, f = far, l = left, r = right, t = top, b = bottom

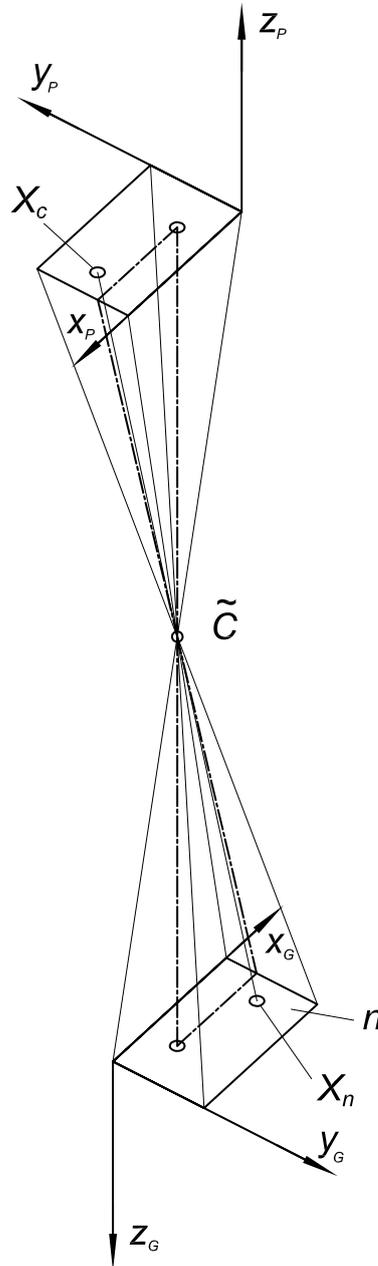


Abbildung 2.4: Zusammenhang zwischen computergraphischer und photogrammetrischer Projektion:

Die Bildpunkte  $X_c$  (im Koordinatensystem  $x_p, y_p, z_p$ ) der realen Kamera und  $X_n$  (im Koordinatensystem  $x_g, y_g, z_g$ ) der computergraphischen Abbildung sind punktsymmetrisch zum gemeinsamen Brennpunkt  $\tilde{C}$

## 2.2 Berechnung der Kameraparameter

Ausgehend von einer Reihe von  $i$  Punktkorrespondenzen  $\mathbf{X}_{c,i} \leftrightarrow \mathbf{X}_{w,i}$  zwischen Bild- und Objektraum lässt sich eine Matrix  $P$  bilden, für die gilt

$$\mathbf{X}_{c,i} = P \mathbf{X}_{w,i} \quad \forall i \quad (2.22)$$

Zum Gewinnen dieser Matrix gibt es mehrere Möglichkeiten, die sich grundlegend in direkte und iterative Ansätze unterscheiden lassen. Es ist jedoch zu bemerken, dass letztere immer auch eine mehr oder minder gute Schätzung als Startwert benötigen, damit die iterative Lösung konvergiert. Aus diesem Grund wird oft ein direktes Verfahren zur Bestimmung des Startwertes in einem ersten Schritt eingesetzt. Tabelle 2.1 zeigt einen Überblick über einige Kamerakalibrierungsmodelle und deren Eigenschaften.

Modell	Parameter	Vorteile	Nachteile
Lochkamera	7	linear, einfach, stabil	modelliert keine reale Kamera
DLT <sup>a</sup>	11	linear, stabil, benötigt 6 Punktkorrespondenzen (3D)	modelliert keine nichtlinearen Verzerrungen
CDLT (koplanare DLT)	11	linear, stabil, benötigt 6 Punktkorrespondenzen (2D)	modelliert keine nichtlinearen Verzerrungen, benötigt initiale Schätzung einiger Parameter
Extended DLT	14-18	modelliert Linsenverzerrungen	iterative Konvergenz oder nichtlineare Suche
Traditionell photogrammetrisch	14+	exakte Ergebnisse	nichtlineare Suche, rechenintensiv, präzises Kalibrierobjekt notwendig
Vereinfachung nach Tsai	11-16	gute Ergebnisse, Source Code frei verfügbar	nichtlineare Suche, Abhängig von der Genauigkeit der Daten

<sup>a</sup> Direkte lineare Transformation

Tabelle 2.1: Kamerakalibrierungsmodelle (vgl. Bakstein 1999)

Da koplanare Modelle<sup>7</sup> auf Grund der Verallgemeinerung auf beliebige Geländeflächen generell ausscheiden, gilt es, zwischen dem einfachen Lochkameramodell, der direkten linearen Transformation (DLT) sowie ihren Erweiterungen und dem traditionellen photogrammetrischen Modell mit den Vereinfachungen nach Tsai (1987) zu entscheiden. Dass

<sup>7</sup> Derartige Modelle bilden lediglich eine Ebene im Raum auf die Bildebene ab, nicht aber ein Volumen

das verwendete Modell in einem nachfolgenden Schritt in ein computergraphisches Kameramodell umgewandelt werden soll, das gewissen Einschränkungen unterworfen ist, muss bei der Wahl hinsichtlich der modellierten Kameraparameter berücksichtigt werden. Somit erweist sich die DLT als optimal, da sie alle notwendigen linearen Parameter beibehält, allerdings nichtlineare Erweiterungen vernachlässigt. In Hinblick auf die nichtlineare Optimierung lassen sich allerdings zusätzliche Bildverzeichnungen durch beispielsweise Linssenverzerrungen leicht ergänzen.

### 2.2.1 Direkte lineare Transformation (DLT)

Die direkte lineare Transformation (Abdel-Aziz und Karara 1971) ist ein Verfahren zur Lösung eines überbestimmten Gleichungssystems unter Minimierung des algebraischen Fehlers:

$$\mathbf{x}_k \propto A \mathbf{y}_k \text{ für } k = 1, \dots, N \quad (2.23)$$

Angewandt auf das Kalibrierungsproblem kann für jede Punktrelation der verfügbaren Kalibrierpunkte ein Gleichungssystem aufgestellt werden. Durch Umformung von Gleichung 2.22 unter Verwendung des Kreuzproduktes erhält man

$$\mathbf{X}_{c,i} \times P \mathbf{X}_{w,i} = \mathbf{0} \quad (2.24)$$

Dies ist insofern zulässig, als  $\mathbf{X}_{c,i}$  und  $P \mathbf{X}_{w,i}$ , abgesehen von einem Skalierungsfaktor  $w_i$ , in die gleiche Richtung zeigen. Setzt man nun Gleichung 2.25 ein, ergibt sich das ausmultiplizierte Kreuzprodukt zu Gleichung 2.26.

$$\mathbf{X}_{c,i} = \begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix} \quad (2.25)$$

$$\begin{bmatrix} \mathbf{0}^T & -w_i \mathbf{X}_{w,i}^T & y_i \mathbf{X}_{w,i}^T \\ w_i \mathbf{X}_{w,i}^T & \mathbf{0}^T & -x_i \mathbf{X}_{w,i}^T \\ -y_i \mathbf{X}_{w,i}^T & x_i \mathbf{X}_{w,i}^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} \stackrel{!}{=} \mathbf{0} \quad (2.26)$$

Dennoch sind nur zwei dieser drei Gleichungen linear unabhängig, da die dritte Zeile aus einer Linearkombination der ersten beiden gewonnen werden kann. Wählt man nun den Skalierungsfaktor  $w_i = 1$  und kombiniert die so aufgestellten Beziehungen von  $n$  Kalibrierpunkten, erhält man ein Gleichungssystem der Form  $A \mathbf{p} = \mathbf{0}$ , wobei  $A$  eine  $2n \times 12$  Matrix ist.

$$2n \left\{ \begin{bmatrix} 0 & 0 & 0 & 0 & -X_0 & -Y_0 & -Z_0 & -1 & y_0 X_0 & y_0 Y_0 & y_0 Z_0 & y_0 \\ X_0 & Y_0 & Z_0 & 1 & 0 & 0 & 0 & 0 & -x_0 X_0 & -x_0 Y_0 & -x_0 Z_0 & -x_0 \\ \vdots & \vdots \\ 0 & 0 & 0 & 0 & -X_n & -Y_n & -Z_n & -1 & y_n X_n & y_n Y_n & y_n Z_n & y_n \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -x_n X_n & -x_n Y_n & -x_n Z_n & -x_n \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{bmatrix} \stackrel{!}{=} \mathbf{0} \right. \quad (2.27)$$

### Lösung des Gleichungssystems

Das Gleichungssystem 2.27 hat, unter Vernachlässigung des beliebigen Skalierungsfaktors, 11 Freiheitsgrade. Das bedeutet, dass für eine exakte Lösung zumindest  $5\frac{1}{2}$  Kalibrierpunkt-korrespondenzen vorliegen müssen. Im allgemeinen Fall, insbesondere jedoch wenn die Korrespondenzpunkte mit einem gewissen Rauschen (Messunsicherheit) überlagert sind, wird dieses Gleichungssystem überbestimmt sein ( $\geq 6$  Kalibrierpunkte) und der Lösungsansatz entspricht der Minimierung des algebraischen Fehlers (Hartley und Zisserman 2004, S. 88 ff., S. 178 ff., S. 585 ff.).

$$\min_{\mathbf{p}} \frac{\|A\mathbf{p}\|}{\|\mathbf{p}\|} \quad (2.28)$$

Die triviale Lösung  $\mathbf{p} = \mathbf{0}$  wird durch die Nebenbedingung  $\|\mathbf{p}\| = 1$  ausgeschlossen. Das Ergebnis dieser Minimierungsaufgabe ist der Eigenvektor von  $A^T A$  mit dem kleinsten Eigenwert.

Der Lösungsvektor  $\mathbf{p} = [ \mathbf{p}_1 \ \mathbf{p}_2 \ \mathbf{p}_3 ]^T$  kann sodann umgeschrieben werden zu der Lösungsmatrix  $P$ , die die affine Transformation von Objekt- zu Bildraum darstellt.

$$P = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{bmatrix} \quad (2.29)$$

## 2.3 Dekomposition der Transformationsmatrix

Als Ergebnis der direkten linearen Transformation steht die gesamte Transformationsmatrix  $P$  zur Verfügung. Sowohl zur nichtlinearen Optimierung als auch zur späteren Überführung in ein computergraphisches Kameramodell ist es notwendig, die inneren und äußeren Kameraparameter aus dieser Matrix zu gewinnen.

Zu diesem Zweck lassen sich die Eigenschaften der jeweiligen Matrixdarstellung heranziehen, da, wie bereits erwähnt, die innere Kalibrierung einer oberen Dreiecksmatrix und der linke Teil der äußeren Kalibrierung einer Rotationsmatrix entspricht. Letztere zeichnet sich

durch die Eigenschaften einer orthogonalen Matrix aus: Die Zeilenvektoren spannen eine Orthonormalbasis auf und repräsentieren gleichzeitig Rotationsachsen. Somit gilt auch  $Q^T = Q^{-1}$  und  $Q^T Q = Q Q^T = I$ . Die Spaltenvektoren repräsentieren die Abbildungen der Fluchtpunkte im Bild.

Ausgehend von der linksseitigen  $3 \times 3$  Teil-Matrix von  $P$  lässt sich nun folgende Zerlegung durchführen

$$P = [M \mid -M\tilde{C}] = K [R \mid -R\tilde{C}] \quad (2.30)$$

woraus direkt folgt

$$M = KR \quad (2.31)$$

Das Problem erinnert stark an die QR-Dekomposition, allerdings ist hier  $K$  die obere Dreiecksmatrix und  $R$  die Orthogonale, sodass es sich korrekterweise um eine RQ-Dekomposition handelt. Da diese selten direkt verfügbar ist, soll im Folgenden näher darauf eingegangen werden.

### 2.3.1 RQ-Dekomposition

Um die RQ-Dekomposition durchzuführen gibt es mehrere Möglichkeiten. Ähnlich der QR-Dekomposition lassen sich das Gram-Schmidtsche Orthogonalisierungsverfahren, Householdertransformationen oder Givens-Rotationen anwenden. Des Weiteren ist es auch möglich, die RQ-Zerlegung durch Invertieren auf eine QR-Zerlegung zurückzuführen. Da sich das Verfahren mit Givens-Rotationen im vorliegenden Fall insofern anbietet, als dass die Rotationswinkel und -matrizen des Verfahrens direkt dem Rotationsteil der äußeren Kalibrierung der Kamera entsprechen, wird eben dieses hier näher behandelt.

#### Givens-Rotationen

Givens-Rotationen (nach Wallace Givens) (vgl. Press u. a. 1992) bzw. Jacobi-Rotationen (nach Carl Gustav Jacobi) stellen Rotationen in einer Ebene dar, die durch je zwei Koordinatenachsen aufgespannt werden. Für den dreidimensionalen Fall lauten die Rotationsmatrizen um die jeweiligen Achsen wie folgt:

$$Q_x = \begin{bmatrix} 1 & & \\ & \cos(\theta) & -\sin(\theta) \\ & \sin(\theta) & \cos(\theta) \end{bmatrix} Q_y = \begin{bmatrix} \cos(\theta) & \sin(\theta) & \\ & 1 & \\ -\sin(\theta) & \cos(\theta) & \end{bmatrix} Q_z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & \\ \sin(\theta) & \cos(\theta) & \\ & & 1 \end{bmatrix} \quad (2.32)$$

An der Struktur der Drehmatrizen kann man erkennen, dass bei der Multiplikation mit einer solchen je nach Achse jeweils nur zwei der drei Spalten mit einer Linearkombination ihrer selbst ersetzt werden, während eine Spalte unverändert bleibt. Diese Eigenschaft lässt sich nun nutzen, um Schritt für Schritt beliebige Einträge der Ausgangsmatrix null zu setzen. Möchte man beispielsweise von einer  $3 \times 3$  Matrix  $K$  den Eintrag  $k_{32}$  über eine Rotation um die  $x$ -Achse null setzen, so ergibt sich die Berechnung des Winkels zu

$$K' = K Q_x = \begin{bmatrix} k_{11} & k_{12} \cos(\theta) + k_{13} \sin(\theta) & k_{13} \cos(\theta) - k_{12} \sin(\theta) \\ k_{21} & k_{22} \cos(\theta) + k_{23} \sin(\theta) & k_{23} \cos(\theta) - k_{22} \sin(\theta) \\ k_{31} & k_{32} \cos(\theta) + k_{33} \sin(\theta) & k_{33} \cos(\theta) - k_{32} \sin(\theta) \end{bmatrix}$$

$$k_{32} \cos(\theta) + k_{33} \sin(\theta) \stackrel{!}{=} 0$$

$$\begin{aligned} \sin(\theta) &= -\frac{k_{32}}{\sqrt{k_{32}^2 + k_{33}^2}} \quad \wedge & \cos(\theta) &= \frac{k_{33}}{\sqrt{k_{32}^2 + k_{33}^2}} \quad \vee \\ \sin(\theta) &= \frac{k_{32}}{\sqrt{k_{32}^2 + k_{33}^2}} \quad \wedge & \cos(\theta) &= -\frac{k_{33}}{\sqrt{k_{32}^2 + k_{33}^2}} \end{aligned} \quad (2.33)$$

Es ist anzumerken, dass dieses Verfahren daher auch für unterschiedliche Matrixzerlegungen angewendet werden kann.

### Mehrdeutigkeiten

Unter Berücksichtigung der Bedingungen zum Nullsetzen der Matrixeinträge (siehe Gleichung 2.33) ist ersichtlich, dass je Eintrag zwei mögliche Lösungen existieren. Hartley und Zisserman (2004) empfehlen, diese Mehrdeutigkeit unter der Bedingung aufzulösen, dass alle Diagonaleinträge der oberen Dreiecksmatrix positiv sind. Dies ist allerdings für den Fall des computergraphischen Modells, also dass die  $y$ -Achse des Bildes nach unten und demnach die  $z$ -Achse in die Bildebene hinein zeigt, nicht der Fall. Für diese Konfiguration gibt es nur Lösungen, für die alle Diagonaleinträge negativ sind. Um dennoch das beschriebene Modell weiter verwenden zu können, lässt sich eine Korrekturmatrix folgender Struktur erstellen:

$$\zeta(K) = \begin{bmatrix} \text{sgn}(fm_x) & 0 & 0 \\ 0 & \text{sgn}(fm_y) & 0 \\ 0 & 0 & \text{sgn}(k_{33}) \end{bmatrix} \quad (2.34)$$

Die Matrizen für innere Kalibrierung und Rotation können nun unter Zuhilfenahme von  $\zeta(K)$  korrigiert werden:

$$K_{\text{kor}} = K \zeta(K) \quad (2.35)$$

$$R_{\text{kor}} = \zeta(K) R \quad (2.36)$$

## 2.4 Nichtlineare Optimierung

Während der DLT-Ansatz einer Minimierung des algebraischen Fehlers entspricht, zielen nichtlineare Optimierungen auf eine Minimierung des geometrischen Fehlers ab.

$$\min_P \sum_i d(\mathbf{x}_{i,i}, P \mathbf{X}_{w,i}) \quad (2.37)$$

Unter der Annahme von gaußverteilten Messfehlern entspricht dies der Maximum-Likelihood-Schätzung. Wird die zu minimierende Fehlerfunktion allerdings nur gemäß Gleichung 2.37

definiert, ergibt sich zwar aus photogrammetrischer Sicht ein Modell, das bei Konvergenz den geringsten geometrischen Fehler aufweist. Allerdings ist damit nicht sichergestellt, dass sich das Modell als solches auch in eine computergraphische Darstellung überführen lässt. Insbesondere ist der Parameter  $\gamma$ , der den Winkel zwischen den Bildachsen repräsentiert, problematisch. Aus diesem Grund empfehlen Hartley und Zisserman (2004) die folgende Erweiterung der Fehlerfunktion, wobei für den zweiten Ergänzungsterm davon ausgegangen wird, dass das Seitenverhältnis des Bildes bereits durch die Anzahl der Bildpunkte ausgeglichen ist, die Pixel selbst also als quadratisch angenommen werden können:

$$\min_P \sum_i d(\mathbf{x}_{i,i}, P \mathbf{X}_{w,i}) + \omega \gamma^2 + \omega(\alpha_x - \alpha_y)^2 \quad (2.38)$$

Durch diese Erweiterung stellt sich allerdings das Problem, dass das Gesamtmodell wieder ungenauer wird bezüglich des reinen geometrischen Fehlers. Insbesondere bei Kalibrierpunktanordnungen, die degenerativen Anordnungen<sup>8</sup> nahe kommen, steigt der Rückprojektionsfehler sehr stark an. Da im Anwendungsfall gerade der Fall, dass alle Kalibrierpunkte in oder zumindest annähernd in einer Ebene liegen, durchaus vorkommen kann, wurde nach einer Möglichkeit gesucht, den in diesem Schritt entstehenden Fehler zu minimieren. Ergebnis ist ein mehrstufiger nichtlinearer Optimierungsprozess, der zunächst das Lochkammermodell schrittweise mit ansteigenden Werten für  $\omega$  möglichst an ein überführbares Modell annähert und dabei Gleichung 2.38 minimiert. In einem zweiten Schritt wird dann nach Umwandlung in ein computergraphisches Modell eben dieses nochmals optimiert, allerdings unter reiner Berücksichtigung des geometrischen Fehlers. Dieser Ansatz berücksichtigt allerdings noch keine nichtlinearen Verzerrungen.

### 2.4.1 Optimierungsverfahren

Auf Grund der Problemstruktur und der Berechnungsgeschwindigkeit hat sich im Bereich der Kamerakalibrierung der Levenberg-Marquardt-Algorithmus<sup>9</sup> durchgesetzt. Er zählt zu den Methoden der lokalen nichtlinearen Optimierung ohne Nebenbedingungen und ist eine Kombination aus dem Gauß-Newton- und einem Gradientenverfahren, da der Gauß-Newton-Algorithmus selbst nur geringe Robustheit aufweist. Das Minimierungsproblem wird dabei in jedem Iterationsschritt linearisiert, wozu die Hesse-Matrix berechnet werden muss, zumindest jedoch die Jacobi-Matrix für die vereinfachte Berechnung gemäß Gleichung 2.39. Durch Anpassung eines Fehlergrenzwertes kann die Minimierung bei jedem Schritt erzwungen werden. Schließlich geht der Levenberg-Marquardt-Algorithmus lokal in den Gauß-Newton-Algorithmus über. Die Konvergenz ist allerdings quadratisch.

$$H = \nabla^2 f(x) = J(x)^T J(x) \quad (2.39)$$

$$x_{i+1} = x_i - (H + \lambda I)^{-1} \nabla f(x_i) \quad (2.40)$$

Gleichung 2.40 zeigt den Aktualisierungsschritt des reinen Levenberg-Algorithmus (Ranganathan 2004). Der Steuerfaktor  $\lambda$  regelt dabei die Gewichtung des Gradientenverfahrens.

<sup>8</sup> Der Fall degenerativer Punktanordnungen wird in Kapitel 2.6.2 behandelt.

<sup>9</sup> Marquardt 1963.

Sinkt der Fehler nach einem Berechnungsschritt, so wird  $\lambda$  typischerweise um den Faktor 10 verkleinert, da die quadratische Annäherung an  $f(x)$  offenbar richtig ist und somit der Einfluss des Gradientenverfahrens verringert wird. Andernfalls soll dem Gradienten stärker gefolgt werden, was zu einer Erhöhung von  $\lambda$  um den Faktor 10 führt. Das Problem dieser Vorgehensweise liegt darin, dass bei großen Werten für  $\lambda$  die berechnete Hesse-Matrix kaum mehr berücksichtigt wird, obwohl sie durchaus für die Gewichtung der einzelnen Parameter verwendet werden kann. Dies führt zum Levenberg-Marquardt-Algorithmus, dessen Aktualisierungsschritt Gleichung 2.41 zeigt.

$$x_{i+1} = x_i - (H + \lambda \operatorname{diag}(H))^{-1} \nabla f(x_i) \quad (2.41)$$

Der gesamte Algorithmus ergibt sich somit zu:

1. Durchführen eines Aktualisierungsschrittes gemäß Gleichung 2.41.
2. Berechnung des Fehlers mit dem neuen Parametervektor.
3. Hat sich der Fehler erhöht, wird der Schritt verworfen (die Werte des Parametervektors also zurückgesetzt) und  $\lambda$  erhöht. Anschließend wird mit Schritt 1 fortgefahren.
4. Hat sich der Fehler verringert, werden die aktualisierten Parameter übernommen und  $\lambda$  verringert.

### 2.4.2 Optimierung des Lochkameramodells

Mittrapiyanuruk (2006) beschreibt eine grundlegende Optimierung des Kameramodells in Matlab für den koplanaren Fall. Für die Optimierung wird hier allerdings die erweiterte Fehlerminimierung aus Gleichung 2.38 herangezogen und das Modell entsprechend auf eine allgemeine dreidimensionale Abbildung erweitert. Die Rotationsmatrix wird gemäß der Rodrigues-Formel (Gleichung 2.10) parametrisiert. Die Matlab-Implementierung zur Berechnung der Jacobi-Matrix ist in Kapitel A.1 angefügt.

### 2.4.3 Optimierung des computergraphischen Modells

Die Optimierung des computergraphischen Modells erfolgt auf ähnliche Weise, allerdings wird als Modell das DirectX eigene Kameramodell, wie in Abschnitt 2.1.3 beschrieben, verwendet. Es handelt sich um ein rechtshändiges System, das allerdings in DirectX explizit erstellt werden kann. Die Berechnung der partiellen Ableitungen kann auch hier dem Kapitel A.1 entnommen werden. Zur Umrechnung der Matrizen aus den Ausgangsparametern wurden eigene Matlab-Funktionen gemäß der DirectX Dokumentation implementiert, die sich im Anhang finden.

## 2.5 Numerische Probleme

Setzt man Punktkorrespondenzen entsprechend der direkten linearen Transformation in Gleichung 2.27 ein, stellt man fest, dass die Matrixeinträge für das Ausgleichsverfahren

stark in ihrem Wertebereich variieren. Je nach Größenordnung des aufgespannten Volumens in Weltkoordinaten ergeben sich Unterschiede über mehrere Zehnerpotenzen hinweg. Dieser Umstand wirkt sich numerisch negativ auf die Singulärwertzerlegung aus, die zur Lösung des Gleichungssystems verwendet wird.

