Stereobildbasierte Kollisionsvermeidung für einen unbemannten Kleinhubschrauber

Diplomarbeit

zur Erlangung des akademischen Grades Diplominformatiker
eingereicht am Institut für Informatik
der Humboldt-Universität zu Berlin

von

Franz Andert

April 2006

Gutachter: Prof. Dr. Ralf Reulke (HU Berlin, Institut für Informatik)
Prof. Dr. Herbert Jahn (DLR, Institut für Robotik und Mechatronik)

Betreuer: Dr.-Ing. Gordon Strickert (DLR, Institut für Flugsystemtechnik)

Registrier-Nr.: .....
eingereicht am: .....

Platzhalter für das Original der Aufgabenstellung.

Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Ich bin damit einverstanden, dass die vorliegende Arbeit in der Bibliothek des Instituts für Informatik der Humboldt-Universität zu Berlin ausgestellt werden darf.

Berlin, den 10. April 2006 .....

Übersicht

Der Rahmen dieser Arbeit ist ein Projekt des Instituts für Flugsystemtechnik am Deutschen Zentrum für Luft- und Raumfahrt (DLR) in Braunschweig. Forschungsgegenstand ist ein unbemannter Kleinhubschrauber, an dem Techniken des autonomen Flugs erprobt werden.

Die Arbeit beschreibt die Entwicklung einer Strategie zur Kollisionsvermeidung unter Verwendung einer Stereokamera. Sie ermöglicht die dreidimensionale Rekonstruktion der Umgebung, damit die Messung von Entfernungen und letztendlich die Erkennung von Hindernissen und hindernisfreien Bereichen. Das implementierte Verfahren wird in einer Simulationsumgebung getestet, die die originale Hardware des Hubschraubers verwendet und dessen Flugeigenschaften nachbildet. So kann für einen realen Flug abgeschätzt werden, wie sich der Helikopter bei Hindernissen verhält und inwieweit Kollisionen vermieden werden können.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

1.1 Unbemannte fliegende Systeme

1.1.1 Der Helikopter ARTIS

ARTIS[1] ist ein Flugversuchsträger auf Basis eines unbemannten Kleinhubschraubers zu Forschungszwecken. Dieses Projekt wurde 2003 am Institut für Flugsystemtechnik des DLR gestartet und seitdem ständig weiterentwickelt. Angestrebte Anwendungsziele sind unter anderem Rettungsmissionen *(Search and Rescue)* und die Unterstützung von bemannten Helikoptern zur Aufklärung *(Manned/Unmanned Teaming)*.

Neben der Verbesserung der Flugeigenschaften und der damit verbundenen vereinfachten Handhabung ist das autonome Fliegen ein Schwerpunkt dieses Projekts. Bildgestützte Anwendungen sind beispielsweise das selbstständige Landen auch auf sich bewegendem Untergrund und die Verfolgung von Objekten wie Fahrzeugen. Die Verfolgung eines Musterfahrzeugs wurde in Flugversuchen bereits erfolgreich durchgeführt