### 2.5.1 Normalisierung der Kalibrierpunkte

Da die DLT darüber hinaus nicht invariant gegenüber Ähnlichkeitstransformationen ist, ist eine Normalisierung<sup>10</sup> der Bild- und Weltkoordinaten unbedingte Voraussetzung für die Anwendung des Algorithmus. Die Messpunkte werden so ähnlichkeitstransformiert, dass ihr Schwerpunkt im Ursprung liegt und die mittlere geometrische Distanz zu diesem im zweidimensionalen Fall  $\sqrt{2}$ , im dreidimensionalen  $\sqrt{3}$  beträgt. Entsprechende Transformationen zu normalisierten Kalibrierpunkten  $\hat{\mathbf{X}}_c$  und  $\hat{\mathbf{X}}_w$  sind wie folgt:

$$\hat{\mathbf{X}}_c = \begin{bmatrix} \frac{\sqrt{2}}{d_B} & 0 & \frac{-\sqrt{2} \bar{d}_{B,X}}{d_B} \\ 0 & \frac{\sqrt{2}}{d_B} & \frac{-\sqrt{2} \bar{d}_{B,Y}}{d_B} \\ 0 & 0 & 1 \end{bmatrix} \mathbf{X}_c \quad \hat{\mathbf{X}}_w = \begin{bmatrix} \frac{\sqrt{3}}{d_W} & 0 & 0 & \frac{-\sqrt{3} \bar{d}_{W,X}}{d_W} \\ 0 & \frac{\sqrt{3}}{d_W} & 0 & \frac{-\sqrt{3} \bar{d}_{W,Y}}{d_W} \\ 0 & 0 & \frac{\sqrt{3}}{d_W} & \frac{-\sqrt{3} \bar{d}_{W,Z}}{d_W} \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{X}_w \quad (2.42)$$

## 2.6 Ergebnisse

Für die folgenden Auswertungen dienen die Kalibrierdaten aus dem ersten Testeinsatz des Gesamtsystems während des alpinen Skiworldcups in Kitzbühel. Insgesamt waren 11 Kameras des Typs AXIS<sup>11</sup> Q1755, 9 davon zur Erfassung der Abfahrtsstrecke Streif und zwei weitere zur Bilderfassung des Ganslernhanges für den Slalombewerb im Einsatz. Die Aufnahme der Weltpunkte erfolgt mittels differenzieller GPS-Messung (siehe Kapitel 3). Die erreichte Genauigkeit der Punkte in Weltkoordinaten lag dabei unter einem halben Meter, vielfach im Bereich weniger Zentimeter. Die Abbildungen 2.5 - 2.7 zeigen drei Konstellationen von Kalibrierpunkten aus dem Testdatensatz. Um den Fehler, verursacht durch die Ebenheit der Kalibrierpunktconstellation (siehe Abschnitt 2.6.2), zu minimieren, wurde für jeden Punkt sowohl der Fußpunkt als auch die Spitze der Messlanze vermessen, was zu einer Maximierung des aufgespannten Volumens selbst bei ebenen Geländeflächen führt, wenngleich ein zusätzlicher Fehler entsteht, wenn die Lanze zur Messung nicht im Lot gehalten wird, da die zwei Messpunkte nicht mehr unabhängig voneinander sind. Zur Kalibrierung wurde der vorgestellte Algorithmus eingesetzt, der dem Gold-Standard-Algorithmus<sup>12</sup> mit Erweiterung der Optimierung für DirectX entspricht.

### 2.6.1 Messungen

Als repräsentative Messungen wurden drei der 11 Kameras ausgewählt, die möglichst unterschiedliche Lage in Bezug auf die Geländefläche besitzen. Tabelle 2.2 zeigt die Ka-

<sup>10</sup> vgl. Hartley und Zisserman 2004.

<sup>11</sup> AXIS ist ein eingetragenes Markenzeichen der Axis Communications AB

<sup>12</sup> Hartley und Zisserman 2004.

librierergebnisse im Detail. In Klammer ist neben dem Verfahren die Anzahl der Iterationsschritte der nichtlinearen Optimierung angeführt. Neben dem Residuum des algebraischen Ausgleichsverfahrens sind der endgültige Rückprojektionsfehler im Bildraum sowie der Rückprojektionsfehler in Weltkoordinaten angegeben. Letzterer weist dabei eine hohe Ungenauigkeit auf, da die Kalibrierpunkte oft außerhalb der vermessenen Geländeﬂäche liegen und somit nur durch Extrapolation der Vermessungsdaten ein Schnittpunkt gewonnen werden kann. Oft ist nicht einmal das möglich, was die Anzahl der Kontrollpunkte erheblich einschränkt.

Auf den Rückprojektionsfehler im Bildraum wirken sich lediglich die Messunsicherheit des Kalibrierpunktes selbst sowie etwaige Fehler bei der Zuordnung der Bildkorrespondenz aus. Da diese von Hand durchgeführt wurde und die Kontrastverhältnisse zum Teil sehr schlecht waren, lassen sich in den Aufzeichnungen immer wieder Fehlzweisungen finden, die allerdings durch die übrigen Messpunkte ausgeglichen werden. Sie gehen dennoch in die Fehlercharakteristik mit ein und verschlechtern den Rückprojektionsfehler entsprechend. Der Prozess der nichtlinearen Optimierung mit Nebenbedingungen zur Modellanpassung wurde mehrfach iterativ mit ansteigendem Wert für  $\omega$  (siehe Gleichung 2.38) durchgeführt. Dadurch steigt der Fehler im Bildraum, allerdings sinkt der Fehler in Weltkoordinaten in gleichem Maße. Die Optimierung des DirectX Kameramodells hat dann nur mehr wenig Einfluss, was aber zu erwarten war, da ja das Kameramodell zum Zeitpunkt der Überführung bereits durch die Nebenbedingungen stark an das virtuelle Kameramodell angenähert ist. Der Fehler sinkt nur mehr gering in der Größenordnung von  $10^{-6}$  Pixel. Das starke Ansteigen der Abweichung in Weltkoordinaten liegt hauptsächlich daran, dass sich die Kontrollpunkte am Rand der Geländeﬂäche befinden und somit gerade bei Geländekanten eine korrekte Rückprojektion bei idealer Punktlage nicht mehr möglich ist. Die Abbildungen 2.5 - 2.7 zeigen die einzelnen Punktkonfigurationen. Die jeweiligen Fußpunkte dienen als Kontrollpunkte im Objektraum, sofern sie auf der Geländeﬂäche liegen. Obwohl für jede Perspektive lediglich vier Kalibrierpunkte aufgenommen wurden, ist der resultierende Gesamtfehler verhältnismäßig klein.

Tabelle 2.3 zeigt die kalibrierte innere Orientierung der Kameras jeweils mit und ohne Optimierung. Der Einfluss des Skew-Parameters auf die Abweichung zu den tatsächlichen Kameradaten lässt sich deutlich erkennen, allerdings auch die Minimierung des selben durch die mehrstufige nichtlineare Optimierung.

### 2.6.2 Auswirkungen der Lage der Kalibrierpunkte

Die Qualität der direkten linearen Transformation und somit allen nachfolgenden Algorithmen hängt wesentlich von der Anordnung und Messgenauigkeit der Kalibrierpunkte ab.

Kalibrierung (Iter.)	DLT [res] algebraischer Fehler	Bild [px]		Welt Rückprojektion <sup>a</sup> [m]	
		mittlerer Fehler	maximaler Fehler	mittlerer Fehler	maximaler Fehler
Mausefalle, DLT	0,0035	2,7360	3,9242	11,3757	22,6333
Mausefalle, LM Kamera (30)	0,0035	8,6590	12,3561	4,6646	13,6161
Mausefalle, LM DirectX <sup>b</sup> (0)	0,0035	2,7360	3,9242	11,3757	22,6333
Mausefalle, LM (41+4)	0,0035	8,6590	12,3561	4,6646	13,6161
Lärchenschuss, DLT	0,0062	4,8032	6,7414	3,6227	4,8758
Lärchenschuss, LM Kamera (33)	0,0062	6,8431	9,1506	5,2849	9,1430
Lärchenschuss, LM DirectX <sup>c</sup> (1)	0,0062	4,8032	6,7414	3,6227	4,8758
Lärchenschuss, LM (33+1)	0,0062	6,8431	9,1506	5,2849	9,1430
Oberhausberg, DLT	0,0115	7,3428	18,2238	2,2197	6,3048
Oberhausberg, LM Kamera (19)	0,0115	5,4162	8,4997	1,5349	4,8038
Oberhausberg, LM DirectX (7)	0,0115	6,8898	15,0610	2,7822	7,0090
Oberhausberg, LM (19+6)	0,0115	5,5203	8,2860	3,0865	6,5112

<sup>a</sup> Nicht alle Referenzpunkte lagen innerhalb des vermessenen Geländeprofiles. Aus diesem Grund gibt es zum Teil erhebliche Abweichungen.

<sup>b</sup> ohne Konvergenz

<sup>c</sup> Konvergenz außerhalb des dargestellten Wertebereiches

Tabelle 2.2: Gesamtfehler

Kalibrierung	Brennweite [Px]	Öffnungswinkel horizontal [°]	Öffnungswinkel vertikal [°]	Skew $\gamma$ [Px]
Kameradaten	1764,32	50°	28,125°	
Mausefalle, DLT	1314,28	49,9183°	30,6321°	12,0872
Mausefalle, LM	1338,94	50,7158°	30,0939°	6,3732
Lärchenschuss, DLT	1399,18	48,5514°	28,8234°	29,7173
Lärchenschuss, LM	1408,91	48,8477°	28,6329°	8,1323
Oberhausberg, DLT	1433,35	47,7210°	27,7443°	71,1320
Oberhausberg, LM	1439,25	47,8944°	27,6383°	3,0244

Tabelle 2.3: Innere Orientierung der Kamera



Abbildung 2.5: Kalibrierbeispiel „Mausefalle“

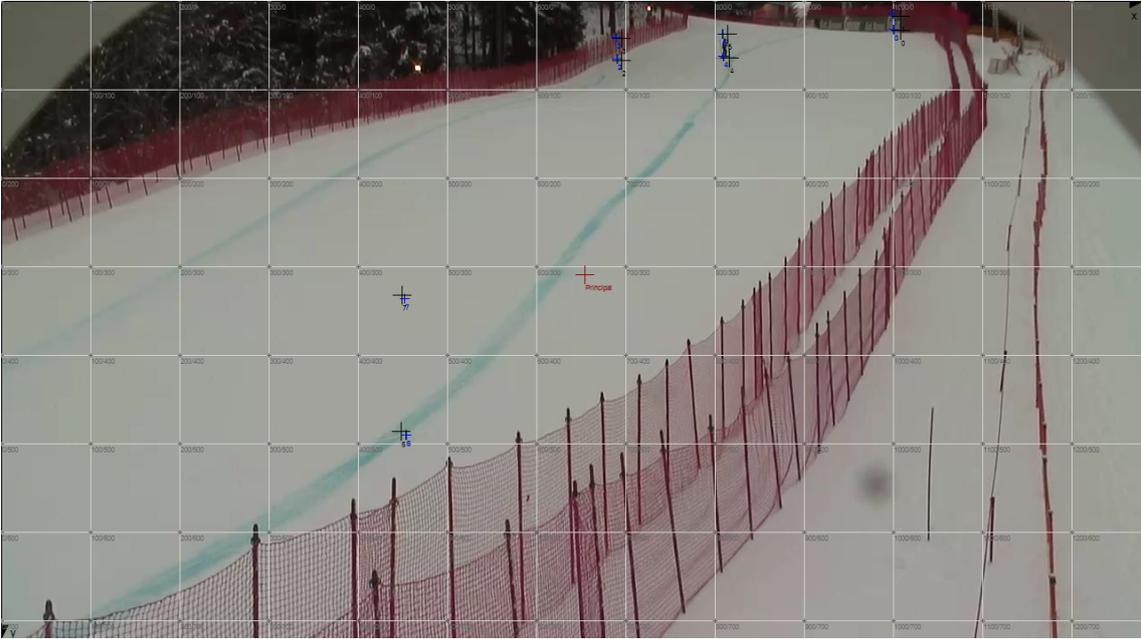


Abbildung 2.6: Kalibrierbeispiel „Lärchenschuss“

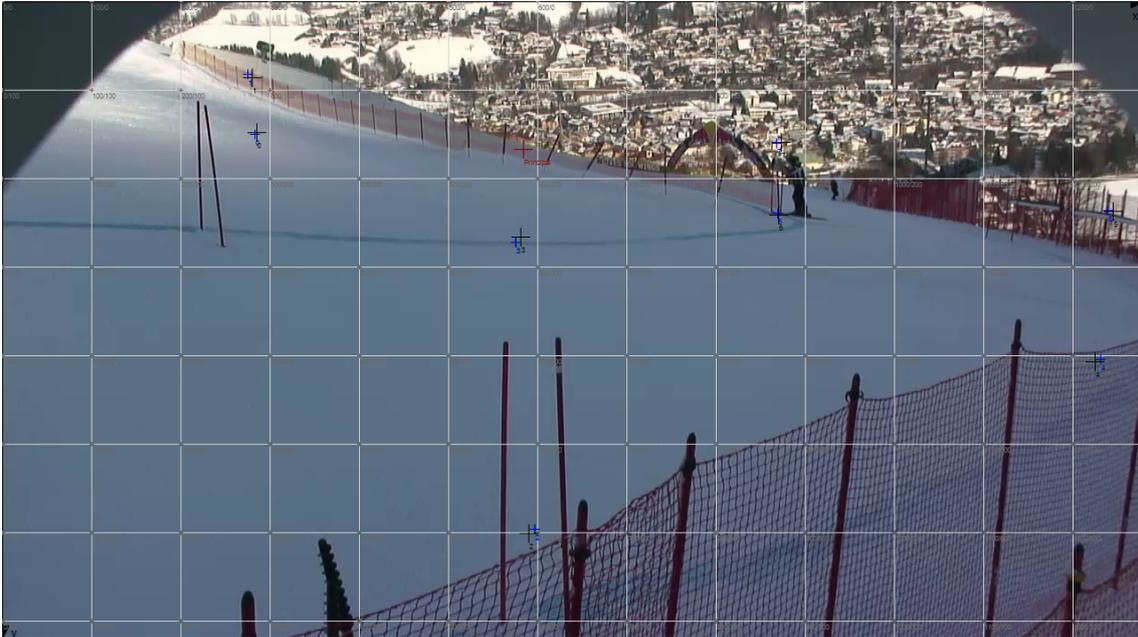


Abbildung 2.7: Kalibrierbeispiel „Oberhausberg“

### Degenerative Konfigurationen

Für die verwendete 6-Punkt DLT existieren zwei degenerative Konfigurationen:

1. Alle Kalibrierpunkte liegen in einer Ebene oder auf einer Geraden, die durch den Brennpunkt verläuft.
2. Sowohl Kamera als auch Kalibrierpunkte liegen auf einer kubischen Wendelinie<sup>13</sup>

Im vorliegenden Anwendungsfall ist insbesondere der erste Fall von Bedeutung, da Geländeflächen durchaus Ebenen darstellen können. Die Auswirkungen der Ausdehnung in der dritten Dimension sind in Abbildung 2.8 zusammengefasst. Liegt sie unter 15%, steigt der Kalibrierfehler rapide an, natürlich mit Ausnahme der koplanaren DLT, deren Fehlercharakteristik sich invers verhält.

Um diese Problematik auszugleichen, kann jeder Messpunkt nicht nur auf der Geländefläche, sondern auch mit einem Höhenoffset vermessen werden, sodass sich in Summe doppelt so viele Kalibrierpunkte ergeben. Dies ist insofern günstig, als die Antenne des DGPS Systems ohnehin auf einer Messlanze angebracht ist, um einen konstanten Abstand zum Boden zu halten. Mit einer Höhe von 2,20m kann auf diese Weise vor allem im Nahbereich der Kamera eine Aufweitung des Kalibriervolumens ohne große Schwierigkeit erreicht werden.

### Bildhauptpunkt

Eine weitere Unsicherheit in der Kalibrierung ergibt sich durch die Schätzung des Bildhauptpunktes. Es zeigt sich, dass die direkte lineare Transformation zu einer Maximierung

<sup>13</sup> In der Literatur häufig als „twisted cubic“ bezeichnet

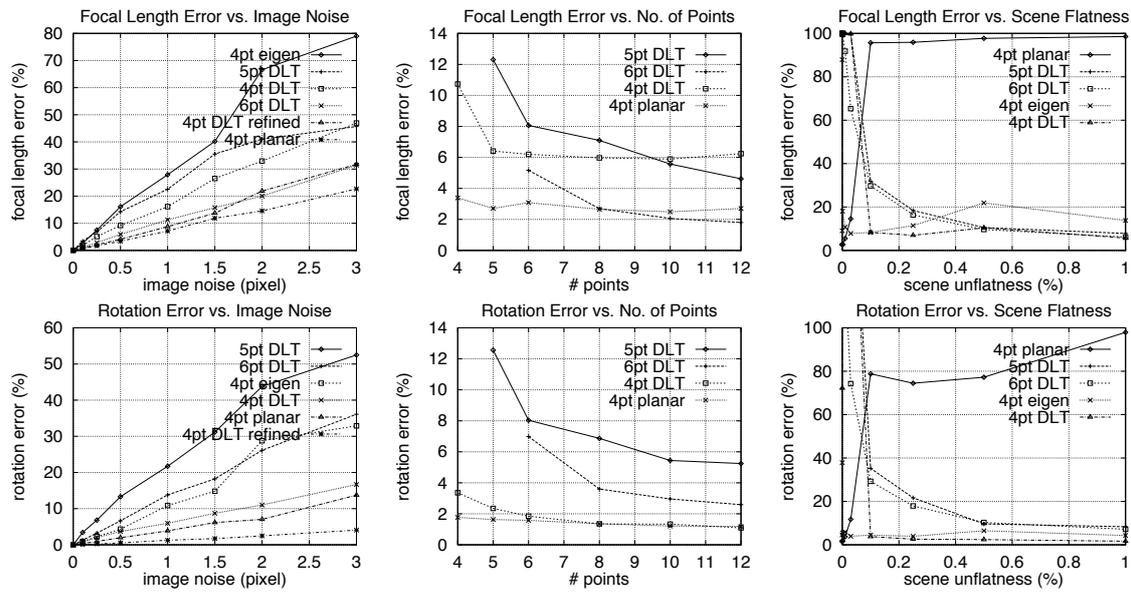


Abbildung 2.8: Vergleich unterschiedlicher Kalibriermodelle (Triggs 1999)

der Brennweite in der Projektion führt. Bei ungleichmäßig angeordneten Punkten ergibt sich daraus eine Verschiebung der optischen Achse in Richtung weiter entfernter Punkte (deutlich in Abbildung 2.5 ersichtlich). Ruiz, Teruel und Garcia-Mateos (2002) zeigen, dass allgemein eine verlässliche Bestimmung des Bildhauptpunktes bei kleinem bis mittlerem Öffnungswinkel der Kamera und verrauschten Messdaten nicht möglich ist. Es wird allerdings auch darauf hingewiesen, dass die genaue Position irrelevant für die Kalibrierung ist und daher vielfach vereinfachte Kameramodelle bessere Ergebnisse liefern, wenn eine sub-pixelgenaue Auflösung ohnehin nicht notwendig ist. Nachdem DirectX die Verwendung getrennter Öffnungswinkel ausgehend vom Bildhauptpunkt unterstützt (vgl. Gleichung 2.18), kann daher auch eine nahezu beliebige Position der optischen Achse modelliert werden. Dies ist auch dann möglich, wenn der Bildhauptpunkt außerhalb des Bildes liegt.

### 2.6.3 Linsenverzerrungen

Berücksichtigt man in den Beispielen Messunsicherheiten auf Pixelebene, lässt sich eine leichte radiale Verzerrung ausgehend vom Bildmittelpunkt feststellen, die allerdings selbst an den Randbereichen des Bildes eine Abweichung von zwei Pixel bei einem horizontalen Öffnungswinkel von  $50^\circ$  nicht übersteigt. Für stärkeren Weitwinkelbereich sollte eine Modellierung dieser Verzerrung zur Stabilisierung des Algorithmus in Weiterentwicklungen mit einbezogen werden.

## Kapitel 3

# Tiefenrekonstruktion

In diesem Kapitel soll die Vorgehensweise zur Gewinnung des dreidimensionalen Weltpunktes im Objektraum aus einem zweidimensionalen Bildpunkt erarbeitet werden. Im Regelfall geht durch die Projektion der Kamera die Tiefeninformation verloren. Es gibt mehrere Methoden, diese Information wiederzugewinnen<sup>1</sup>. Dazu zählen die zusätzliche Messung von Tiefeninformationen über Sensoren, aktives Sehen, verbunden mit der gezielten (kodierten) Ausleuchtung der Szene<sup>2</sup>, binokulare Stereopsis oder auch die Analyse mehrerer aufeinanderfolgender monokularer Bilder. Die Ausnutzung zusätzlicher Informationen aus Einzelbildern führt unter der Voraussetzung starker Annahmen zu einschränkenden Methoden, die Textur oder Schattierung des betrachteten Objektes mit einbeziehen. Dieser Arbeit liegt die Idee zu Grunde, die Tiefeninformation aufgrund eines Modells der Fläche wieder zu gewinnen, auf der sich die Objekte von Interesse bewegen. Gelingt es, den Fußpunkt eines Objektes zu lokalisieren, kann die Position in Weltkoordinaten über den Schnittpunkt zwischen dem vom Bildpunkt ausgehenden Strahl im Sichtvolumen mit der Oberfläche des Modells gewonnen werden.

### 3.1 Grundlegende Überlegungen

Allgemein lässt sich das folgende Verfahren überall dort anwenden, wo sichergestellt ist, dass sich die Weltpunkte auf einer Fläche befinden, für die ein Modell verfügbar oder erstellbar<sup>3</sup> ist. Existiert ein virtuelles Modell der betrachteten Fläche, lässt sich eine virtuelle Kamera so positionieren, dass sie ein Abbild des Modells erzeugt, das geometrisch mit dem Bild der realen Kamera zusammenfällt (siehe Abbildung 3.1). Diese Minimierungsaufgabe ist bereits durch die Kamerakalibrierung, wie im letzten Kapitel beschrieben, gelöst. In einem weiteren Schritt muss nun zunächst das Modell der betrachteten Geländefläche gewonnen und sodann, ausgehend von innerer und äußerer Kalibrierung der Kamera, der Schnittpunkt mit diesem für jeden möglichen Bildpunkt berechnet werden. Dieser Vorgang kann im Vorhinein offline durchgeführt werden, sodass für die Echtzeitverarbeitung bereits eine Zuordnungstabelle zwischen Bild- und Weltpunkten vorliegt.

---

<sup>1</sup> Chen, Brown und Song 2000.

<sup>2</sup> Pages u. a. 2003.

<sup>3</sup> Auf die kombinierte Kalibrierung und Modellgenerierung wird in dieser Arbeit später eingegangen werden

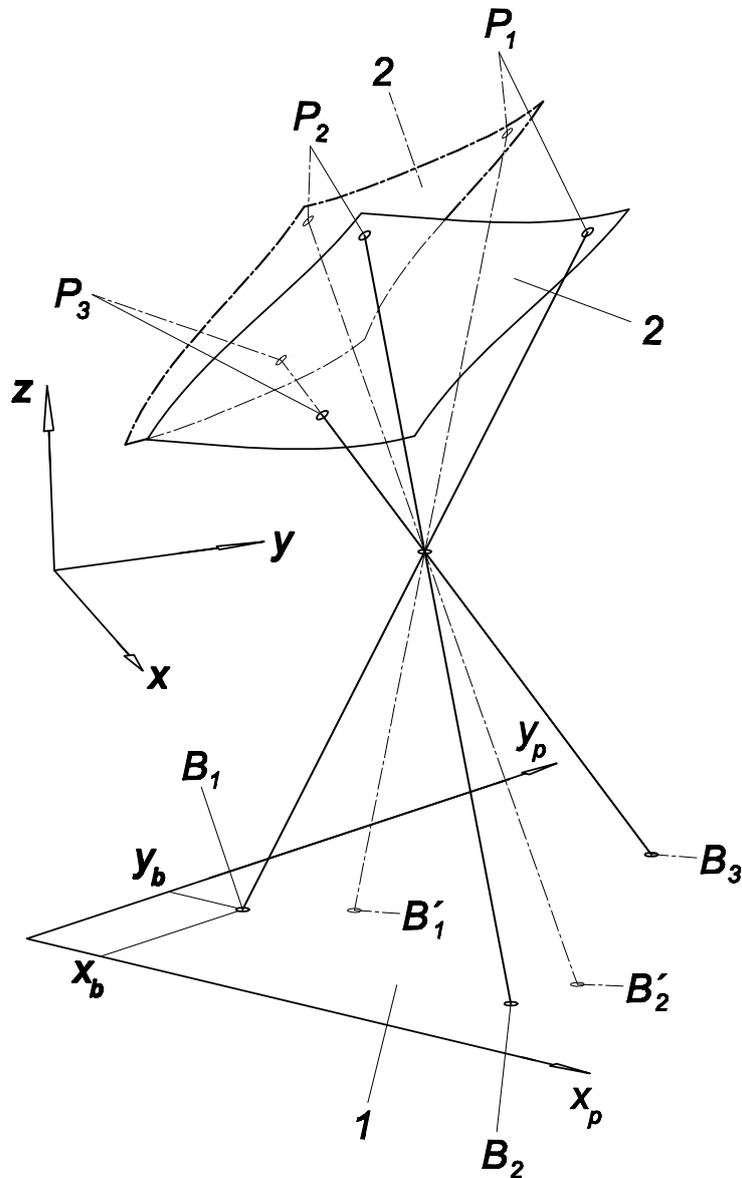


Abbildung 3.1: Übereinstimmung von virtuellem Modell und dem tatsächlichen Objekt: Die relative Orientierung zwischen Kamera und Modell wird so angepasst, dass die perspektivische Projektion (1) des Modells (2) mit der Projektion des realen Objektes zusammenfällt. Dadurch wird hinsichtlich der Bildkoordinaten  $(x_b, y_b)$  die Distanz zwischen Projektionen der Modellpunkte ( $B_1 - B_3$ ) und Projektionen der entsprechenden Weltpunkte ( $B'_1 - B'_3$ ) minimiert. Dies entspricht dem in Kapitel 2.4 beschriebenen Rückprojektionsfehler.

## 3.2 Modellierung

Das Modell der betrachteten Ebene ist ein zentrales Element für die Bestimmung der Weltkoordinaten eines verfolgten Objektes. Es muss daher sehr präzise mit der Realität übereinstimmen und auch witterungsbedingte Einflüsse wie zum Beispiel Schneeauflagen berücksichtigen. Da entsprechende Geländedaten oft nur in geringer Auflösung vorhanden sind und die Anpassung derselben sehr aufwendig und fehleranfällig wäre, ist es zweckmäßig, zeitnah zur geplanten Objektpositionsmessung ein aktuelles Geländemodell zu erstellen. Im Wesentlichen sind zwei Verfahren in Betracht zu ziehen: Die Modellbildung aus Luftbildern, wie zum Beispiel der Videoaufzeichnung eines Überfluges, oder die direkte geodätische Vermessung von Geländepunkten. In beiden Fällen muss aus den so gewonnenen Modelldaten ein durchgehendes Oberflächennetz gebildet werden.

## 3.3 Modellbildung aus Luftbildaufnahmen

Die bildgestützte Modellbildung basiert auf der Berechnung von Homographien (vgl. Han und Kanade 1998). Dabei wird aus zumindest zwei Ansichten der gleichen Szene eine Rekonstruktion der Tiefeninformation durchgeführt. Je mehr Bilder zur Verfügung stehen, desto besser kann mit Hilfe von statistischen Methoden eine robuste Homographie gewonnen werden. Das entstehende Modell ist bis auf den Skalierungsfaktor und einen Referenzpunkt definiert, das heißt es muss nachträglich noch an die reale Welt hinsichtlich der Dimensionierung angepasst werden. Im Gegensatz zu der Modellaufnahme mittels Laserscannern ist zwar die abdeckbare Fläche wesentlich größer, doch muss auch mit hoher Redundanz, Ausreißern und großem Rauschanteil umgegangen werden (vgl. Salman und Yvinec 2009).

Oft ist durch Verdeckungen keine vollständige Rekonstruktion möglich, was insbesondere in Kantenbereichen und in der Umgebung von Baumbewuchs zu Problemen führt. Beschneite Geländeflächen machen darüber hinaus das Auffinden von Features für die Homographieberechnung schwierig, was häufig eine genaue Nachbildung unmöglich macht und an den Randbereichen verstärkt zu Fehlern führt (siehe Abbildung 3.2). Diese Aspekte machen zwingend eine manuelle Nachbearbeitung des Modells notwendig. In Kombination mit dem benötigten Überflug zur Gewinnung des Bildmaterials ist die Methode somit sowohl zu aufwändig als auch zu unpräzise für den gegebenen Anwendungsfall.

## 3.4 Modellbildung aus geodätisch erfassten Geländepunkten

Durch Stereopsis können zwar rasch viele Weltpunkte gewonnen werden, jedoch unter Aufgabe des tatsächlichen Geländeprofiles. Wählt man statt dessen markante Geländepunkte explizit aus, lässt sich mit einer wesentlich geringeren Punkteanzahl ein übereinstimmendes Geländemodell aufbauen. Unter der Annahme, dass die Geländefläche aus Verbindungslinien normal zur Falllinie, die durch je zwei gegenüberliegende Punkte definiert sind, vollständig erzeugt werden kann, lässt sich das Modell durch Messung eben dieser Randpunkte in Form einer Wendelfläche bilden. Diese Annahme ist für den Anwendungsfall zulässig, da bei Skipisten die Krümmung normal zur Falllinie im Allgemeinen vernachlässigbar ist.

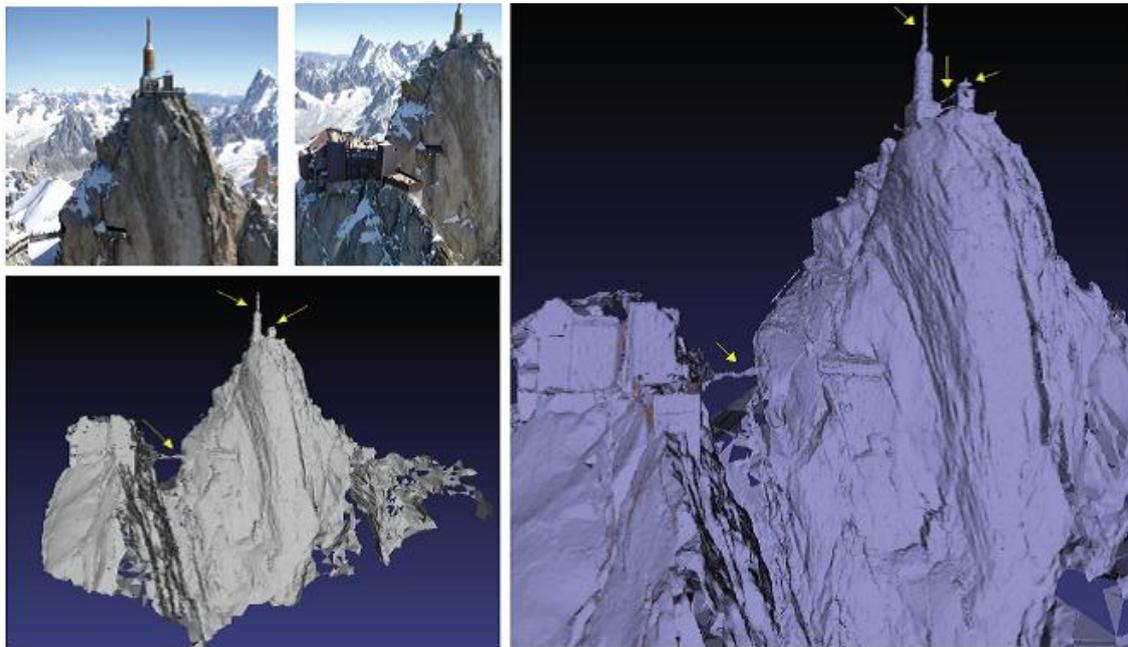


Abbildung 3.2: Modellrekonstruktion aus Luftbildaufnahmen (Salman und Yvinec 2009): Zwei Bilder aus dem Aiguille du midi Datensatz (©B.Vallet/IMAGINE).

Darunter eine Rekonstruktion geringer Auflösung (270000 Dreiecke), rechts die hochauflösende Rekonstruktion (1000000 Dreiecke).

In beiden Rekonstruktionen werden zwar Details erkannt und wiedergegeben, allerdings ist die Wiedergabe im Bereich von Schneefeldern mit Fehlern behaftet.

Für die Erfassung der Geländepunkte kommt ein globales Navigationssatellitensystem (GNSS) zum Einsatz, das ohne zusätzliche Stützdaten eine zu geringe Genauigkeit aufweist. Geht man von einer Bildauflösung von  $1080 \times 720$  px aus, so ergibt sich eine Bild-diagonale von 1298 px. Deckt man damit im Mittel eine Strecke von 500 – 600 m ab, ist die Auflösung pro Pixel, rückprojiziert in den Objektraum, ein halber Meter. Berücksichtigt man die perspektivische Projektion, ergibt sich für den Nahbereich eine höhere Auflösung im Dezimeter- beziehungsweise Sub-Dezimeterbereich.

### 3.4.1 Differenzielle GNSS Messungen

Satellitensysteme zur Positionsbestimmung bedienen sich der Laufzeitdifferenzen zwischen den Signalen unterschiedlicher Satelliten. Dies erfordert allerdings eine sehr genaue Zeitsynchronisation aufgrund der hohen Ausbreitungsgeschwindigkeit. Während die Zeitbasen der Satelliten von den Betreibern hochgenau synchron gehalten werden, wird auf Empfängerseite aus Kostengründen im Regelfall darauf verzichtet. Zu den drei Unbekannten der Empfängerposition kommt aus diesem Grund noch ein zusätzlicher Synchronisationsfehler, was die Notwendigkeit von vier Laufzeitmessungen für die Positionsbestimmung erklärt. Atmosphäre, Zeitdrift und Mehrwegausbreitung beeinflussen die Laufzeit zusätzlich, wobei der größte Fehler durch die Auswirkungen der Ionosphäre verursacht wird. Durch differenzielle Verfahren lassen sich aber die dadurch entstehenden Fehler genauso wie die vom Betreiber aus militärischen Gründen durchgeführte Verfälschung weitgehend ausgleichen. (Wübbena und Bagge 1995)

## GPS

Neben dem russischen GLONASS<sup>4</sup> ist NAVSTAR-GPS das einzig kommerziell nutzbare Satellitenortungssystem. Je nach Anzahl und Position der sichtbaren Satelliten ergibt sich im frei nutzbaren Bereich nach Abschaltung der als Selective Availability bezeichneten künstlichen Verschlechterung des Signals eine maximal erreichbare Genauigkeit von 5 m in horizontaler Richtung und 10 m in vertikaler Richtung.

## DGPS

Bei differenziellen Methoden geht man davon aus, dass sich die Auswirkungen atmosphärischer Störungen in einem begrenzten Gebiet nicht wesentlich unterscheiden. Kennt man somit den Messfehler für einen Referenzort, lässt sich dieser auch für Messungen in der Umgebung korrigieren, da damit die tatsächlichen Ausbreitungsgeschwindigkeiten der Satellitensignale bestimmt werden können. Die Übertragung des Korrektursignals kann entweder direkt über Funk erfolgen oder aber über Mobilfunknetze. Letztere Methode wird insbesondere bei Netzwerken von Referenzstationen gewählt, wie zum Beispiel auf nationaler Ebene APOS<sup>5</sup> und auf internationaler Ebene EGNOS<sup>6</sup> in Europa oder WAAS<sup>7</sup> in den USA. Standard netzwerkbasierter Übertragung ist das Networked Transport of RTCM via Internet Protocol (Ntrip).

<sup>4</sup> Globalnaja Nawigazionnaja Sputnikowaja Sistema

<sup>5</sup> Austrian Positioning Service des Österreichischen Bundesamtes für Eich- und Vermessungswesen

<sup>6</sup> European Geostationary Navigation Overlay Service

<sup>7</sup> Wide Area Augmentation System

Der APOS-Dienst betreibt ein Netzwerk von Referenzstationen, dessen Messdaten zentral verarbeitet und dazu verwendet werden, ortsbezogene Korrekturdaten in Echtzeit mittels virtuellen Referenzstationen zur Verfügung zu stellen. Die Auflösung von DGPS liegt bei einem Meter, da durch Mehrwegausbreitung entstehende Fehler nicht korrigiert werden können.

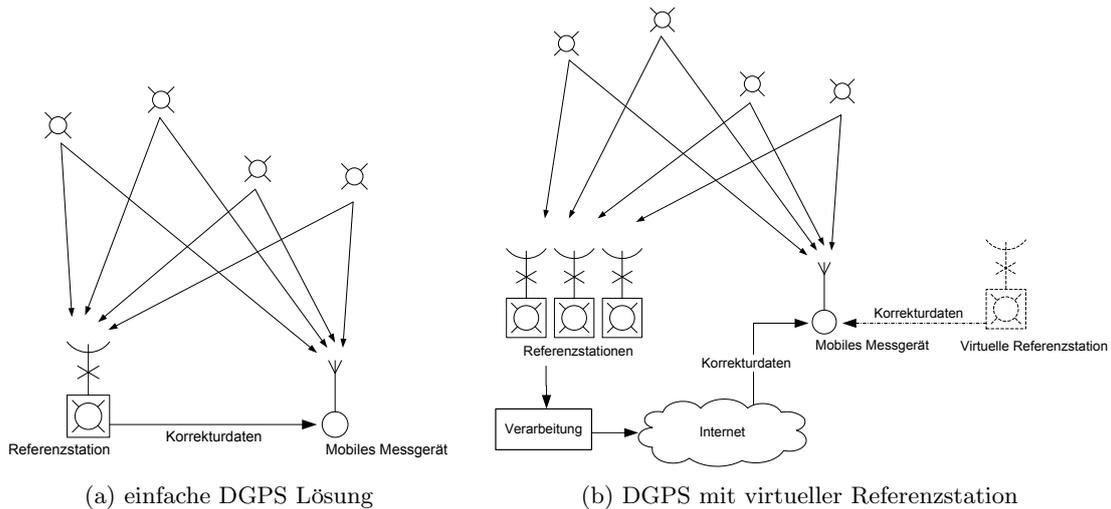


Abbildung 3.3: DGPS Systeme

## RTK

Real Time Kinematic GPS (RTK-GPS oder auch kurz RTK) verwendet neben dem C/A Code<sup>8</sup> auch die Trägerphase des Signals zur Positionsbestimmung. Aus diesem Grund ist die maximale Entfernung zur Basisstation auf 10 km limitiert. Die Initialisierung benötigt im Regelfall eine Minute, bedingt durch den Aufwand zur Eliminierung der Mehrdeutigkeiten im Phasensignal. Dann kann eine Genauigkeit von 2 cm erreicht werden (fixed - Messung mit zumindest fünf Satelliten).

### 3.4.2 Vergleich der Messmethoden

Für die Vermessung von Geländeflächen, im vorliegenden Anwendungsfall überwiegend in alpinem Gelände, sind neben der Genauigkeit der Messung auch andere Faktoren von Bedeutung. Dazu zählen die benötigte Zeit zur Initialisierung, die Robustheit gegenüber Störungen, beispielsweise Verdeckung, und das Verhalten bei bewegtem Empfänger. Vor allem die Abschattung durch Bäume in Kombination mit instabiler Datenverbindung beeinträchtigt die Vermessung stark. Ein Vergleich unterschiedlicher GNSS Systeme in Bezug auf diese Gegebenheiten wurde von Morales und Tsubouchi (2007b) durchgeführt. Im Test

<sup>8</sup> Coarse Acquisition Code oder auch Civilian Code ist der von den GPS-Satelliten generierte pseudo-zufällige Code, der zu Laufzeitmessung verwendet wird

waren insgesamt sieben unterschiedliche Empfängertypen unterschiedlicher Hersteller, jeweils mit einfacher differenzieller Korrektur und RTK. Überprüft wurde das Verhalten unter freiem Himmel sowie unter partieller Abschattung durch Bäume.

Die Ergebnisse (siehe Tabellen 3.1 und 3.2) zeigen, dass unter freiem Himmel, wie erwartet, RTK-Systeme eine höhere Genauigkeit mit sich bringen. Allerdings können im Bereich von Abschattung mit differenziellem GPS deutlich exaktere Messwerte erzielt werden, da starke Störungen den sofortigen Verlust der Phaseninformation bedeuten. Die Verfügbarkeit ist unter freiem Himmel bei beiden Methoden sehr hoch, unter einem Blätterdach zeigen RTK-Systeme jedoch deutliche Einbrüche. Dies liegt daran, dass sie zur Initialisierung insgesamt fünf Satelliten benötigen und dass die Zeiten für eine Reinitialisierung wesentlich höher sind. So benötigte das RTK System von Trimble beispielsweise im Schnitt 125 Sekunden, um die maximale Genauigkeit wieder zu erreichen.

Konfiguration	Verfügbarkeit [%]	
	freier Himmel	Abschattung
Trimble DGPS	99,93	99,18
NAVCOM DGPS	99,06	96,92
Trimble RTK	99,12	91,21
NAVCOM RTK	99,08	90,18
StarFire	99,04	81,05
StarFire-DGPS	99,04	94,66
StarFire-RTK	99,05	91,39

Tabelle 3.1: Verfügbarkeit unterschiedlicher GPS Systeme (Morales und Tsubouchi 2007b)

Konfiguration	freier Himmel]		Abschattung	
	Präzision <sup>a</sup>	HDOP <sup>b</sup>	Präzision	HDOP
Trimble DGPS	1,39	0,93	1,46	0,93
NAVCOM DGPS	1,95	1,50	1,83	2,31
Trimble RTK	1,27	1,12	9,44	4,66
NAVCOM RTK	0,07	1,19	3,49	2,05
StarFire	1,66	1,40	9,15	3,35
StarFire-DGPS	1,94	1,49	3,23	2,33
StarFire-RTK	0,19	1,28	6,21	2,66

<sup>a</sup> Das Maß der Präzision leitet sich aus den Standardabweichungen der x- und y-Koordinaten in der Ausgabe des Empfängers ab

<sup>b</sup> Horizontal Dilution Of Precision

Tabelle 3.2: Genauigkeit unterschiedlicher GPS Systeme (Morales und Tsubouchi 2007b)

Lässt sich daher sowohl für die Kalibrierung als auch für den Aufbau des Gittermodells eine so hohe Anzahl an Messpunkten gewinnen, sodass eine Ausgleichsrechnung durchgeführt werden kann, ist aus Stabilitätsgründen eine reine DGPS-Lösung vorzuziehen, da sie im Gelände wesentlich bessere Bedingungen für den Messprozess bietet. Dies ist im Bereich alpiner Sportveranstaltungen insofern entscheidend, als nur eine sehr begrenzte Zeitdauer für die Vermessung der Strecke eingeplant werden kann.

### 3.4.3 Überführung in kartesische Koordinaten mit Referenzpunkt

Da die Modellpunkte nach der Aufnahme in ellipsoidischen Koordinaten vorliegen, ist aus Gründen der leichteren Weiterverarbeitung und zur besseren Beurteilung der Messungen eine Umrechnung in ein lokales kartesisches Koordinatensystem notwendig. Es wird eine Koordinatentransformation in geozentrisch-kartesische Koordinaten durchgeführt, gefolgt von Translation und Rotation, um das resultierende Koordinatensystem zu einem lokalen Referenzpunkt zu verschieben, der den Ursprung kennzeichnet. Die Transformation<sup>9</sup> der ellipsoidischen Koordinaten, Länge ( $\eta$ ), Breite ( $\varphi$ ) und Höhe ( $h$ ), in kartesische lautet unter Berücksichtigung der Konstanten (Gleichung 3.2) für das WGS 84<sup>10</sup> Referenzellipsoid:

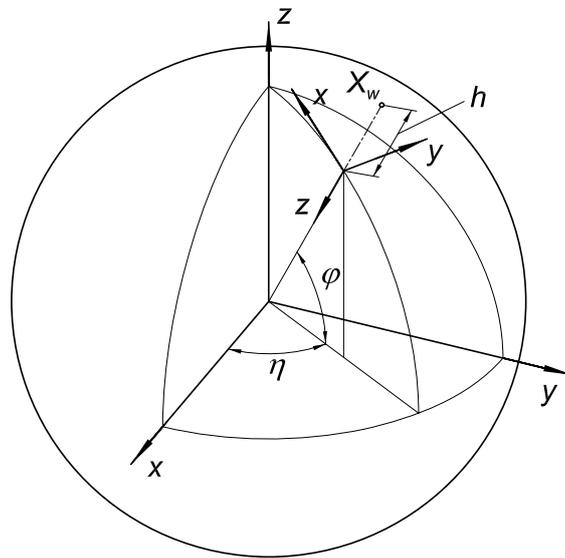


Abbildung 3.4: Koordinatentransformation zur Umrechnung eines ellipsoidischen Datums in kartesische Koordinaten des Referenzsystems.

$$\begin{aligned}
 x_{gc} &= (r_n + h) \cos \varphi \cos \eta \\
 y_{gc} &= (r_n + h) \cos \varphi \sin \eta \\
 z_{gc} &= (r_n (1 - e^2) + h) \sin \varphi
 \end{aligned} \tag{3.1}$$

<sup>9</sup> nach *Formeln und Konstanten für die Berechnung der Schweizerischen schiefachsigen Zylinderprojektion und der Transformation zwischen Koordinatensystemen.*

<sup>10</sup> World Geodetic System aus dem Jahr 1984

$$\begin{aligned}
&\text{gro\ss e Halbachse: } a = 6378137m \\
&\text{Abplattung: } f = \frac{1}{298,257223563} \\
&\text{kleine Halbachse: } b = a(1 - f) = 6356752,314m \\
&\text{erste numerische Exzentrizit\u00e4t: } e = \frac{\sqrt{a^2 - b^2}}{a} \\
&\text{Normalkr\u00fcmmungsradius: } r_n = \frac{a}{\sqrt{1 - e^2 \sin^2 \varphi}} \tag{3.2}
\end{aligned}$$

Die resultierenden Koordinaten haben den Erdmittelpunkt als Ursprung, der in einem weiteren Schritt auf den lokalen Referenzpunkt verschoben wird. Dies erfolgt durch die Translation um die Position des neuen Ursprungs in kartesischen, geozentrischen Koordinaten.

$$\begin{bmatrix} x_{rc} \\ y_{rc} \\ z_{rc} \end{bmatrix} = \begin{bmatrix} x_{gc} \\ y_{gc} \\ z_{gc} \end{bmatrix} - \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} \tag{3.3}$$

Letztendlich wird das Koordinatensystem so im Referenzpunkt gedreht, dass die x-Achse im Referenzpunkt in Richtung Norden und die y-Achse in Richtung Osten zeigt. Dazu muss eine passive Drehung (eine Drehung des Koordinatensystems) zuerst um den Winkel  $\eta$  um die z-Achse und dann um den Winkel  $\varphi$  um die y-Achse durchgef\u00fchrt werden. Somit ergibt sich f\u00fcr ein Rechtssystem die z-Achse in Richtung Erdmittelpunkt. Verglichen mit der H\u00f6henangabe des WGS 84 - Datums ist die z-Achse jedoch invertiert.

$$R_o = \begin{bmatrix} -\sin \eta_o \cos \varphi_o & -\sin \eta_o \sin \varphi_o & \cos \eta_o \\ -\sin \varphi_o & \cos \varphi_o & 0 \\ -\cos \eta_o \cos \varphi_o & -\cos \eta_o \sin \varphi_o & -\sin \eta_o \end{bmatrix} \tag{3.4}$$

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = R_o \begin{bmatrix} x_{rc} \\ y_{rc} \\ z_{rc} \end{bmatrix} \tag{3.5}$$

### R\u00fccktransformation

F\u00fcr die R\u00fccktransformation von kartesischen in ellipsoidische Koordinaten muss zun\u00e4chst das geozentrische Koordinatensystem wiederhergestellt werden.

$$\begin{bmatrix} x_{gc} \\ y_{gc} \\ z_{gc} \end{bmatrix} = R_o^{-1} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} \tag{3.6}$$

Da Breite ( $\varphi$ ) und Höhe ( $h$ ) voneinander abhängig sind, ist für die Transformation in das WGS 84 Referenzsystem keine geschlossene Lösung möglich. Sie muss iterativ aus den Gleichungen 3.7 gewonnen werden.

$$\begin{aligned}\eta &= \arctan \frac{y_{gc}}{x_{gc}} \\ \varphi &= \arctan \frac{\frac{z_{gc}}{\sqrt{x_{gc}^2 + y_{gc}^2}}}{1 - \frac{r_n e^2}{r_n + h}} \\ h &= \frac{\sqrt{x_{gc}^2 + y_{gc}^2}}{\cos \varphi} - r_n\end{aligned}\tag{3.7}$$

$$\varphi_0 = \frac{z_{gc}}{\sqrt{x_{gc}^2 + y_{gc}^2}}\tag{3.8}$$

Zu Beginn der iterative Berechnung kann der Startwert  $\varphi_0$  für die geographische Breite angenommen werden. Die Lösung konvergiert sehr rasch, da  $h \ll r_n$ .

#### 3.4.4 Aufbau des Modells

Die rechnerische Repräsentation des Modells soll die Schnittpunktberechnung erleichtern und gleichzeitig eine komfortable Darstellung ermöglichen. Wählt man Vertices als Entsprechung für die vermessenen Geländepunkte, lässt sich bei entsprechender Anordnung in Dreiecken ein Gittermodell (Mesh) generieren. Diese Darstellungsform ist für beide Anforderungen von Vorteil, wenngleich bei der Aufnahme der Stützpunkte auf eine gleichmäßige Verteilung geachtet werden muss. Führt man jedoch eine Interpolation zwischen den Messpunkten durch, kann auf diesen Anspruch verzichtet werden, da bei geringem Punktabstand die Anzahl der Vertices auf beiden Seiten der Geländefläche ausgeglichen werden kann.

Ein weiteres Problem stellt die Zugänglichkeit idealer Messpunkte dar. Kann aus gegebenen Umständen nur ein Punkt innerhalb der zu beobachtenden Fläche vermessen werden, lässt sich aus der Verbindungslinie mit dem gegenüberliegenden Punkt der nicht zugängliche Streckenabschnitt extrapolieren, um ein vollständiges Modell zu erhalten.

### 3.5 Wiedergewinnung des Weltpunktes

Durch Verlust der Tiefeninformation bei der Projektion eines Weltpunktes auf die Bildebene ergibt sich bei der Rückprojektion eines Bildpunktes ein Sichtstrahl im Objektraum. Im Modell ist dies eine von der Nearplane zur Farplane verlaufende Gerade, die bei Durchdringung der beiden die Schnittpunkte  $\mathbf{X}_n$  und  $\mathbf{X}_f$  erzeugt. Auf eben dieser Gerade liegt auch der tatsächliche Weltpunkt, der erster Schnittpunkt mit dem Modell, sofern er selbst Element der Modellfläche ist, was der ursprünglichen Annahme entspricht (siehe Abbildung 3.6).

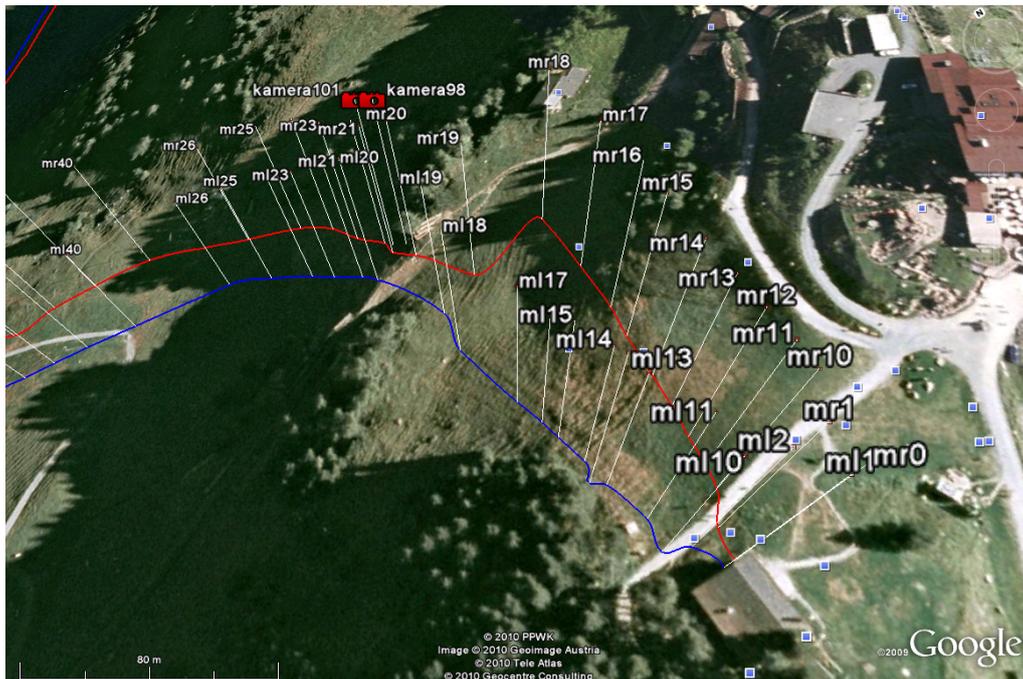


Abbildung 3.5: Vermessene Punkte und Interpolation der links- und rechtsseitigen Flächenbegrenzung mit den zugehörigen Kamerapositionen im Bereich Mausefalle.

### 3.5.1 Schnittpunktgewinnung

Die Schnittpunktberechnung zwischen einer Geraden und einem Drahtgittermodell lässt sich auf die Durchdringung einer Dreiecksfläche durch eine Gerade zurückführen. Dies ist in der Computergraphik ein bekanntes Problem; Rafael, Segura und Feito (2001) geben einen Überblick über bekannte Algorithmen. In einer ersten Phase wird ein etwaiger Schnittpunkt zwischen dem Sichtstrahl und der von dem Dreieck aufgespannten Ebene ermittelt, danach wird überprüft, ob dieser Schnittpunkt innerhalb des entsprechenden Dreiecks liegt. Dies lässt sich sehr effektiv in Baryzentrischen Koordinaten durchführen und wird heutzutage von jeder computergraphischen Entwicklungsumgebung unterstützt. Die Komplexität des Algorithmus hängt dabei direkt mit der Größe des Modells zusammen, allerdings sind entsprechende Verfahren auf Grund des breiten Anwendungsgebietes hoch optimiert (Shevtsov, Soupikov und Kapustin 2007), was eine sehr schnelle Berechnung aller Schnittpunkte für eine Perspektive ermöglicht.

Ein wesentlicher Parameter für die Qualität der Rückgewinnung ist der Schnittwinkel  $\gamma$ , da ein schleifender Schnitt im Objektraum erheblich größere Unsicherheiten mit sich bringt, insbesondere, als der Fußpunkt des Objektes auf der Geländefläche nicht immer zuverlässig erkannt werden kann.

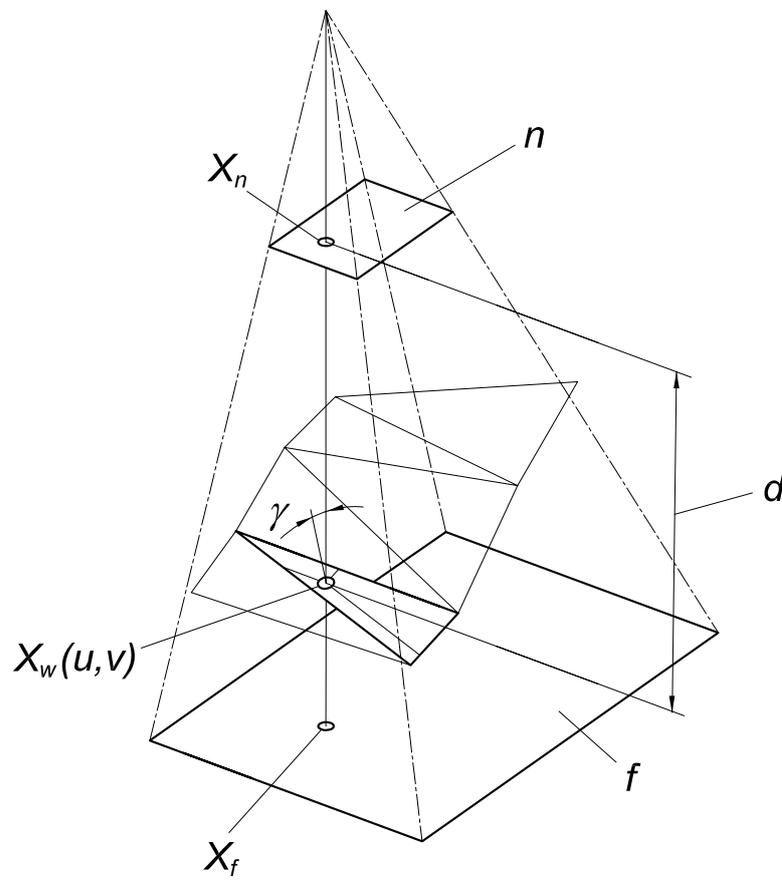


Abbildung 3.6: Schnittpunktgewinnung

### 3.6 Ergebnisse

Unter Verwendung der Kalibrierdaten der jeweiligen Kamera (siehe Kapitel 2) kann eine virtuelle Ansicht auf das Modell generiert werden. Der Prozessablauf hierzu ist in Abbildung 3.7 dargestellt. Durch Überlagerung mit dem tatsächlichen Kamerabild wird das Ergebnis des Prozesses visualisiert und der Gesamtfehler, entstehend durch Projektion des Weltpunktes  $\mathbf{X}_w$  in die Bildebene ( $\mathbf{X}_c$ ) sowie anschließende Rückprojektion und Schnittpunktgewinnung, berechnet. Für die 3,2 km lange Abfahrtsstrecke „Streif“ wurden je Seite 125 Geländepunkte vermessen, aus denen nach Interpolation 3140 Punkte im Abstand von einem Meter für die Modellbildung generiert wurden. Um eine gleichmäßige Punktanzahl zu erhalten, wurde die Gesamtstrecke der gegenüberliegenden Seite durch diese Punktanzahl geteilt, was einen mittleren Punktabstand von 0,99 m ergab. Aus den 6280 Vertices wurden sodann insgesamt 3140 Dreiecke zur Modellierung der Geländeﬂäche gebildet.

Die Auswertung aller 9 Kameras ergab einen mittleren Fehler von 6,89 m; dieser schließt allerdings außerhalb des Meshes liegende Referenzpunkte mit ein. Der von diesen Abweichungen bereinigte, mittlere Fehler liegt bei 3,75 m, bei vier Kameras aber deutlich unter diesem Durchschnitt. Je nach Punktkonfiguration lässt sich vom Bildhauptpunkt ausgehend eine radiale Verzerrung feststellen, die auf die nicht korrigierte Linsenverzeichnung zurückzuführen ist. Aufgrund der perspektivischen Projektion wächst der Fehler auch mit der Entfernung zur Kamera, was sich in Abweichungen des Öffnungswinkels und der Rotation derselben manifestiert. Vor allem die horizontale Achse ist davon betroffen, was in Anbetracht des Seitenverhältnisses allerdings zu erwarten ist.

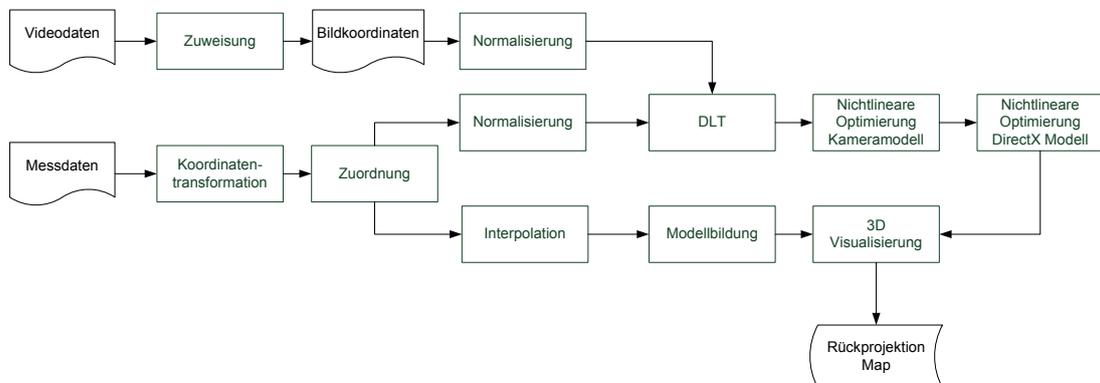


Abbildung 3.7: Prozessablauf zur Gewinnung der Korrespondenzpunkte für die Rückprojektion

Die Abbildungen 3.8 - 3.10 zeigen drei repräsentative Streckenabschnitte, zwei davon mit besonders flachem Blickwinkel, um das Modell gegenüber der Geländeﬂäche besser erkennen zu können. Durch einen Messfehler im oberen Bereich der Strecke ist eine deutliche Verschiebung in der Punktzuordnung sichtbar, die allerdings aufgrund der hohen Punktanzahl kein Problem darstellt. Abbildung 3.8 beinhaltet zusätzlich vier Kontrollpunkte, die jeweils mit ihrer tatsächlichen Position ( $t$ ) und der Rückprojektion ( $p$ ) eingezeichnet

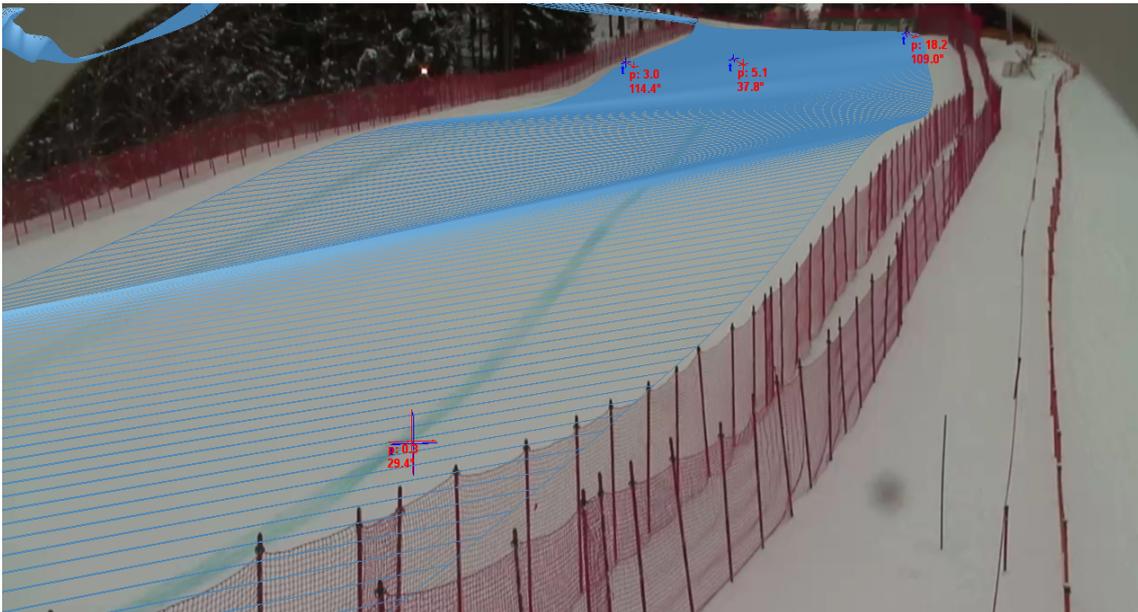


Abbildung 3.8: Streckenabschnitt „Lärchenschuss“ mit Kalibrierpunkten (blau), Rückprojektionsabweichung (rot) und Schnittwinkel  $\gamma$

sind. Zusätzlich ist auch der Schnittwinkel  $\gamma$  zwischen Flächennormalen und Sichtstrahl angegeben. Abbildung 3.10 zeigt die gegenseitige Verdeckung des Modells bei aktiviertem Z-Buffer. Da auch die Rückseite des Modells dargestellt wird, sind in der Abbildung nicht sichtbare Streckenabschnitte vorhanden.



Abbildung 3.9: Streckenabschnitt „Oberhausberg“

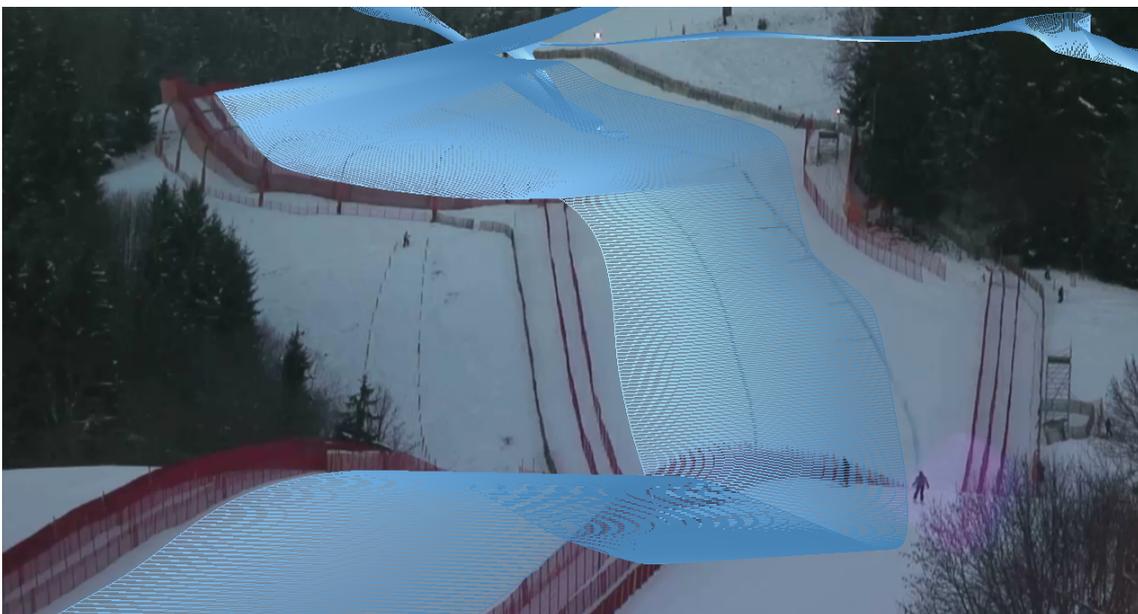


Abbildung 3.10: Streckenabschnitt „Zielschuss“

## Kapitel 4

# Objekterkennung

In der Kette der Verarbeitungsschritte des Kamerabildes, die in Echtzeit durchgeführt werden müssen, stellt die Objekterkennung den ersten Bereich dar. Es gilt, durch Separation von Vorder- und Hintergrund sich bewegende Objekte zu extrahieren und in weiterer Folge dieselben zu klassifizieren und zu filtern. Der Schritt der Klassifikation beinhaltet die Extraktion von Objekteigenschaften wie Schwerpunkt der Pixelwolke, Orientierung und Histogramm. Der erste Bereich wird als Bildverarbeitung bezeichnet, deren Ergebnis das in Position und Größe bekannte Objekt ist. Die Aufgabe der zweiten Gruppe von Verarbeitungsschritten, der Objektverarbeitung, ist die nähere Bestimmung und in weiterer Folge auch die Zuordnung von Objekten.

### 4.1 Ansätze zur Bilderkennung für Sportanwendungen

Es existieren vielerlei Ansätze, Objekte in sportspezifischen Anwendungen zu erkennen und zu verfolgen (Lepetit und Fua 2005a). Der vorliegende Anwendungsfall unterscheidet sich allerdings in mehrerlei Hinsicht von den bereits in der Literatur verfügbaren. Viele Methoden schränken das Umfeld stark ein, im Sportbereich wird beispielsweise ein definiertes Spielfeld beobachtet. Daraus ergeben sich meist definierte Lichtverhältnisse, konstante oder nur leicht veränderliche Größenverhältnisse und insbesondere nur leicht variable Beleuchtungsbedingungen. Unter freiem Himmel bringt der schneebedeckte Boden allerdings starke Reflexionen und, durch Überbelichtung und kurze Belichtungszeit bedingt, geringen Kontrast mit sich. Aufgrund der resultierenden schlechten Farbwiedergabe scheiden histogrammbasierte Ansätze weitgehend aus (Mauthner und Bischof 2007). Darüber hinaus ist die Objektkontur je nach Perspektive starken Veränderungen unterworfen, was darauf basierende Verfahren in Kombination mit auftretenden Schneefahnen, und insbesondere deren Schatten, ausschließt. Die Berechnung eines optischen Flusses<sup>1</sup> erzielt zwar gute Detektionsergebnisse, ist allerdings aufwendig und daher für HD-Auflösung<sup>2</sup> nur schwer zu realisieren. Aus diesen genannten Gründen wurde eine Methode gesucht, die sowohl rechnerisch als auch hinsichtlich der Datenabhängigkeiten geringe Komplexität aufweist,

---

<sup>1</sup> Der optische Fluss bezeichnet ein Vektorfeld, das die Bewegung für jeden Bildpunkt einer Bildsequenz angibt

<sup>2</sup> HD-Auflösung bezeichnet hier die von der Kamera generierten Formate HDTV 1080i (1920 x 1080 Pixel, interlaced) und HDTV 720p (1280 x 720 Pixel, progressive)

was eine massive Parallelisierung in der Verarbeitung begünstigt. Es muss festgehalten werden, dass das hier erarbeitete Verfahren nicht vollständig wissenschaftlich ergründet und optimiert wurde, sondern vielmehr einen Anstoß für weitere Forschung darstellt.

## 4.2 Bildverarbeitung

Bildverarbeitung stellt große Ansprüche an die Verarbeitungsleistung, sofern man Bilder hoher Auflösung mit einer gewissen Bildrate in Echtzeit bearbeitet. Oft muss man dazu auf spezielle Hardware ausweichen oder aber eine gezielte Anpassung des Sourcecodes auf die Zielarchitektur vornehmen. In dieser Arbeit wird durch gezielte Exposition von Parallelismus, selektive Bildverarbeitung und komplexitätsarme Algorithmen ein entsprechender Durchsatz erreicht.

Zunächst erfolgt die Separation von Vorder- und Hintergrund, wonach durch Differenzbildung auf den Hintergrund ein Bild der aktuell beweglichen Objekte entsteht. Durch Schwellwertentscheidung auf Pixelebene entsteht sodann eine Binärmaske, die rudimentär veränderliche Bereiche einer gewissen Intensität einschließt. Folglich muss diese Maske von Störungen bereinigt und optimiert werden. Auf Grund der Tatsache, dass Schnee als hellstes Objekt angenommen werden kann, ist es möglich, durch Ausblenden einer Schlüsselfarbe und eines gewissen Toleranzbereiches um diese (Keying) Störungen durch Schneewolken zu entfernen. Bildrauschen, Tore und Schatten können zum Teil durch morphologische Operationen auf der Binärmaske korrigiert werden. Schließlich werden die so entstehenden Pixelwolken zu Objekten zusammengefasst und soweit abstrahiert weiter verarbeitet. Abbildung 4.1 zeigt die genannten Schritte in einzelne Filter organisiert, die wiederum zu einer als Filtergraph<sup>3</sup> bezeichneten Verarbeitungskette zusammengefasst wurden.

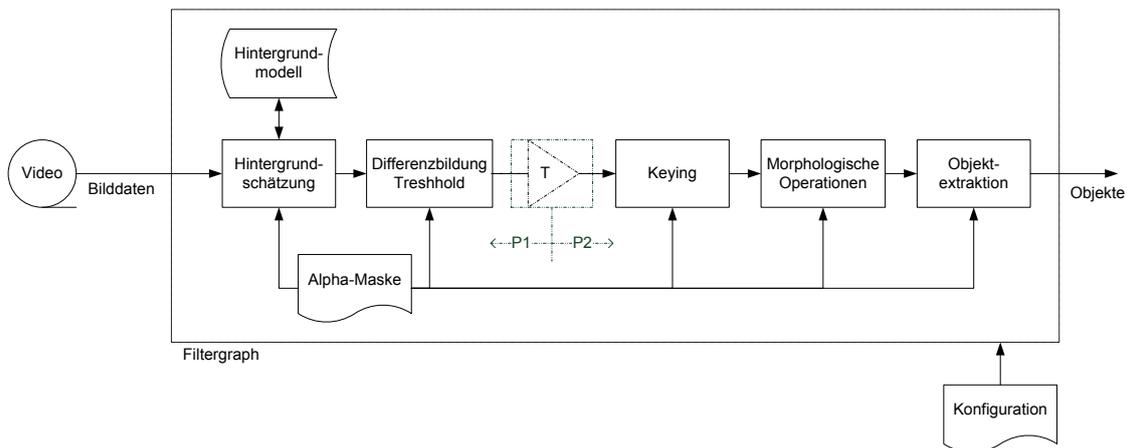


Abbildung 4.1: Bildverarbeitung

<sup>3</sup> Details zur Umsetzung werden in Kapitel 5 behandelt.

### 4.2.1 Laufzeitoptimierung

Für die Laufzeitoptimierung der gesamten Filterkette können drei Methoden angewandt werden. Zu allererst lässt sich der zu verarbeitende Bildbereich auf die Region von Interesse einschränken. Dies ist in Abbildung 4.1 durch die Alpha-Maske angedeutet, die jeden Bildpunkt von der Bearbeitung entweder ein- oder ausschließt. Da die einzelnen Verarbeitungsschritte oder auch Filter keine rekursiven Datenabhängigkeiten aufweisen, können sie in Form einer Pipeline abgearbeitet werden, im weiteren Sinne also parallel zueinander für aufeinanderfolgende Bilder. Direkte Unterstützung durch das Betriebssystem gewährleistet die Verwendung von Threads für parallel ablaufende Aktivitäten. Aus diesem Grund wird die Pipeline durch modular einfügbare Threadentkopplungen realisiert, wie in Abbildung 4.1 dargestellt. Da die gesamte Verarbeitungszeit pro Bild minimal die Rechenzeit der Stufe mit der längsten Laufzeit sein kann, ist zusätzlich für aufwendige Bearbeitungsprozesse eine Parallelisierung innerhalb der Filter notwendig. Diese lässt sich leicht durch Aufteilung des Bildbereiches erreichen, verarbeitungstechnisch erweist sich eine vertikale Unterteilung als günstig. Abbildung 4.2 zeigt die Einschränkung des Bildbereiches bei gleichzeitiger Zuordnung zu 4 Verarbeitungseinheiten (Bereiche A1 – A4).

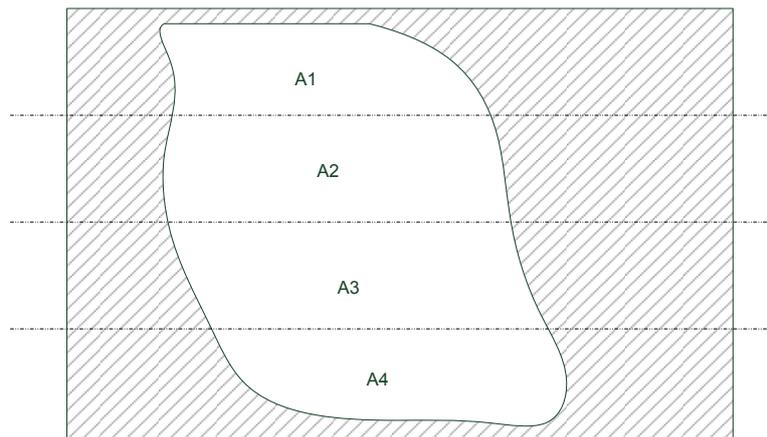


Abbildung 4.2: Bildaufteilung

### 4.2.2 Hintergrundschätzung

In der Literatur findet sich eine Vielzahl von Hintergrundschätzungsverfahren (Piccardi 2004), dazu zählen Mittelwertbildung, Kerndichteschätzung (KDE<sup>4</sup>) oder auch Eigenbackgrounds. Da der aktive Verarbeitungsbereich auf die zu beobachtende Geländefläche eingeschränkt ist und eine Kamerabewegung weitgehend auszuschließen ist, kann ein sehr einfaches Hintergrundmodell verwendet werden. Die rasche Bewegung einzelner Objekte vor einem statischen Hintergrund ermöglicht das schrittweise Annähern der Intensitätswerte des temporären Hintergrundes an die des aktuellen Kamerabildes. Die Updatevorschrift

<sup>4</sup> Kernel density estimation

lässt sich mit Hilfe der Farbtintensitäten  $I_B$  des Hintergrundes und  $I_F$  des Vordergrundes wie folgt anschreiben.

$$I_{B,t+1} = \begin{cases} I_{B,t} + 1 & I_{F,t} > I_{B,t} \\ I_{B,t} - 1 & I_{F,t} < I_{B,t} \\ I_{B,t} & I_{F,t} = I_{B,t} \end{cases} \quad (4.1)$$

### 4.2.3 Keying

Als Keying bezeichnet man das Entfernen von Bildelementen mit einer bestimmten Schlüsselfarbe beziehungsweise mit einem Farbwert in einem Toleranzbereich um diese. Das Verfahren ist aus Film- und Fernsehtechnik bekannt und wird hier in einer modifizierten Form angewendet. Bereiche mit einem entsprechenden Farbwert werden nicht direkt ausgeblendet, sondern in der dem Bild zugehörigen Binärmaske für die Objekterkennung ausgeschlossen. Dies ermöglicht die einfache Entfernung von Störungen, die in einem bekannten Farbbereich liegen, der sich von dem des Objektes von Interesse unterscheidet. So erscheint beispielsweise Schnee unter klarem Himmel auf Grund der Reflexionen leicht bläulich und wird oft überbelichtet, was einen deutlichen farblichen Kontrast zu Skifahrern bildet. Selbst Überschneidungen der Farbbereiche wirken sich wegen der nachfolgenden morphologischen Operationen nicht negativ auf die Erkennungsrate aus. Zur Entfernung von Schatten lässt sich diese Methode in unterschiedlichen Farbräumen anwenden; insbesondere der der menschlichen Wahrnehmung nachempfundene HSV<sup>5</sup>-Farbraum eignet sich dazu.

### 4.2.4 Morphologische Operationen

Zur Reduktion von Störungen wie Bildrauschen eignen sich morphologische Operationen (Steinmüller 2008). Über ein frei definierbares Strukturelement kann Einfluss auf die Form einer binären Pixelanordnung genommen werden. Entsprechende Filter sind nichtlinear und beschreiben Mengenoperationen zwischen dem Strukturelement und einem binären Bild. Die zwei Basisoperatoren sind Dilation<sup>6</sup> und Erosion<sup>7</sup>, aus denen sich komplexere Operationen zusammensetzen lassen. Die Anwendung eines strukturierenden Elementes  $B$  auf ein binäres Bild  $A$  lässt sich mathematisch wie folgt ausdrücken, wobei  $(B)_x$  die Translation von  $B$  um  $x$  bezeichnet,  $\hat{B}$  ist die Spiegelung von  $B$  um den Ursprung (Rodriguez und Ayala 2001).

$$\begin{aligned} \text{Erosion} \quad A \ominus B &= \{x | (B)_x \subseteq A\} \\ \text{Dilation} \quad A \oplus B &= \{x | (\hat{B})_x \cap A \neq \{\}\} \\ \text{Closing} \quad A \oplus_B X &= (A \oplus X) \cap B \end{aligned} \quad (4.2)$$

<sup>5</sup> Hue, Saturation, Value

<sup>6</sup> lat. dilatare: erweitern, ausdehnen

<sup>7</sup> lat. erodere: abnagen

Dilation gefolgt von Erosion schließt Figuren und entfernt leichtes Rauschen (morphologischer Closing-Operator). Zur Reduktion von starken Schlagschatten wird hier die Kombination Erosion - Dilation - Erosion gewählt.

#### 4.2.5 Objektextraktion

Zur einfacheren Verarbeitung werden einzelne Objekte in einem letzten Schritt mit Hilfe der Binärmaske aus dem Bild extrahiert. Die Zuordnung einzelner Pixel zu Objekten wird in der Literatur als Connected component labeling (Di Stefano und Bulgarelli 1999) bezeichnet. Sind zusammenhängende Objekte eindeutig identifiziert, können ihre umschließenden Rechtecke bestimmt werden, die als neue Bezugssysteme die Grundlage für weitere Verarbeitungsschritte bilden. Das somit beschnittene Originalbild gemeinsam mit dem entsprechenden Teil der Binärmaske zur Erhaltung der Objektsilhouette bildet das Ausgabedatum des Bildverarbeitungsbereiches.

### 4.3 Objektverarbeitung

Neben einfachen Filteraufgaben über Histogramme und Größenverhältnisse steht in der Objektverarbeitung die Bestimmung des Fußpunktes sowie die Zuordnung eines Objektes zu seinen jeweiligen Entsprechungen über mehrere Bilder hinweg im Vordergrund. In dieser Arbeit wird erstere näher behandelt.

#### 4.3.1 Bildmomente

Momente von Helligkeitsverteilungen sind in der Bildverarbeitung ein wichtiges Klassifikationsmittel (Teh und Chin 1988; Chaumette 2004). Durch geeignete Wahl der Gewichtung lassen sich geometrische Interpretationen ableiten, allen voran Fläche, Schwerpunkt und Ausrichtung von Pixelwolken. Nichtzentrierte Momente der zweidimensionalen, diskreten Intensitätsverteilung  $I(x, y)$  eines Grauwertbildes berechnen sich wie folgt.

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \quad (4.3)$$

Aus diesen lassen sich die Fläche ( $A$ ) und der Schwerpunkt ( $\zeta$ ) ableiten. Letzterer ist für die Berechnung zentraler Momente  $\mu_{pq}$  notwendig.

$$A = M_{00}$$

$$\zeta = \begin{bmatrix} x_\zeta \\ y_\zeta \end{bmatrix} = \begin{bmatrix} \frac{M_{10}}{M_{00}} \\ \frac{M_{01}}{M_{00}} \end{bmatrix} \quad (4.4)$$

$$\mu_{pq} = \sum_x \sum_y (x - x_\zeta)^p (y - y_\zeta)^q I(x, y) \quad (4.5)$$

Über die normierten zentralen Momente lässt sich die Kovarianzmatrix der Intensitätsverteilung aufstellen. Die Eigenwerte dieser Matrix entsprechen den beiden Hauptrichtungen der Kovarianzellipse, aus deren Rotation zum Bezugskordinatensystem die Orientierung des Objektes berechnet werden kann.

$$\mu'_{pq} = \frac{\mu_{pq}}{\mu_{00}}$$

$$\text{cov}[I(x, y)] = \begin{bmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{bmatrix} \quad (4.6)$$

Der Rotationswinkel  $\psi$  der Hauptachse entspricht dem Winkel des Eigenvektors mit dem größten Eigenwert.

$$\psi = \frac{1}{2} \arctan \left( \frac{2 \mu'_{11}}{\mu'_{20} - \mu'_{02}} \right) \quad (4.7)$$

Neben diesen Charakteristiken können skalierungs- und rotationsinvariante Momente zur Objektverfolgung eingesetzt werden. Dies ist Gegenstand zukünftiger Entwicklungen.

### 4.3.2 Gewinnung des Fußpunktes

Aufgrund der deutlichen Exzentrizität des menschlichen Körpers kann der Fußpunkt im Stand über den Schnittpunkt der Rotationsachse mit der Fußlinie angenähert werden. Da die Orientierung des Objektes allerdings nicht immer zuverlässig feststellbar ist, lässt sich zur Fußpunktbestimmung auch der auf die Fußlinie projizierte Schwerpunkt heranziehen. Für die Positionsbestimmung eines Skifahrers erwies sich ein gewichteter Mittelwert aus beiden Größen am geeignetsten, der den Einfluss der Rotationsachse mit steigender Distanz zum projizierten Schwerpunkt reduzierte.

### 4.3.3 Objektverfolgung

Für eine erste Exploration des Entwurfraumes wurde die Zuordnung von Objekten über mehrere Bilder und Kameras hinweg im bereits rückprojizierten Objektraum vorgenommen. Durch Bestimmung der idealen Fahrlinie kann die Bewegung des Skifahrers eindimensional auf eben dieser modelliert werden (siehe Abbildung 4.3). Daraus resultiert auch eine Filterung nach zulässigen Objekten gemäß ihrer Distanz zu dieser Linie. Für die Prädiktion von Positionswerten wurde ein Kalman-Filter eingesetzt, wobei die Beschleunigung als Rauschen angenommen wurde. Trotz dieser rudimentären Modellierung, die weder den Streckenverlauf noch das dynamische Verhalten des Rennläufers einschloss, konnten bereits gute Ergebnisse erzielt werden, wenngleich in diesem Bereich weitere Forschungen zur Steigerung der Zuverlässigkeit notwendig sind.

## 4.4 Ergebnisse

Zur Illustration wurde eine Bildsequenz des Streckenabschnitts Lärchenschuss gewählt. Abbildung 4.4 zeigt einzelne Bilder dieser Sequenz, die mit den extrahierten Pixelwolken

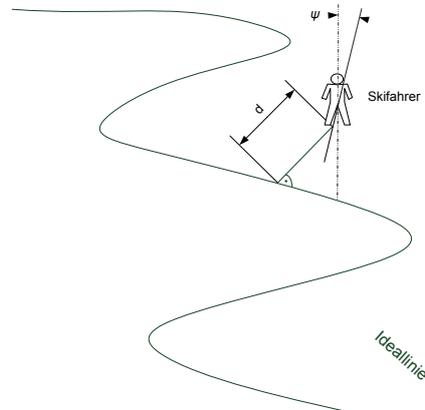


Abbildung 4.3: Ideallinie

(grün) und den daraus resultierenden Objekten (rot) überlagert wurden. Die Bilder (b) und (e) sind in den Abbildung 4.5 und 4.6 in ihren Verarbeitungsschritten dargestellt. Bild (a) zeigt hierbei jeweils den relevanten Ausschnitt des Ausgangsbildes, (b) das zu diesem Zeitpunkt adaptiv ermittelte Hintergrundbild, (c) das Differenzbild der beiden letzten in verstärkter Darstellung, (d) die Binärmaske nach der Schwellwertbildung, (e) das extrahierte Objekt und schließlich (f) die berechneten Objektmerkmale wie projizierten Schwerpunkt und Rotation. Filtert man Fehlerkennungen, wie sie beispielweise im Super-G durch sich bewegende Tore entstehen (Abb. 4.5), lässt sich mit der vorgestellten Methode eine sehr gute Erkennungsrate erzielen. Einzig unter starker Sonneneinstrahlung führen die Schatten der durch die Skifahrer erzeugten Schneewolken zu Problemen, da sie durch Keying auf Grund der Farbähnlichkeit mit dem Skifahrer nicht eliminiert werden können.

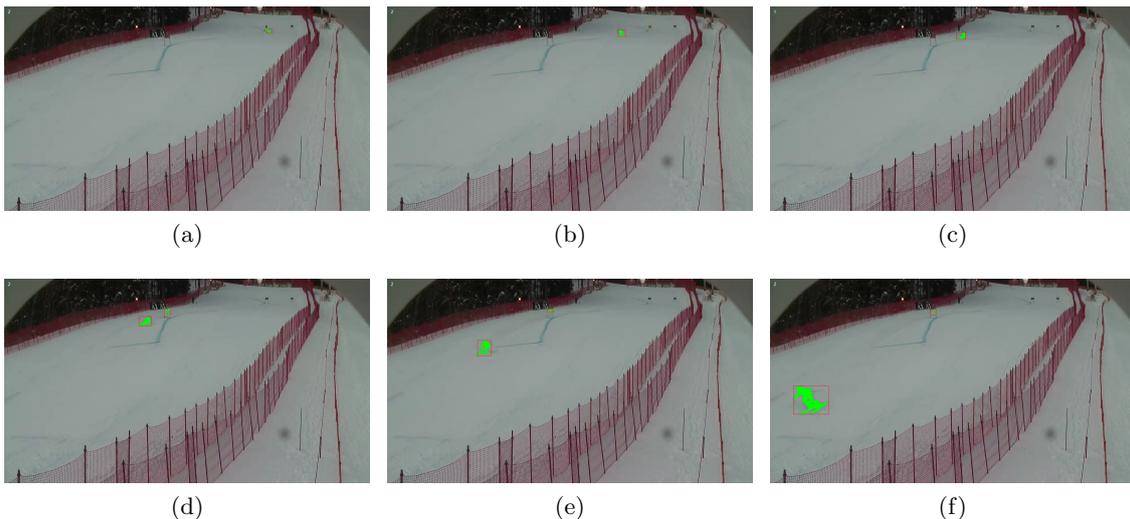


Abbildung 4.4: Objekterkennung auf dem Streckenabschnitt „Lärchenschuss“, Bildsequenz

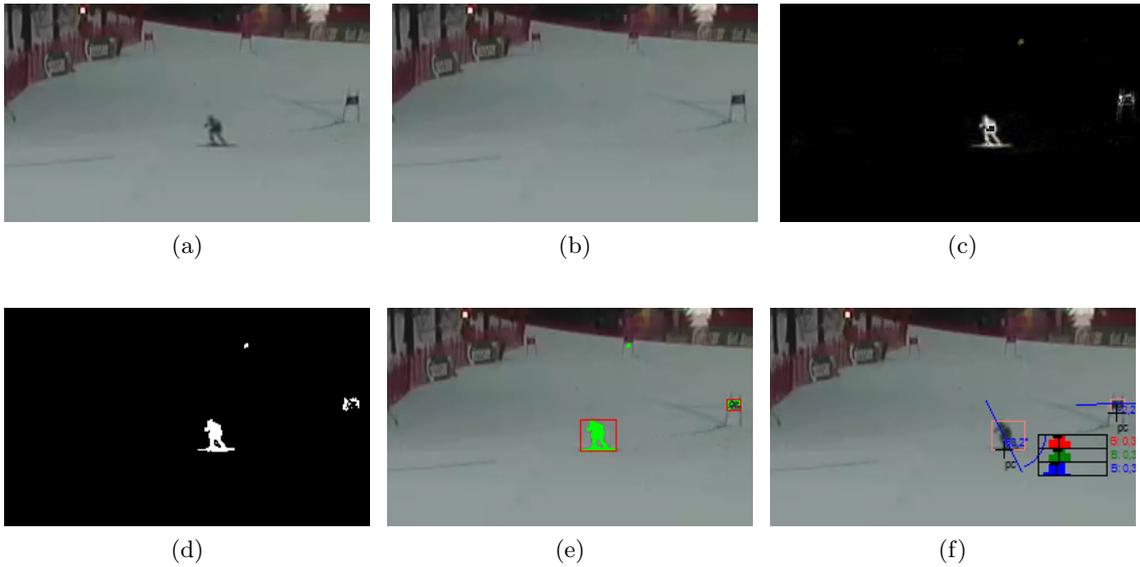


Abbildung 4.5: Objekterkennung auf dem Streckenabschnitt „Lärchenschuss“, Bild (b)

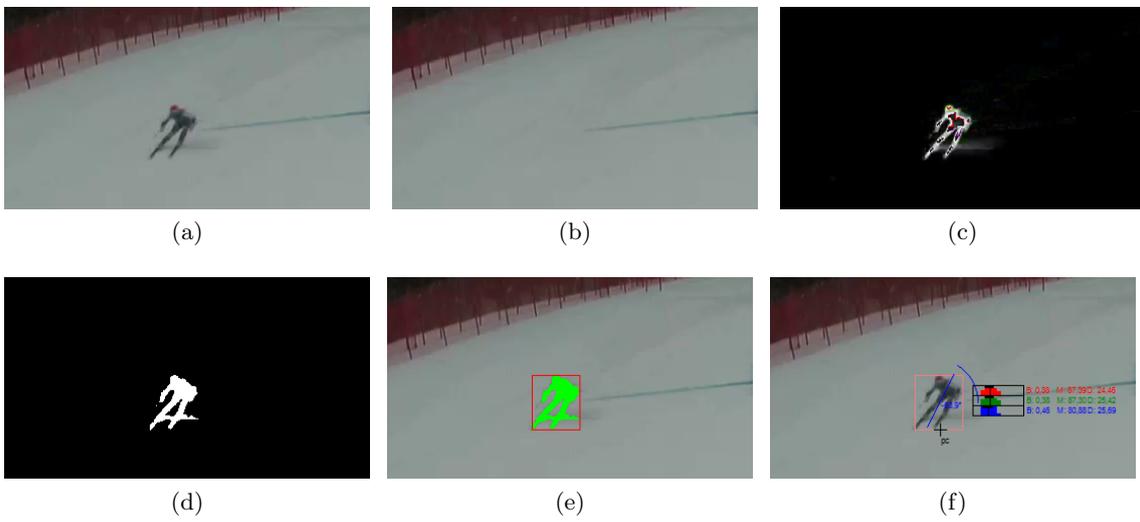


Abbildung 4.6: Objekterkennung auf dem Streckenabschnitt „Lärchenschuss“, Bild (e)

# Kapitel 5

## Integration

Dieses Kapitel beschäftigt sich mit der Zusammenführung, Implementierung und Optimierung der erarbeiteten Komponenten sowie dem Aufbau des Gesamtsystems. Dies schließt eine flexible Modularisierung, optimierte Datenübertragung, Auswahl der Entwicklungsumgebung und letztendlich auch die Entscheidung über entsprechende Zielhardware mit ein. Abschließend dient der reale Einsatz des ersten Prototypen der Verifikation des Entwurfes und der Validierung jener Überlegung, die den Anstoß zu dieser Arbeit gegeben hat.

### 5.1 Anforderungen

Im Umfeld von Sportveranstaltungen sind ab einer gewissen Bewerbsgröße eine Reihe von Anforderungen zu erfüllen, insbesondere, wenn die Veranstaltung durch den Rundfunk direkt übertragen wird. Diese lassen sich in die folgenden Gruppen grob unterteilen.

#### **Echtzeitfähigkeit**

Das System muss gewisse Antwortzeiten einhalten, um eine graphische Einblendung im Fernsehbild synchron zum Geschehen halten zu können. Die maximale Verzögerung wird hier durch die Bildrate der Übertragung bestimmt, wobei diese auf Grund der Verarbeitungszeit oftmals nicht eingehalten werden kann. Wichtig ist somit, die visuelle Wahrnehmung einer Verzögerung auszuschließen, was bei einer Verarbeitungszeit von unter einer Zehntelsekunde für den ungeübten Beobachter gewährleistet ist.

#### **Verarbeitungsrate**

Ein weiterer entscheidender Aspekt ist die Aktualisierungsrate der Ein- und Ausgabedaten. Die Verarbeitung bzw. die einzelnen Verarbeitungsstufen bei einer Pipelinestruktur müssen auf jeden Fall vor dem Eintreffen des nächsten Eingabedatums alle Vorgänge abgeschlossen haben, um keine Pufferüberläufe zu verursachen.

Bei der Geschwindigkeit eines Rennläufers von 100 km/h legt dieser in einer Zehntelsekunde ca. 2,8 m zurück, was je nach Entfernung zur aufzeichnenden Kamera im Bereich weniger Pixel liegt. Ist die Distanz im Bild zu groß, lässt sich die Objektzuordnung nur mehr schwer durchführen, ist sie zu klein, kann die Geschwindigkeit durch die Rasterung

nicht mehr zuverlässig berechnet werden. Beachtet man auch die zu verarbeitende Bildgröße von  $1280 \times 720$  Pixel, ist eine Rate von 10 Bildern pro Sekunde anstrebenswert.

### **Modularität und Wartbarkeit**

Da das System in seinem Endausbau für unterschiedliche Sportarten eingesetzt werden soll, ist bereits im Entwurf auf einen modularen Aufbau zu achten, um Schlüsselmodule für unterschiedliche Anwendungen anpassen und austauschen zu können. Dies erleichtert nicht nur die Entwicklung sondern auch die Wartbarkeit, da neue Forschungsergebnisse rasch umgesetzt und getestet werden können, was wiederum schnellere Produktzyklen ermöglicht.

### **Konfigurierbarkeit**

Gerade der Bereich Bildverarbeitung muss spezifisch an die Gegebenheiten vor Ort angepasst werden können. Dies erfordert die Verwaltung unterschiedlicher Vorlagen für entsprechende Einsatzgebiete, die adaptiert und neu abgelegt werden. Die Konfiguration selbst bezieht sich hierbei auf unterschiedliche Detailebenen, vom Gesamtsystem bis zu einzelnen Filterparametern.

### **Steuerbarkeit**

Im Fehlerfall muss es möglich sein, auf das System korrigierend einzuwirken, um fehlerhafte Ausgabedaten möglichst schnell unterdrücken und den korrekt operativen Zustand wieder herstellen zu können. Weiters ist damit auch eine Visualisierung der gewonnenen Daten verbunden, was eine rasche Überprüfung und Beurteilung ermöglicht. Für die Konfiguration ist eine zentrale Steuerkomponente unverzichtbar. Nur so kann auf wechselnde Bedingungen reagiert und eine kurze Aufbauzeit gewährleistet werden.

### **Ausfallsicherheit**

Schließlich muss das System sowohl stabil gegenüber äußeren Einflüssen also auch gegenüber internen Fehlern sein. Die zentralen Komponenten sollten aus diesem Grund immer zumindest doppelt ausgeführt, Wiederanlaufpunkte definiert und gesichert und entsprechende Hardwarekomponenten als Ersatz in Reserve gehalten werden.

## **5.2 Datenabhängigkeiten**

Ausgangspunkt für den weiteren Entwurfsprozess bilden die inhärenten Datenabhängigkeiten der zur Lösung der Gesamtaufgabe benötigten Komponenten. Sie gliedern sich in die Bereiche Kamerakalibrierung, Modellierung des Geländeprofiles, Tiefenrekonstruktion und schließlich Bild- und Objektverarbeitung. Abbildung 5.1 zeigt die Prozessstruktur, wobei sich zwei Bereiche deutlich unterscheiden lassen: Lediglich die grau hinterlegten Komponenten müssen Echtzeitfähigkeit bieten, alle übrigen Prozesse können im Vorfeld der Echtzeitverarbeitung durchgeführt werden. Dieses Modell zeigt bereits die entsprechenden Schnittstellen zwischen den einzelnen Komponenten.

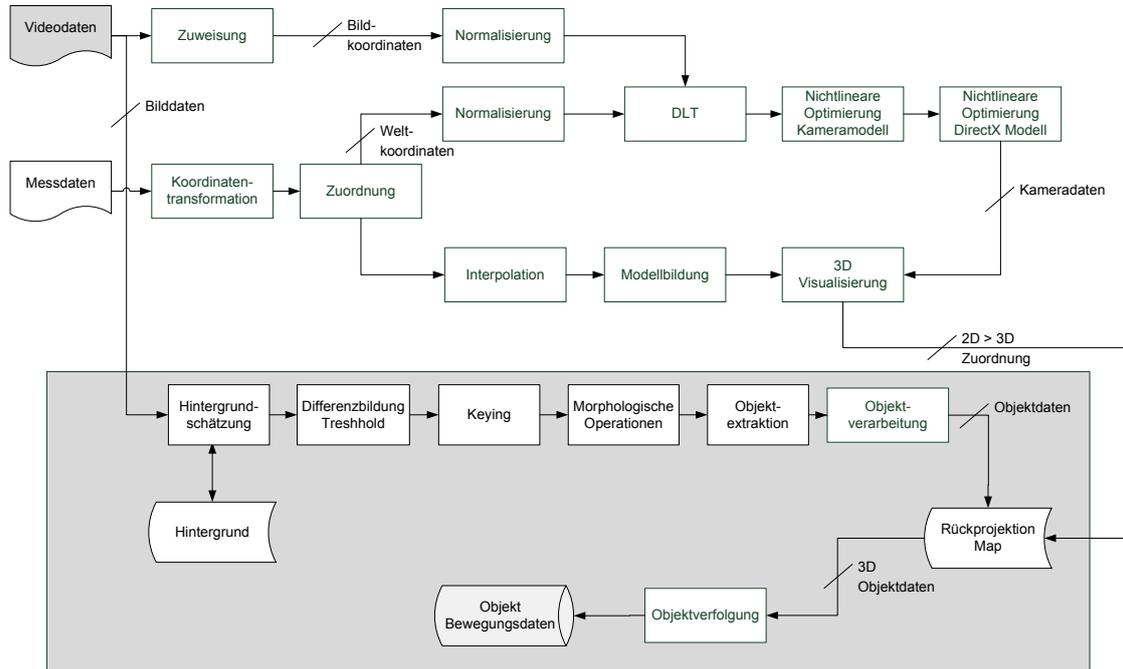


Abbildung 5.1: Prozessablauf und Datenabhängigkeiten des Gesamtsystems

### 5.3 Logische Verarbeitungseinheiten

In einem weiteren Schritt werden nun die einzelnen Komponenten zu Modulen zusammengefasst, die logischen Verarbeitungseinheiten entsprechen. Im Bereich der Vorverarbeitung sind drei Gruppen maßgebend, Kamerakalibrierung, Erzeugung des Oberflächenmodells und schließlich, resultierend aus beiden, die Zuordnung von zweidimensionalen Bildpunkten auf dreidimensionale Welt- oder Objektpunkte (siehe Abbildung 5.2). Für diese Komponenten werden einerseits die erfassten Messdaten der Kalibrier- und Modellpunkte benötigt, andererseits ist auch ein Kamerabild je Kalibrierpunkt bzw. eine Videoaufzeichnung der Kalibrierung notwendig, um Koordinatenpaare zuordnen zu können.

#### Bild- und Objektverarbeitung

Nach direkter linearer Transformation, nichtlinearer Optimierung, Modellaufbau und dreidimensionaler Visualisierung kann schließlich für jeden Bildpunkt des Kamerabildes ein entsprechender Objektpunkt auf der Modelloberfläche berechnet werden, sofern er denn existiert. Diese Daten werden in einer Tabelle abgelegt und gemeinsam mit der kameraspezifischen Konfiguration der dem Bereich zugeordneten Verarbeitungseinheit zur Verfügung gestellt. Deren zentrales Element bildet der sogenannte Filtergraph, der eine Umgebung für sowohl Bild- als auch Objektverarbeitung darstellt.

## Konfiguration

Insgesamt 4 Grundkomponenten stehen für den Aufbau desselben zur Verfügung: Bildquellen, Bild-zu-Bild-, Bild-zu-Objekt- sowie Objekt-zu-Objekt-Filter. Filterparameter und Reihenfolge der einzelnen Komponenten sind beliebig konfigurierbar, sofern aneinander grenzende Schnittstellen kompatibel sind. Als Austauschformat für den Aufbau der Filterstruktur sowie für das Setzen einzelner Parameter wurde XML<sup>1</sup> gewählt.

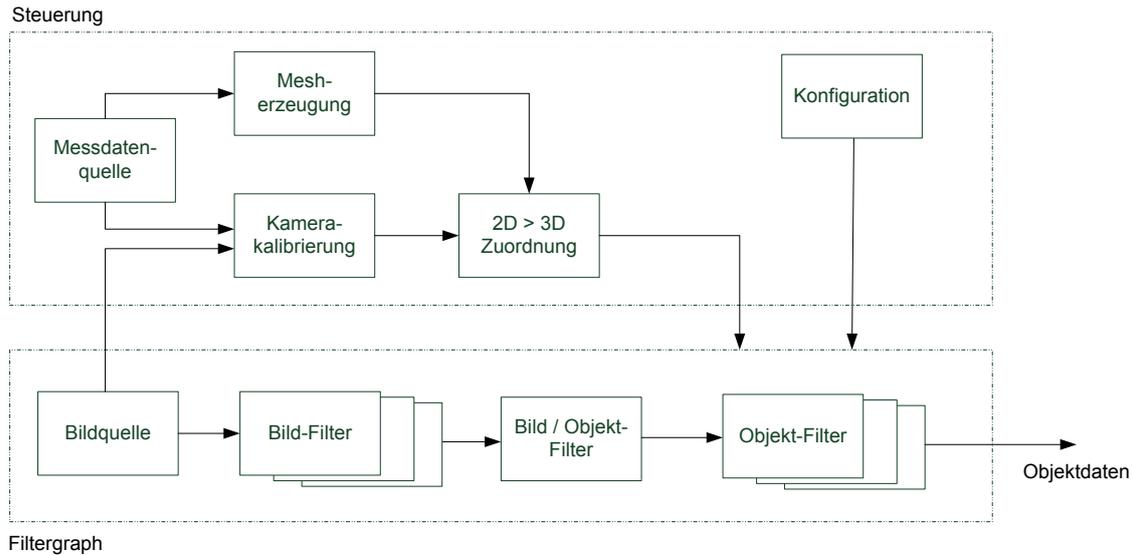


Abbildung 5.2: Logische Verarbeitungseinheiten

## Zentrale und dezentrale Verarbeitung

Bislang wurde nur von einer Bildquelle ausgegangen; für die konkrete Anwendung müssen allerdings mehrere Abschnitte parallel verarbeitet werden. Logisch existiert diese Trennung aufgrund der perspektivischen Projektion nur im Bildraum. Sie kann daher aufgegeben werden, sobald Objektdaten in Weltkoordinaten vorliegen. Eine derartige Aufteilung erweist sich als vorteilhaft, da bildverarbeitende Prozesse erfahrungsgemäß weit höhere Rechenleistung benötigen und gleichzeitig die zu verarbeitende Datenmenge mit steigender Abstraktion sinkt. Es ist daher sinnvoll, den ersten Teil des Filtergraphen je Kamera dezentral logisch nahe an derselben anzuordnen, den zweiten jedoch zentral für alle Bildquellen abzuarbeiten. Diese Architektur (siehe Abbildung 5.3) erleichtert kameraübergreifende Operationen deutlich und sorgt gleichzeitig für entsprechende Lastverteilung.

Das Modell einer zentralen Verarbeitungseinheit birgt darüber hinaus den Vorteil in sich, dass elementare Steueraufgaben, wie beispielsweise Athletenverwaltung über einen zentralen Knotenpunkt abgewickelt werden können.

<sup>1</sup> Extensible Markup Language (engl.: erweiterbare Auszeichnungssprache)

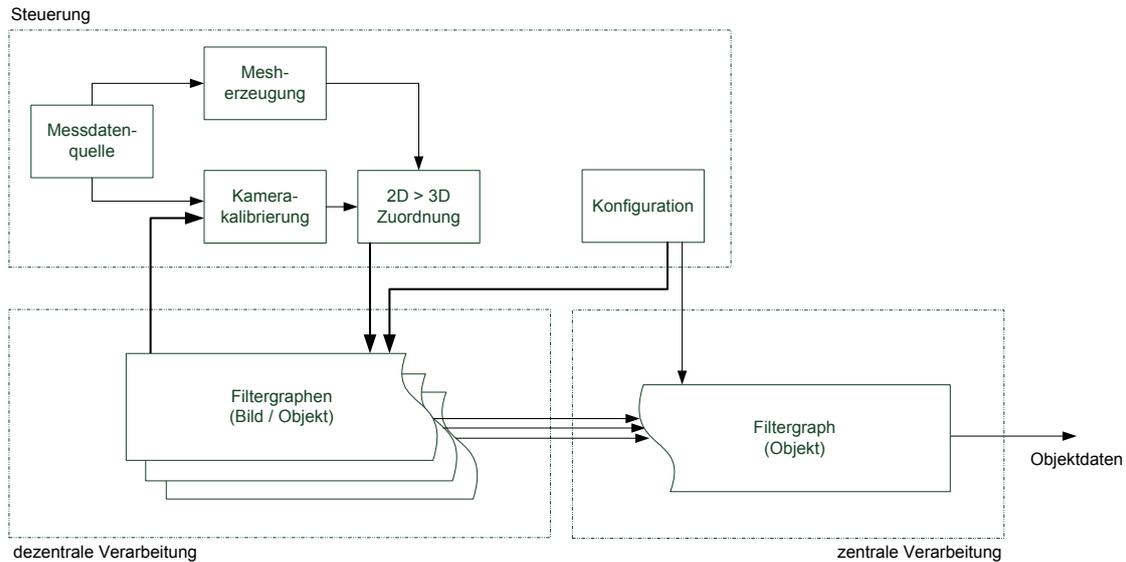


Abbildung 5.3: Dezentrale logische Verarbeitungseinheiten

## 5.4 Entwicklungsumgebung

Die Auswahl der Entwicklungsumgebung und damit verbunden auch der Programmiersprache richtet sich in erster Linie nach den Eigenschaften des geplanten Systems. Unter Berücksichtigung der in Abschnitt 5.1 erwähnten Anforderungen lässt sich rasch eine Eingrenzung durchführen. Die Modularität der Komponenten erfordert Objektorientierung, die geforderte Verarbeitungsrate hohe Performance und die Anforderungen hinsichtlich Flexibilität sowie Erweiterbarkeit eine komfortable Entwicklungsumgebung. Da das System aufgrund seiner vielseitigen Einsatzmöglichkeiten weitgehend hardwareunabhängig bleiben soll, ist eine Hochsprache vorzuziehen. C++, C#<sup>2</sup> und Java<sup>3</sup> fallen aus diesen Gründen in die engere Wahl. C++ bietet hohe Performance, allerdings nur mäßigen Komfort und verlängert somit gleichzeitig die Entwicklungszeit. C# und Java sind sehr ähnlich; C# ermöglicht allerdings durch die Integration von „unsafe code“ die Verwendung von Zeigern außerhalb der Common Language Runtime (CLR) und bietet so einen massiven Geschwindigkeitsvorteil in der Bildverarbeitung. Lutz und Laplante (2003) verglichen C++ und C# hinsichtlich ihrer Performance und Echtzeitfähigkeit und zeigten Probleme der .NET Umgebung für „harte“ Echtzeitbedingungen auf. Die hier geforderte „weiche“ Echtzeit lässt sich jedoch mit selbstständigem Memorymanagement und gezielten Optimierungen erfüllen. Der Performanceverlust ist vergleichsweise minimal, sodass sich die Verwendung von C# .NET in einem ersten Prototypen anbietet.

<sup>2</sup> C# ist eine von Microsoft im Rahmen der .NET-Plattform entwickelte Programmiersprache. C# ist bei ECMA und ISO als Standard registriert.

<sup>3</sup> Java ist eine Programmiersprache und eingetragenes Warenzeichen der Firma Sun Microsystems.

## 5.5 Physikalische Verarbeitungseinheiten

Die Struktur der physikalischen Verarbeitungseinheiten entspricht weitgehend dem logischen Aufbau. Ursprünglich war eine Erstverarbeitung direkt bei der Kamera zur Reduktion der Datenmenge geplant, was aber aufgrund der Unterstützung des H.264<sup>4</sup> Standard direkt durch die Kamera zugunsten einer physikalisch zentralen Verarbeitung aufgegeben wurde. Diese Entscheidung reduziert den dezentralen Hardwareaufwand enorm und erlaubt höhere Flexibilität in der Kamerapositionierung. Der Verlust an Bildqualität durch die Codierung, der in der weiteren Verarbeitung deutlich erkennbar ist, wird durch die gewonnene Stabilität und Verarbeitungsleistung ausgeglichen. Letztere kann kompakt und skalierbar durch Serverblades<sup>5</sup> zur Verfügung gestellt werden, wie in Abbildung 5.4 dargestellt. Für Steueraufgaben können handelsübliche Rechner eingesetzt werden.

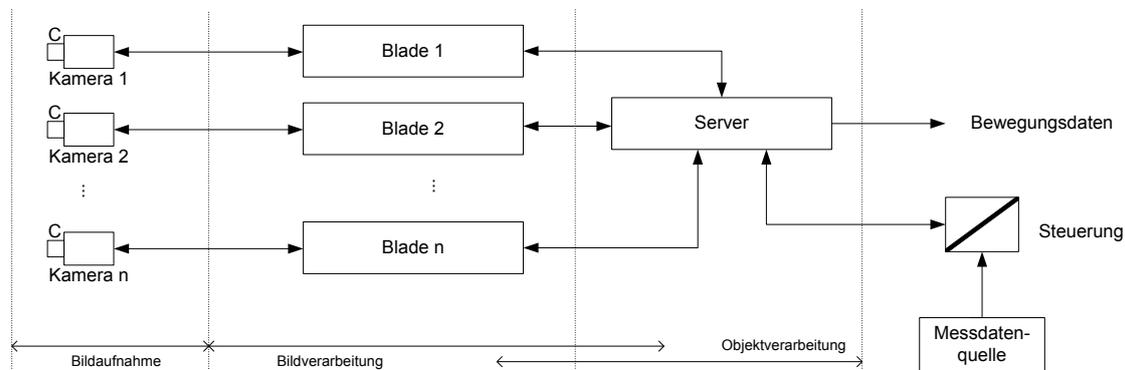


Abbildung 5.4: Hardwarestruktur

## 5.6 Umsetzung und Anwendung

Erstmals wurde das beschriebene System im Winter 2009 in Schladming (ORF „Das Rennen“) testweise mit nur einer Kamera eingesetzt. Die Ergebnisse dienten der letzten Optimierung für die Referenzanwendung, dem 70. Hahnenkammrennen 2010 in Kitzbühel im Rahmen des alpinen Skiworldcups. Die hier präsentierten Daten und Ergebnisse beziehen sich auf diese Veranstaltung. Definiert wurden insgesamt 11 Erkennungsbereiche, 9 davon auf der Abfahrtsstrecke „Streif“ und weitere 2 auf dem Slalomhang „Ganslern“. Somit konnte eine Evaluierung für zwei sehr unterschiedliche Einsatzbereiche durchgeführt werden.

<sup>4</sup> H.264/MPEG-4 AVC ist ein von der Internationale Fernmeldeunion (ITU) und der Moving Picture Experts Group (MPEG) 2003 verabschiedeter Videokompressions-Standard, der eine sehr effiziente Codierung von hochauflösendem Videomaterial ermöglicht. (*ITWissen - Das große Online-Lexikon für Informationstechnologie*)

<sup>5</sup> Serverblades sind unabhängige Computersysteme mit Mikroprozessoren, Arbeitsspeicher, Netzwerkan-schlüssen oder LAN-Switches auf einem Motherboard, die modular in einem Baugruppenträger zu einem Bladeserver konfiguriert werden. (*ITWissen - Das große Online-Lexikon für Informationstechnologie*)

### 5.6.1 Zielhardware

Für die zentrale Datenverarbeitung wurde ein Bladesystem der Firma Hewlett-Packard verwendet, das aus 8 ProLiant BL280c G6 Server bestückt mit je 2 Intel Xeon<sup>6</sup> E5520 Prozessoren und je 2 × 2 GB Arbeitsspeicher bestand. Die genannten Prozessoren haben mit einer maximalen Leistungsaufnahme von 80 W eine Basistaktrate von 2,26 GHz bei vier Kernen und acht Threads. Der Speicherzugriff erfolgt mit maximal 8,5 GB/s pro Modul direkt vom Prozessorkern. Nur mehr für die Kommunikation mit der restlichen Peripherie wird der von Intel entwickelten QuickPath Interconnect - Bus (QPI) verwendet, ein Nachfolger des Front Side Bus, der eine Transferrate von 5.86 GT/s bidirektional unterstützt.

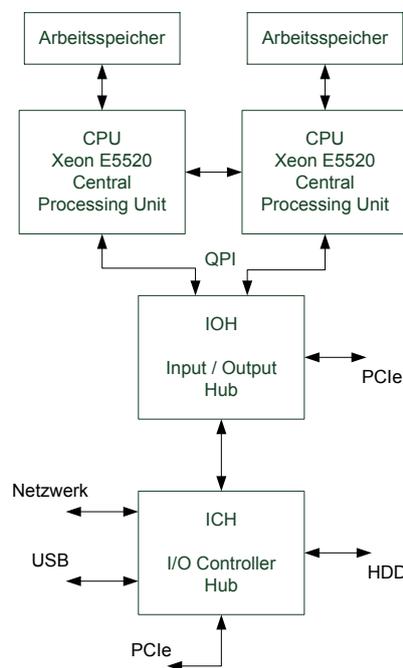


Abbildung 5.5: Intel Xeon E5520 Chipset

Auf diesem System konnte in Performancetests die geforderte Bildrate von 10 fps für zwei Kameras je Blade bei einer durchschnittlichen Gesamtauslastung von 70% erreicht werden. Bei Steigerung der Framerate kam es zu Einbrüchen in der Dekodierung des H.264-Datenstroms, deren Ursache Gegenstand zukünftiger Untersuchungen ist.

### 5.6.2 Kommunikation

Die größte Herausforderung in der Anwendung war die Sicherstellung zuverlässiger Kommunikationswege zwischen den einzelnen Komponenten. Im Kernsegment zwischen Bildverarbeitung und zentraler Objektverarbeitung konnte die Blade-interne Netzwerkkom-

<sup>6</sup> Intel und Xeon sind eingetragene Markenzeichen der Intel Corporation

munikation genutzt werden, da der Serverknoten einem der Blades zugeordnet wurde. Die Steuerung erfolgte über zwei abgesetzte Frontends, die gemeinsam mit Visualisierung und TV-Graphik über Gigabit-Ethernet angebunden werden konnten. Die Verbindung zu den einzelnen Kameras entlang der Strecke gestaltete sich durch die begrenzte Anzahl an Übertragungskanälen und deren unterschiedliche Typen ungleich schwieriger. Der untere Bereich der Abfahrt konnte direkt über verdrehte Zweidrahtleitungen der Zeitmessung erreicht werden, der obere Bereich über einen Lichtwellenleiter des ORF und Zweidraht- sowie Koaxialstrecken. Die Umsetzung erfolgte über entsprechende Modems bei einer nominellen Bandbreite von 200 Mbit/s, was zur Übertragung der komprimierten Bilddaten ausreichend war. Abbildung 5.6 zeigt die gesamte Verbindungsstruktur.

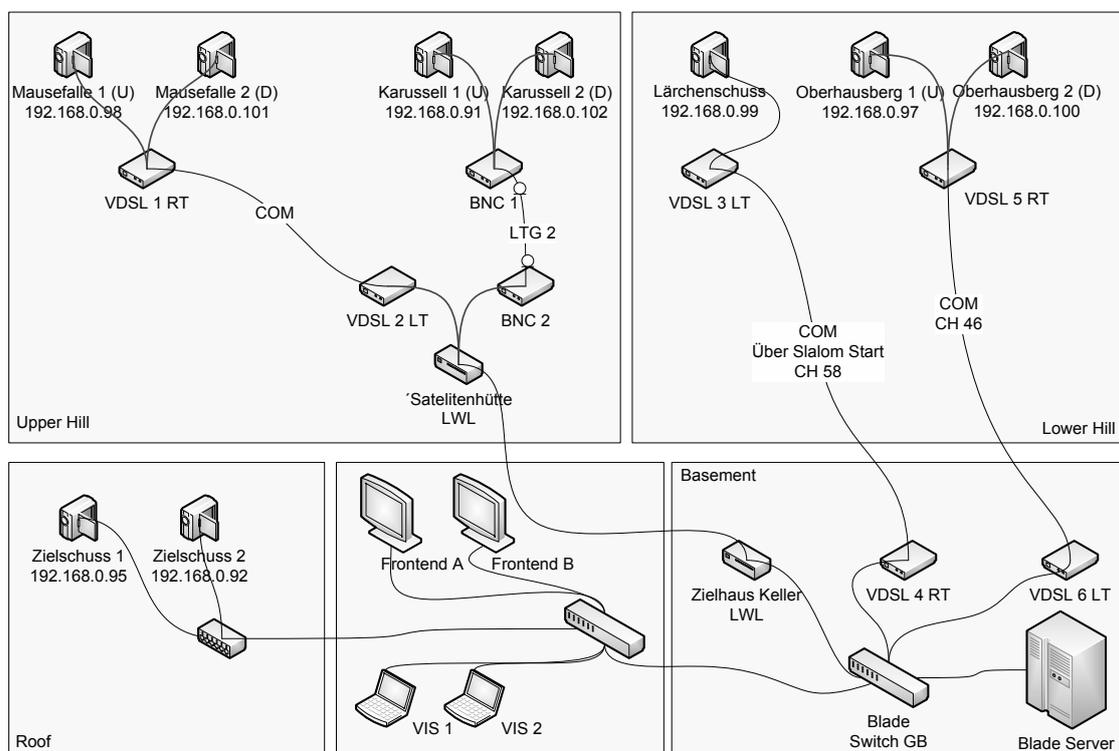


Abbildung 5.6: Systemaufbau Kitzbühel, Kommunikationswege

### 5.6.3 Vermessung und Modell

Für die Erfassung der Geländepunkte wurde ein RTK-GPS System von Trimble in Kombination mit dem APOS Onlinedienst verwendet (vgl. Abschnitt 3.4). Die Zuverlässigkeit war die höchste unter allen anderen im Vorfeld getesteten Systemen, wenngleich in manchen Regionen durch naheliegende Bäume keine RTK-fixed Position vermessen werden konnte (vgl. Abschnitt 3.4.2). Als nachteilig erwies sich die lange Initialisierungszeit, die vielfach über zwei Minuten lag und den Messprozess auf der Strecke deutlich erschwerte, da der tatsächliche Messzeitpunkt nicht klar bestimmbar war und in der Software des

Messgerätes falsche Punkte nicht gänzlich gelöscht werden konnten. Die Verbindung zum APOS-Dienst war aber trotz des alpinen Geländes durchwegs stabil.

Die aufgenommenen Randpunkte der Strecke sind in Abbildung 5.7 als interpolierte Linien eingezeichnet und kennzeichnen den Streckenverlauf. Die aus dem Kalibrierungsprozess rückgewonnen Kamerapositionen wurden in das ellipsoidische Koordinatensystem rückgerechnet und entsprechend dargestellt. Nur in zwei Fällen ergab sich im abgebildeten, ersten Kalibrierprozess eine deutliche Abweichung von der realen Kameraposition, die aber durch bessere Wahl der Kalibrierpunkte korrigiert werden konnte.

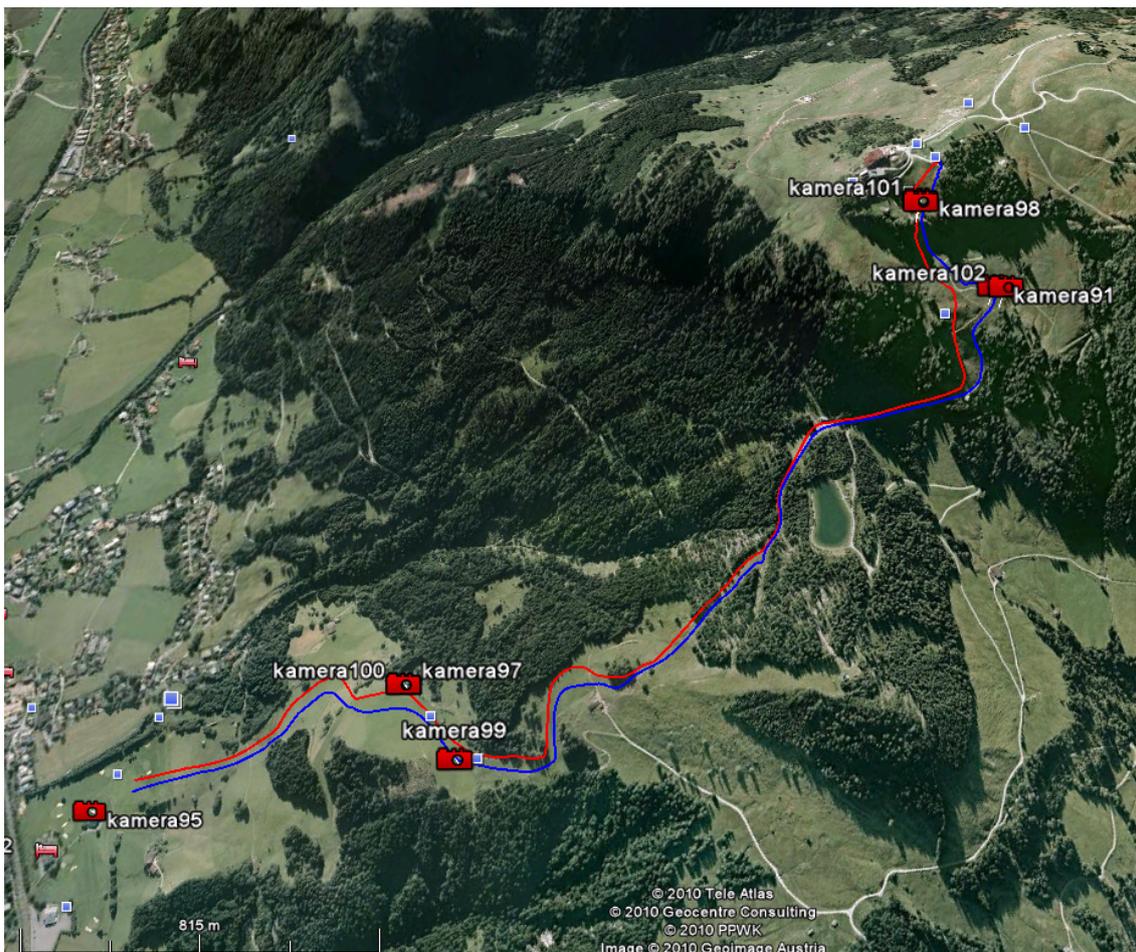


Abbildung 5.7: Vermessungs- und Kalibrierergebnisse für die Abfahrtsstrecke

Detaillierte Ergebnisse der Kalibrierung und Modellbildung sind in den Kapiteln 2 und 3 vermerkt.

### 5.6.4 Ergebnisse

Das Zusammenspiel aller Komponenten stellt die Objektverfolgung in Kombination mit dreidimensionaler Positionsermittlung dar. Sie erlaubt erstmals die Evaluierung des Gesamtsystems hinsichtlich der gestellten Vorgaben - und wenngleich es schwierig ist, ohne entsprechende Vergleichsdaten Aussagen über die Güte der resultierenden Positionsdaten zu treffen, kann doch eine Reihe von Parametern objektiv beurteilt werden.

#### **Laufzeit und Verzögerung**

Laufzeitmessungen der Verarbeitung wurden vom Eintreffen des Bildes bis zum Eingang der Positionsdaten in der Visualisierungsoberfläche durchgeführt. Die Gesamtverzögerung lag dabei im Mittel bei 40-50 ms und war damit im laufenden Betrieb deutlich unter den geforderten 100 ms. Die Verarbeitungszeit des langsamsten Filters in der Bearbeitungskette konnte durch Parallelisierung auf unter 15 ms gesenkt werden. Die Verzögerung durch Codierung und Decodierung des Bildes seitens der Kamera und des mitgelieferten Developmentkits beträgt mindestens 30 ms, kann allerdings auf Grund der fehlenden Zeitbasis nicht exakt bestimmt werden. Genaue Angaben sind aufgrund der massiven Parallelverarbeitung in diesem Entwicklungsstadium noch nicht möglich.

#### **Bildrate**

Die Analyse der im Bilddatenstrom vorhandenen Zeitstempel ergab, dass bei einer Sollrate von 10 fps abwechselnd nach 80 ms und 120 ms ein Bild durch die Kamera generiert wird, was als Jitter einen negativen Effekt auf zeitbasierte Nachverarbeitung hat. Insbesondere die Geschwindigkeitsberechnung ist davon betroffen und kann daher an dieser Stelle nicht zur wissenschaftlichen Analyse herangezogen werden. Dieser Effekt entfällt beim Betrieb mit 25 Bildern pro Sekunde.

#### **Fehlerbehandlung**

Es wurden zwei Verfahren integriert, um Fehler in der Objektzuordnung während der Laufzeit korrigieren zu können. Eine Methode war die Einführung von Bereichen, die der Skifahrer nacheinander passieren musste. Konnte er zwischen zwei dieser Sektionen nicht erkannt werden, war zumindest ab dem nächsten Bereich eine korrekte Zuordnung wieder automatisiert durchführbar. Diese Vorgehensweise ermöglichte auch den problemlosen Übergang zwischen Kameras und konnte mit Zeitmesssignalen gekoppelt werden. Die zweite Methode gestattete den manuellen Eingriff des Anwenders über die Benutzeroberfläche. Über einen berührungsempfindlichen Bildschirm konnten Objektzuordnungen entsprechend korrigiert werden, was in manchen Fällen allerdings zusätzliche Fehler in den Messdaten generierte.

#### **Kamerapositionen**

Die Auswertung der Ergebnisse zeigte deutlich schlechtere Erkennungsraten bei Kameras mit besonders flachem Sichtwinkel, was allerdings in Anbetracht der Bildrate und der rudimentären Bilderkennung zu erwarten war. Dazu kommt auch die Problematik schleifender Schnitte, die in diesem Fall bei der Gewinnung des Weltpunktes zwingend entstehen.

**Objektverfolgung**

Abgesehen von den genannten Problemen war die automatische Erkennung und Verfolgung von Objekten erfolgreich. Trotz der ungenauen Zeitbasis konnten auch Geschwindigkeit und Distanz zum Führenden in der Projektion auf die Ideallinie bestimmt werden. Abbildung 5.8 zeigt den über die Zeit anwachsenden zurückgelegten Weg eines Skifahrers, aufsummiert über alle erfassten Wegpunkte und in der Projektion auf die Ideallinie für die Abfahrtsstrecke. Die berechneten Daten für einen Läufer auf der Slalomstrecke sind in Abbildung 5.9 dargestellt.

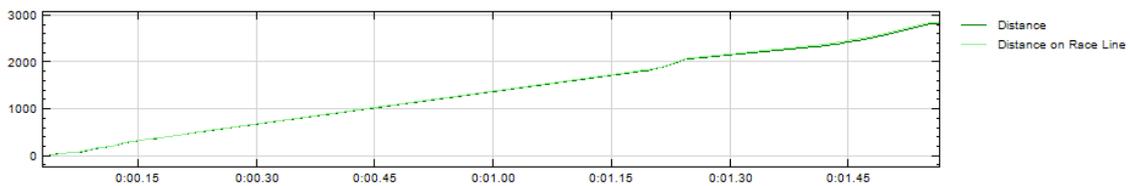
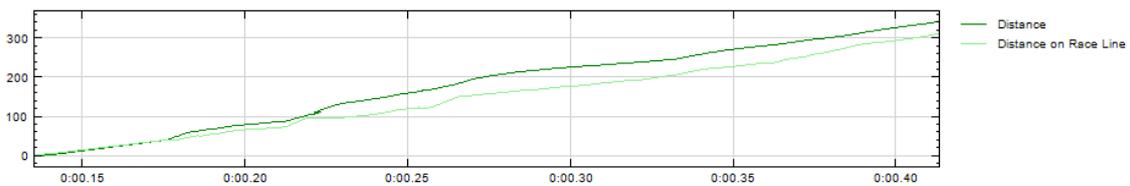
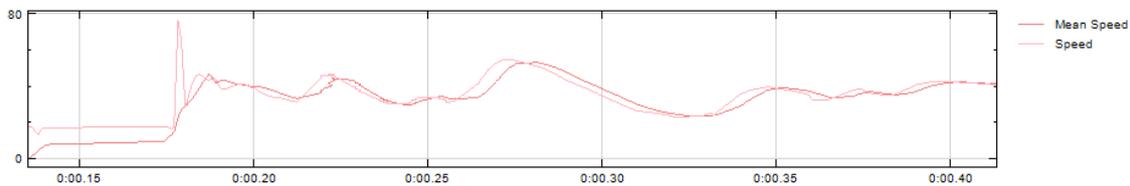


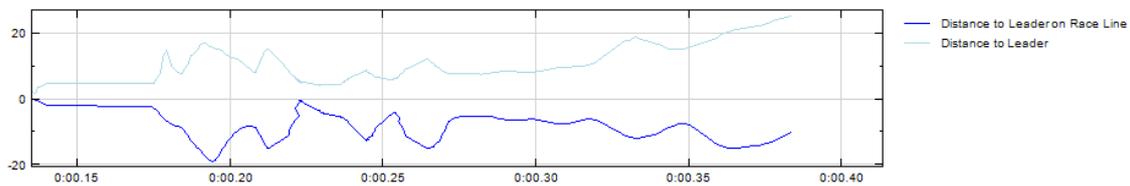
Abbildung 5.8: Distanzberechnung für einen Athleten auf der Abfahrtsstrecke



(a)



(b)



(c)

Abbildung 5.9: Distanz (a), Geschwindigkeit (b) und Distanz zum Führenden (c) für einen Athleten im Slalom

Die Abbildungen 5.10 und 5.11 zeigen die in ellipsoidische Koordinaten rücktransformierten Positionsdaten unterschiedlicher Rennläufer.



Abbildung 5.10: Rücktransformierte Positionsdaten eines Läufers im Bereich „Mausefalle“

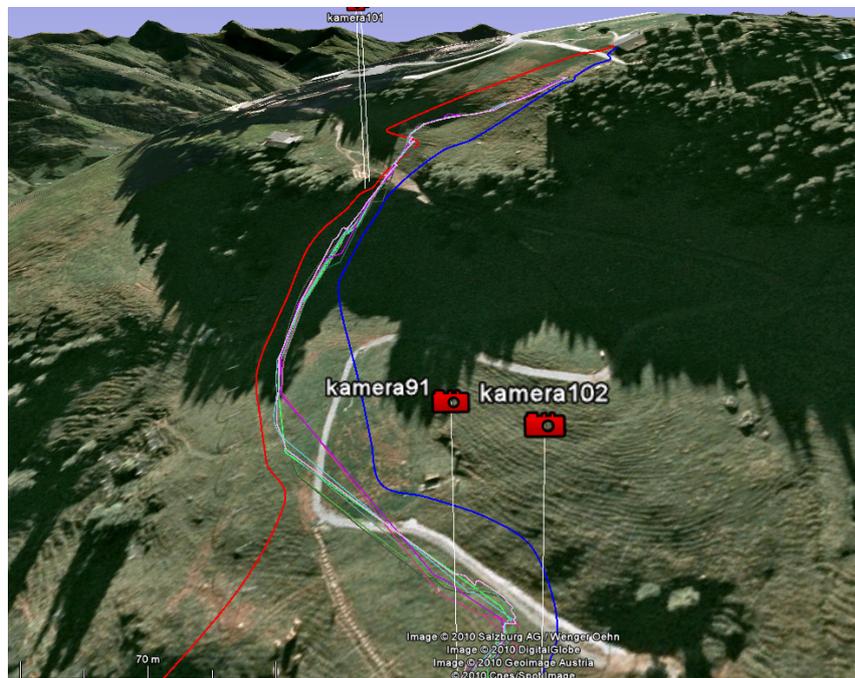


Abbildung 5.11: Vergleich der Positionsdaten unterschiedlicher Athleten zu Beginn der Abfahrt

# Kapitel 6

## Zusammenfassung

Es konnte gezeigt werden, dass ein bildgestütztes System zur dreidimensionalen Positionsbestimmung von Objekten unter Einschränkungen auch monokular aufgebaut werden kann und dass es trotz seiner Komplexität „weichen“ Echtzeitbedingungen genügt, selbst bei abstrakter Entwicklung ohne hardwarenaher Optimierung. Zu diesem Zweck wurden die mathematischen Grundlagen zur Umrechnung von Kameramodellen ohne Tiefeninformation im Bildbereich in solche mit einem Sichtvolumen geschaffen und die benötigten Schritte zur Erstellung derselben aus Punktkorrespondenzen gezeigt.

Es konnte der Aufbau transformierter Modelle von Geländeflächen und die virtuelle Darstellung derselben erarbeitet werden, was in Kombination mit bekannten Kameraparametern eine mit dem realen Bild zusammenfallende Ansicht ermöglicht. Diese ist elementare Grundlage für die Rückgewinnung der Tiefeninformation aus dem zweidimensionalen Kamerabild. Darüber hinaus konnten Ansätze zur automatisierten Objekterkennung und Fußpunktbestimmung für Sportanwendungen präsentiert werden, die Ausgangspunkte für weitere Arbeiten darstellen.

Schließlich wurde ein entsprechendes System real aufgebaut und seine grundsätzliche Funktionalität außer Zweifel gestellt, wenngleich dadurch auch einige Schwachpunkte aufgezeigt wurden.

### 6.1 Ausblick

Der Schwerpunkt dieser Arbeit lag auf der Tiefenrekonstruktion zur dreidimensionalen Positionsbestimmung und allen ihren Elementen. Weitere Forschungen dienen der Verfeinerung dieser Ansätze, insbesondere aber der Vertiefung und Erweiterung von Objekterkennung und -verarbeitung. Der jetzige Entwicklungsstand bildet eine solide Umgebung dafür und ermöglicht schnelle Integration und Evaluierung.

#### Kalibrierung und Modellbildung

Ziel ist es, auch unter widrigen Bedingungen zuverlässig Athleten zu detektieren und über mehrere Bilder hinweg zuzuordnen. Ist dieses erreicht, kann der Kalibrierungsprozess erweitert, automatisiert und mit der Erstellung des Geländeprofiles kombiniert werden. Ein erster Schritt dazu ist die direkte Übertragung der Positionsmesswerte, da auf diese Weise

die Qualität der Kalibrierung permanent überprüfbar wird. In weiterer Folge ermöglicht die automatische Verfolgung des Messgerätes in Bildkoordinaten die Aufnahme vieler Punkt-korrespondenzen, die über einen RANSAC<sup>1</sup>-Algorithmus für eine zuverlässige Kalibrierung ausgewählt werden; eine Ergänzung des Kameramodells ist hierzu unumgänglich. Diese Daten ermöglichen sodann eine Optimierung des Modells der Geländefläche.

### **Objekterkennung und Objektverfolgung**

Die rudimentären Ansätze zur Erkennung von Athleten und zu deren Verfolgung erfordern eine Überarbeitung. Der Erkennungsprozess muss stabiler werden, insbesondere in Bezug auf wechselnde Lichtbedingungen und Schatten. Bei einem Skirennen wirken sich sowohl der Schattenwurf des Skifahrers als auch der von eventuellen Schneewolken negativ auf die Erkennungsrate aus. Die Zuordnung von Objekten muss über mehrere Bilder hinweg über Wahrscheinlichkeitsbeziehungen erfolgen, unterstützt durch die Vorhersage und Korrektur eines Filters, der ein wesentlich genaueres Dynamikmodell des Athleten beinhaltet. Dieses Modell ermöglicht in weiterer Folge auch die Bestimmung zusätzlicher Parameter, wie beispielsweise die durch das Aufkanten eines Skis entstehenden Gegenkräfte.

### **Verarbeitungsleistung und Bildrate**

Die bisherige Architektur ermöglicht in der gewählten Auflösung eine maximale Framerate von 15 Bildern pro Sekunde, was bei geringer Beobachtungsdistanz und hohen Geschwindigkeiten nicht für alle möglichen Anwendungsfälle ausreichend ist. Eine entsprechende Steigerung, möglicherweise auch durch zusätzliche Hardwarekomponenten, ist aus diesem Grund in Betracht zu ziehen.

## **6.2 Schlusswort**

Das vorgestellte Verfahren ist bereits jetzt, zu Beginn der Forschungsarbeit, auf sehr große Resonanz bei Verantwortlichen, nicht nur des Skiworldcups, sondern auch aus den Bereichen der Formel 1 oder des RedBull Airrace, gestoßen. Unabhängig davon, ob die genannten Punkte zukünftig vollständig umgesetzt werden können, wurde mit dieser Arbeit ein völlig neuer Weg in der Analyse und Beurteilung von sportlichen Leistungen beschritten, der ein vielfältiges Anwendungsspektrum eröffnet und neue mediale Zugänge schafft.

---

<sup>1</sup> Random Sample Consensus: Eine Ergänzung zu Ausgleichsverfahren in der Modellschätzung, um grobe Messfehler und Ausreißer zu entfernen.

# Anhang A

## Ergänzungen

### A.1 Matlab

#### Lochkameramodell

Berechnung der Jacobimatrix für den Levenberg-Marquardt Algorithmus für das Lochkameramodell erweitert nach Mittrapiyanuruk 2006.

```
1 clear
2 clc
3
4 % declaration of variables
5 syms u0 v0 au av sk
6 syms tx ty tz wx wy wz
7 syms X Y Z um vm
8 syms f w
9
10 % the intrinsic parameter matrix
11 K=[au sk u0;
12    0 av v0;
13    0 0 f];
14
15 % Expression for the rotation matrix based on the Rodrigues formula
16 theta=sqrt(wx^2+wy^2+wz^2);
17 omega=[0 -wz wy;
18        wx 0 -wx;
19        -wy wx 0];
20 R = eye(3) + (sin(theta)/theta)*omega + ((1-cos(theta))/theta^2)*(omega*omega);
21
22 % Expression for the translation vector
23 t=[tx;ty;tz];
24
25 % perspective projection of the model point (X,Y, Z)
26 uvs= K * [R t] * [X; Y; Z; 1];
27
28 u=uvs(1)/uvs(3);
29 v=uvs(2)/uvs(3);
30
31 % calculate the geometric distance in x and y direction
32 % um,vm =the x and y positions of an extracted corner
33 % u,v = the x and y positions of the projection of the corresponding model point
34
35 dx=(um-u);
36 dy=(vm-v);
37
38 d = (dx^2+dy^2) + w * sk^2 + w * (au-av)^2;
39
40 % Evaluate the symbolic expression of the Jacobian w.r.t. the estimated
41 % parameters
42
43 Jx=jacobian(dx,[au,av,u0,v0,sk,wx wy wz tx ty tz]);
44 Jy=jacobian(dy,[au,av,u0,v0,sk,wx wy wz tx ty tz]);
45
46 Jd=jacobian(d,[au,av,u0,v0,sk,wx wy wz tx ty tz]);
```

## DirectX

Berechnung der Jacobimatrix für den Levenberg-Marquardt Algorithmus für DirectX

```

1 clear
2 clc
3
4 % declaration of variables
5 syms left right top bottom near far
6 syms lax lay laz upx upy upz posx posy posz
7 syms X Y Z um vm
8 syms Width Height TopX TopY ZMin ZMax
9
10 cameraPosition = [posx; posy; posz];
11 cameraTarget = [lax; lay; laz];
12 cameraUpVector = [upx; upy; upz];
13
14 view = directXViewRH(cameraPosition, cameraTarget, cameraUpVector);
15 perspective = directXPerspectiveRH(left, right, top, bottom, near, far);
16
17 T = perspective' * view';
18
19 % perspective projection of the model point (X,Y,Z)
20 uvs = T * [X; Y; Z; 1];
21 uvs = uvs / uvs(4);
22 uvss = directXViewport(Width, Height, TopX, TopY, ZMin, ZMax, uvs);
23
24 u=uvss(1);
25 v=uvss(2);
26
27 dx=(um-u);
28 dy=(vm-v);
29
30 d=dx^2+dy^2;
31
32 Jx=jacobian(dx,[left right top bottom near lax lay laz upx upy upz posx posy posz]);
33 Jy=jacobian(dy,[left right top bottom near lax lay laz upx upy upz posx posy posz]);
34 Jd=jacobian(d,[left right top bottom near lax lay laz upx upy upz posx posy posz]);

```

```

1 function perspective = directXPerspectiveRH(right, left, top, bottom, znearPlane, zfarPlane)
2
3 % 2*znearPlane/(right-left) 0 0 0
4 % 0 2*znearPlane*(top-bottom) 0 0
5 % (left+right)/(right-left) (top+bottom)/(top-bottom) zfarPlane/(znearPlane-zfarPlane) -1
6 % 0 0 znearPlane*zfarPlane/(znearPlane-zfarPlane) 0
7
8 perspective = [2*znearPlane/(right-left) 0 0 0;
9 0 2*znearPlane*(top-bottom) 0 0;
10 (left+right)/(right-left) (top+bottom)/(top-bottom) zfarPlane/(znearPlane-zfarPlane) -1;
11 0 0 znearPlane*zfarPlane/(znearPlane-zfarPlane) 0];

```

```

1 function res = directXViewport(Width, Height, TopLeftX, TopLeftY, MinDepth, MaxDepth, P)
2
3 X = (P(1) + 1) * Width * 0.5 + TopLeftX;
4 Y = (1 - P(2)) * Height * 0.5 + TopLeftY;
5 Z = MinDepth + P(3) * (MaxDepth - MinDepth);
6
7 res = [X; Y; Z];

```

```

1 function view = directXViewRH(cameraPosition, cameraTarget, cameraUpVector);
2 % View Matrix, from DirectX Documentation
3 % zaxis = normal(cameraPosition - cameraTarget)
4 % xaxis = normal(cross(cameraUpVector, zaxis))
5 % yaxis = cross(zaxis, xaxis)
6 %
7 % xaxis.x yaxis.x zaxis.x 0
8 % xaxis.y yaxis.y zaxis.y 0
9 % xaxis.z yaxis.z zaxis.z 0
10 % -dot(xaxis, cameraPosition) -dot(yaxis, cameraPosition) -dot(zaxis, cameraPosition) 1
11
12
13 zaxis = (cameraPosition - cameraTarget);
14 xaxis = (cross(cameraUpVector, zaxis));
15 yaxis = cross(zaxis, xaxis);
16
17 view = [ xaxis(1) yaxis(1) zaxis(1) 0;
18 xaxis(2) yaxis(2) zaxis(2) 0;
19 xaxis(3) yaxis(3) zaxis(3) 0;
20 -dot(xaxis, cameraPosition) -dot(yaxis, cameraPosition) -dot(zaxis, cameraPosition) 1];

```

## A.2 Datenblätter

### AXIS Q1755

Technische Daten – AXIS Q1755 Netzwerk-Kamera	
<b>Kamera</b>	
<b>Modelle</b>	AXIS Q1755 50Hz AXIS Q1755 60Hz
<b>Bildsensor</b>	1/3" CMOS, progressive Abtastung, 2 Megapixel
<b>Objektiv</b>	f 5,1 – 51 mm, F1,8 – 2,1 Automatischer Tag-/Nacht-IR-Filter, Autofokus Brennweitenbegrenzung 10 mm (Weitwinkel) oder 800 mm (Tele) Sichtwinkel, horizontal: 5,4° – 50° M37x0,75-Anschlussgewinde für den Einsatz einer Vorsatzlinse
<b>Minimale Lichtstärke</b>	Farbe: 2 Lux bei 30 IRE, F1,8 Schwarzweiß: 0,2 Lux bei 30 IRE, F1,8
<b>Verschlusszeit</b>	1/10000 s bis 1/2 s
<b>Zoom</b>	10-fach optisch und 12-fach digital, insgesamt 120-fach
<b>Video</b>	
<b>Video-komprimierung</b>	H.264 (MPEG-4 Part 10/AVC) Motion JPEG
<b>Auflösungen</b>	HDTV 1080i 1920 x 1080 HDTV 720p 1280 x 720 Standard-NTSC/PAL für die Installation
<b>Bildrate H.264</b>	30/25 Bilder/s bei allen Auflösungen
<b>Bildrate Motion JPEG</b>	30/25 Bilder/s bei allen Auflösungen
<b>Videostrom</b>	Multistream H.264 und Motion JPEG, gleichzeitige, individuell konfigurierbare Streams mit einer max. Auflösung von 25/30 Bilder/s Kontrollierbare Bildrate und Bandbreite VBR/CBR H.264
<b>Bildeinstellungen</b>	Komprimierung, Helligkeit, Schärfte, Weißabgleich, Belichtungssteuerung, Gegenlichtausgleich, Bildrotation, Spiegelung von Bildern Text- und Bild-Overlay Privatzenonenmaske
<b>Audio</b>	
<b>Audio-Streaming</b>	Zweiwege, Halbduplex
<b>Audio-komprimierung</b>	AAC LC 8/16 kHz G.711 PCM 8 kHz G.726 ADPCM 8 kHz Konfigurierbare Bitrate
<b>Audio-Eingang/-Ausgang</b>	Integriertes Mikrofon, externes Mikrofon oder Audio-Eingang, Audio-Ausgang
<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, QoS 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, Telco-D
<b>Systemintegration</b>	
<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>
<b>Intelligentes Video</b>	Videobewegungserkennung, aktiver Manipulationsalarm, Audioerkennung, Gatekeeper
<b>Alarmauslösung</b>	Intelligentes Video, Temperatur und externe Signale
<b>Alarmereignisse</b>	Hochladen von Dateien über FTP, HTTP und E-Mail, HTTP und TCP Benachrichtigung über E-Mail, HTTP und TCP Aktivierung externer Ausgänge
<b>Video-puffer</b>	96 MB Vor- und Nachalarm
<b>Allgemein</b>	
<b>Prozessor und Speicher</b>	ARTPEC-3, 256 MB RAM, 128 MB Flash
<b>Stromversorgung</b>	8 bis 20 V Gleichstrom, max. 11,2 W 20 bis 24 V Wechselstrom, max. 17,4 VA Power over Ethernet (IEEE 802.3af), Klasse 3 Netzteil nicht im Lieferumfang enthalten
<b>Anschlüsse</b>	RJ-45 für 10BASE-T-/100BASE-TX-PoE, Anschlussblock für die Stromversorgung, E/A-Anschlussblock für zwei konfigurierbare Eingänge/Ausgänge 3,5-mm-Buchse für Mikro/Audio-Eingang, 3,5-mm-Buchse für Audio-Ausgang RS-422/RS-485 Video-Ausgang: 3x RCA Y/Pb/Pr (HD), 1x RCA Composite (SD)
<b>Lokaler Speicher</b>	SD/SDHC-Speicherkartensteckplatz (Speicherkarte nicht im Lieferumfang enthalten)
<b>Betriebsbedingungen</b>	0 – 45 °C Relative Luftfeuchtigkeit 20 – 80 % (nicht kondensierend)
<b>Zulassungen</b>	EN 55022 Klasse B, EN 61000-3-2, EN 61000-3-3, 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
<b>Gewicht</b>	985 g
<b>Im Lieferumfang enthaltenes Zubehör</b>	Anschlusszubehör, Installationsanleitung, CD mit Installationsprogrammen, Aufzeichnungssoftware und Benutzerhandbuch, Windows Decoder-Lizenz für 1 Benutzer
Weitere Informationen finden Sie unter <a href="http://www.axis.com">www.axis.com</a>	

Abbildung A.1: Datenblatt der AXIS Q1755 Kamera

# Literaturverzeichnis

## Kapitel 1: Einleitung

- Lepetit, Vincent. und Pascal. Fua (2005b). *Monocular model-based 3D tracking of rigid objects*. Boston, MA: NOW Publishers. ISBN: 1933019034 9781933019031. URL: [http://books.google.at/books?hl=de&lr=&id=y-C1LBFtZDUC&oi=fnd&pg=PA1&dq=Gimbal+Lock&ots=fzMoWCIClJ&sig=xAjYSqphNDYZbg3VQTLvk4d\\_ym4#v=onepagef=false](http://books.google.at/books?hl=de&lr=&id=y-C1LBFtZDUC&oi=fnd&pg=PA1&dq=Gimbal+Lock&ots=fzMoWCIClJ&sig=xAjYSqphNDYZbg3VQTLvk4d_ym4#v=onepagef=false). Siehe Seite 10.
- Mauthner, Thomas und Horst Bischof (2007). „A Robust Multiple Object Tracking for Sport Applications“. In: *Performance Evaluation for Computer Vision, 31st AAPR/OAGM Workshop 2007*. Austrian Association for Pattern Recognition. Österreichische Computer Gesellschaft, Seiten 81–89. Siehe Seiten 11, 50.
- Nachbauer, W. u. a. (1996). „A Video Technique for Obtaining 3-D Coordinates on Alpine Skiing“. In: *Journal of Applied Biomechanics* 12, Seiten 104–115. Siehe Seite 11.

## Kapitel 2: Kamerakalibrierung

- Abdel-Aziz, Y. I. und H.M. Karara (1971). „Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry“. In: *Proc. the Symposium on Close-Range Photogrammetry, American Society of Photogrammetry*. URL: <http://ci.nii.ac.jp/naid/10020508572/en/>. Siehe Seite 23.
- Bakstein, Hynek (1999). „A Complete DLT-based Camera Calibration with a Virtual 3D Calibration Object“. Magisterarbeit. Faculty of Mathematics und Physics, Charles University, Prague. Siehe Seite 22.
- Grassia, F. Sebastin (1998). „Practical parameterization of rotations using the exponential map“. In: *J. Graph. Tools* 3.3, Seiten 29–48. ISSN: 1086-7651. Siehe Seite 16.
- Hartley, Richard und Andrew Zisserman (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press. ISBN: 0521540518. URL: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0521540518>. Siehe Seiten 13–16, 24, 26, 27, 29, 80.
- Izumi, M. und K. Fukunaga (1996). „Back projection voting considering the error of camera orientation“. In: *Emerging Technologies and Factory Automation, 1996. EFTA '96. Proceedings., 1996 IEEE Conference on*. Band 2, 712–716 vol.2. DOI: 10.1109/ETFA.1996.573992.
- Kraus, Karl (1998). *Photogrammetrie*. Berlin: Gruyter. Siehe Seite 15.

- Labuz, J., M. Thaker und B. Venkateswaran (1991). „Decomposition of the camera calibration matrix“. In: *System Theory, 1991. Proceedings., Twenty-Third Southeastern Symposium on*, Seiten 89–93. DOI: 10.1109/SSST.1991.138520.
- Liu, GuiHua u. a. (2009). „A Novel Camera Calibration Method of Variable Focal Length Based on Single-View“. In: *Electronic Commerce and Security, 2009. ISECS '09. Second International Symposium on*. Band 2, Seiten 125–128. DOI: 10.1109/ISECS.2009.235.
- Marquardt, Donald W. (1963). „An Algorithm for Least-Squares Estimation of Nonlinear Parameters“. In: *Journal of the Society for Industrial and Applied Mathematics* 11.2, Seiten 431–441. ISSN: 03684245. URL: <http://www.jstor.org/stable/2098941>. Siehe Seite 27.
- Mittrapiyanuruk, Pradit (2006). *A Memo on How to Use the Levenberg-Marquardt Algorithm for Refining Camera Calibration Parameters*. Technischer Bericht. URL: <http://cobweb.ecn.purdue.edu/~kak/courses-i-teach/ECE661/> (besucht am 2009.11.15). Siehe Seiten 28, 72.
- Pourreza, R. u. a. (2008). „Robust camera calibration of soccer video using genetic algorithm“. In: *Intelligent Computer Communication and Processing, 2008. ICCP 2008. 4th International Conference on*, Seiten 123–127. DOI: 10.1109/ICCP.2008.4648363.
- Press, William H. u. a. (1992). *Numerical Recipes in FORTRAN 77: The Art of Scientific Computing (v. 1)*. 2. Auflage. Cambridge University Press. ISBN: 052143064X. URL: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/052143064X>. Siehe Seite 25.
- Ranganathan, Ananth (2004). *The Levenberg-Marquardt Algorithm*. Technischer Bericht. Honda Research Institute in Cambridge, MA. Siehe Seite 27.
- Rodrigues, Benjamin Olinde (1840). „Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire“. In: *Journal de Mathématiques Pures et Appliquées* 5, Seiten 380–440. URL: <http://gallica.bnf.fr>. Siehe Seite 17.
- Ruiz, A., P.E. Lopez-de Teruel und G. Garcia-Mateos (2002). „A note on principal point estimability“. In: *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. Band 2, 304–307 vol.2. DOI: 10.1109/ICPR.2002.1048300. Siehe Seite 34.
- Salvi, Joaquim, Xavier Armangué und Joan Batlle (2002). „A comparative review of camera calibrating methods with accuracy evaluation“. In: *Pattern Recognition* 35.7, Seiten 1617–1635. ISSN: 0031-3203. DOI: DOI:10.1016/S0031-3203(01)00126-1. URL: <http://www.sciencedirect.com/science/article/B6V14-45CTFWM-H/2/8e2cf6ae4bcdb469dfb07f02069a1014>.
- Stuelpnagel, John (1964). „On the Parametrization of the Three-Dimensional Rotation Group“. In: *SIAM Review* 6.4, Seiten 422–430. ISSN: 00361445. URL: <http://www.jstor.org/stable/2027966>. Siehe Seite 16.
- Sun, Wei und J.R. Cooperstock (2005). „Requirements for Camera Calibration: Must Accuracy Come with a High Price?“. In: *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*. Band 1, Seiten 356–361. DOI: 10.1109/ACVMOT.2005.102.

- Triggs, B. (1999). „Camera pose and calibration from 4 or 5 known 3D points“. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. Band 1, 278–284 vol.1. DOI: 10.1109/ICCV.1999.791231. Siehe Seiten 34, 80.
- Tsai, R. (1987). „A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses“. In: *Robotics and Automation, IEEE Journal of* 3.4, Seiten 323–344. ISSN: 0882-4967. DOI: 10.1109/JRA.1987.1087109. Siehe Seite 22.
- Ying, Xianghua und Hongbin Zha (2007). „Camera calibration from a circle and a coplanar point at infinity with applications to sports scenes analyses“. In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, Seiten 220–225. DOI: 10.1109/IROS.2007.4399329.
- Zhang, Z. (2000). „A flexible new technique for camera calibration“. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.11, Seiten 1330–1334. ISSN: 0162-8828. DOI: 10.1109/34.888718.

### Kapitel 3: Tiefenrekonstruktion

- Chen, F., G. M. Brown und M. Song (2000). „Overview of 3-D shape measurement using optical methods“. In: *Optical Engineering* 39 (Januar 2000), Seiten 10–22. DOI: 10.1117/1.602438. Siehe Seite 35.
- Clark, R. L. und R. Lee (1997). „Development of topographic maps with RTK-DGPS“. In: Seite 15. DOI: 10.1109/AIM.1997.652873.
- Egger, Dieter. *Formeln und Konstanten für die Berechnung der Schweizerischen schiefachsigen Zylinderprojektion und der Transformation zwischen Koordinatensystemen*. URL: <http://www.swisstopo.admin.ch/internet/swisstopo/de/home/topics/survey/sys/refsys.parsysrelated1.23611.downloadList.12097.DownloadFile.tmp/refsysd.pdf> (besucht am 16.02.2010). Siehe Seite 42.
- Han, Mei und Takeo Kanade (1998). „Homography-Based 3D Scene Analysis of Video Sequences“. In: *In Proceedings of the 1998 DARPA Image Understanding Workshop*, Seiten 154–160. Siehe Seite 37.
- Morales, Y. und T. Tsubouchi (2007a). „DGPS, RTK-GPS and StarFire DGPS Performance Under Tree Shading Environments“. In: Seiten 519–524. DOI: 10.1109/ICITECHNOLOGY.2007.4290370.
- (2007b). „GPS moving performance on open sky and forested paths“. In: Seiten 3180–3185. DOI: 10.1109/IROS.2007.4399307. Siehe Seiten 40, 41.
- Pages, J. u. a. (2003). „Overview of coded light projection techniques for automatic 3D profiling“. In: Band 1, 133–138 vol.1. Siehe Seite 35.
- Rafael, Comparative Study, Rafael J. Segura und Francisco R. Feito (2001). „Algorithms To Test Ray-Triangle Intersection“. In: *Journal of WSCG*, Seiten 200–1. Siehe Seite 45.
- Salman, Nader und Mariette Yvinec (2009). „Surface Reconstruction from Multi-View Stereo“. English. In: *Lecture notes in computer science* (23. September 2009). URL: <http://hal.inria.fr/inria-00422344/PDF/modeling3D-salman.pdf>. Siehe Seiten 37, 38, 80.

- Shevtsov, Maxim, Alexei Soupikov und Alexander Kapustin (2007). „Ray-Triangle Intersection Algorithm for Modern CPU Architectures“. In: *GraphiCon 2007*, Seiten 33–39. Siehe Seite 45.
- Shuxin, Chen, Wang Yongsheng und Chen Fei (2002). „A study of differential GPS positioning accuracy“. In: Seiten 361–364.
- Wübbena, Gerhard und Andreas Bagge (1995). „Präzises DGPS in Echtzeit für Vermessung und GIS-Anwendungen“. In: *DVW-Praxisseminar DGPS-Anwendungen*. Siehe Seite 39.

## Kapitel 4: Objekterkennung

- Chaumette, F. (2004). „Image moments: a general and useful set of features for visual servoing“. In: *Robotics, IEEE Transactions on* 20.4, Seiten 713–723. ISSN: 1552-3098. DOI: 10.1109/TR0.2004.829463. Siehe Seite 54.
- Di Stefano, L. und A. Bulgarelli (1999). „A simple and efficient connected components labeling algorithm“. In: Seiten 322–327. DOI: 10.1109/ICIAP.1999.797615. Siehe Seite 54.
- Lepetit, V. und P. Fua (2005a). „Monocular Model-Based 3D Tracking of Rigid Objects: A Survey“. In: *Foundations and Trends® in Computer Graphics and Vision* 1.1. URL: <http://www.nowpublishers.com/product.aspx?product=CGV&#38;doi=0600000001>. Siehe Seite 50.
- Mauthner, Thomas und Horst Bischof (2007). „A Robust Multiple Object Tracking for Sport Applications“. In: *Performance Evaluation for Computer Vision, 31st AAPR/OAGM Workshop 2007*. Austrian Association for Pattern Recognition. Österreichische Computer Gesellschaft, Seiten 81–89. Siehe Seiten 11, 50.
- Piccardi, M. (2004). „Background subtraction techniques: a review“. In: *Proc. of IEEE SMC 2004 International Conference on Systems, Man and Cybernetics*. Band 4. The Hague, The Netherlands: IEEE, Seiten 3099–3104. Siehe Seite 52.
- Rodriguez, J. und D. Ayala (2001). *Erosion and Dilation on 2D and 3D Digital Images: A new size-independent approach*. VMV. Siehe Seite 53.
- Steinmüller, Johannes (2008). „Morphologische Operationen“. In: *Bildanalyse*, Seiten 65–89. URL: [http://dx.doi.org/10.1007/978-3-540-79743-2\\_3](http://dx.doi.org/10.1007/978-3-540-79743-2_3). Siehe Seite 53.
- Teh, C.-H. und R.T. Chin (1988). „On image analysis by the methods of moments“. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 10.4, Seiten 496–513. ISSN: 0162-8828. DOI: 10.1109/34.3913. Siehe Seite 54.

## Kapitel 5: Integration

- Lipinski, Klaus. *ITWissen - Das große Online-Lexikon für Informationstechnologie*. URL: <http://www.itwissen.info/>. Siehe Seite 63.
- Lutz, M.H. und P.A. Laplante (2003). „C# and the .NET framework: ready for real time?“. In: *Software, IEEE* 20.1, Seiten 74–80. ISSN: 0740-7459. DOI: 10.1109/MS.2003.1159034. Siehe Seite 62.

## Kapitel 7: Ergänzungen

Mittrapiyanuruk, Pradit (2006). *A Memo on How to Use the Levenberg-Marquardt Algorithm for Refining Camera Calibration Parameters*. Technischer Bericht. URL: <http://cobweb.ecn.purdue.edu/~kak/courses-i-teach/ECE661/> (besucht am 2009.11.15). Siehe Seiten 28, 72.

# Abbildungsverzeichnis

2.1	Äußere Orientierung einer Kamera (Hartley und Zisserman 2004, Fig. 5.3) .	14
2.2	Innere Orientierung der Lochkamera (Hartley und Zisserman 2004, Fig. 6.1)	15
2.3	Abbildungsfrustum . . . . .	18
2.4	Zusammenhang zwischen computergraphischer und photogrammetrischer Projektion . . . . .	21
2.5	Kalibrierbeispiel „Mausefalle“ . . . . .	32
2.6	Kalibrierbeispiel „Lärchenschuss“ . . . . .	32
2.7	Kalibrierbeispiel „Oberhausberg“ . . . . .	33
2.8	Vergleich unterschiedlicher Kalibriermodelle (Triggs 1999) . . . . .	34
3.1	Übereinstimmung von virtuellem Modell und dem tatsächlichen Objekt . .	36
3.2	Modellrekonstruktion aus Luftbildaufnahmen (Salman und Yvinec 2009) . .	38
3.3	DGPS Systeme . . . . .	40
3.4	Koordinatentransformation zur Umrechnung eines ellipsoidischen Datums in kartesische Koordinaten des Referenzsystems. . . . .	42
3.5	Vermessene Punkte und Interpolation der links- und rechtsseitigen Flächen- begrenzung . . . . .	45
3.6	Schnittpunktgewinnung . . . . .	46
3.7	Prozessablauf zur Gewinnung der Korrespondenzpunkte für die Rückpro- jektion . . . . .	47
3.8	Streckenabschnitt „Lärchenschuss“ . . . . .	48
3.9	Streckenabschnitt „Oberhausberg“ . . . . .	49
3.10	Streckenabschnitt „Zielschuss“ . . . . .	49
4.1	Bildverarbeitung . . . . .	51
4.2	Bildaufteilung . . . . .	52
4.3	Ideallinie . . . . .	56
4.4	Objekterkennung auf dem Streckenabschnitt „Lärchenschuss“, Bildsequenz .	56
4.5	Objekterkennung auf dem Streckenabschnitt „Lärchenschuss“, Bild (b) . . .	57
4.6	Objekterkennung auf dem Streckenabschnitt „Lärchenschuss“, Bild (e) . . .	57
5.1	Prozessablauf und Datenabhängigkeiten des Gesamtsystems . . . . .	60
5.2	Logische Verarbeitungseinheiten . . . . .	61
5.3	Dezentrale logische Verarbeitungseinheiten . . . . .	62
5.4	Hardwarestruktur . . . . .	63
5.5	Intel Xeon E5520 Chipset . . . . .	64

5.6	Systemaufbau Kitzbühel, Kommunikationswege . . . . .	65
5.7	Vermessungs- und Kalibrierergebnisse für die Abfahrtsstrecke . . . . .	66
5.8	Distanzberechnung für einen Athleten auf der Abfahrtsstrecke . . . . .	68
5.9	Distanz (a), Geschwindigkeit (b) und Distanz zum Führenden (c) für einen Athleten im Slalom . . . . .	68
5.10	Rücktransformierte Positionsdaten eines Läufers im Bereich „Mausefalle“ . . . . .	69
5.11	Vergleich der Positionsdaten unterschiedlicher Athleten zu Beginn der Abfahrt . . . . .	69
A.1	Datenblatt der AXIS Q1755 Kamera . . . . .	74

# Index

- Algorithmus
  - Gauß-Newton-, 27
  - Levenberg-Marquardt-, 27
- APOS, 39
- Bildmomente, 54
- Bildverarbeitung, 51
- Brennpunkt, 20
- Computergraphik, 17
- Connected component labeling, 54
- Dekomposition
  - RQ, 25
  - Transformationsmatrix, 24
- DGPS, 39
- DLT, 23
- E5520, 64
- Echtzeit, 58
- EGNOS, 39
- Entwicklungsumgebung, 62
- Exponentialabbildung, 16
- Farbbereich, 53
- Filter
  - Kalman-, 55
- Fußpunkt, 55
- GNSS, 39
  - DGPS, 39
  - RTK, 39
- GPS, 39
- Gradientenverfahren, 27
- H.264, 63
- Hintergrundschätzung, 52
- Histogramm, 52
- Homographie, 37
- Kamera
  - kalibrierung, 13
  - modelle, 13
  - orientierung, 14
  - rotation, 16
- Keying, 53
- Koordinaten
  - Baryzentrische-, 45
  - Ellipsoidische-, 42
  - Kartesische-, 42
- Kovarianz, 54
- Kreuzprodukt, 17
- Laufzeit, 52
- Luftbild, 37
- Matrix
  - Betrachtungs-, 17
  - Hesse-, 27
  - Jacobi-, 27
  - Kovarianz-, 54
  - Projektions-, 19
  - Transformations-, 24
- Modell, 37
- Modellbildung, 37
- Morphologie, 53
- MPEG, 63
- Nichtlineare Optimierung, 26
- Ntrip, 39
- Objektextraktion, 54
- Objektklassifikation, 54
- Objektverarbeitung, 54
- Objektverfolgung, 55
- Operationen
  - morphologische, 53
- Orientierung, 14
  - äußere, 14
  - innere, 14

- Parallelisierung, 52
- Photogrammetrie, 15
- Punktkorrespondenzen, 22
  
- Rodriguesformel, 17
- Rotation
  - Givens-, 25
  - Kamera-, 16
- RTCM, 39
- RTK-GPS, 40
  
- Schnittpunkt, 45, 55
- Schnittpunktberechnung, 45
- Stereopsis, 35
  
- Tiefenrekonstruktion, 35
  
- Varianz, 54
- Verarbeitung, 63
  
- WAAS, 39
- Winkel
  - Eulersche, 16
  
- Zielhardware, 64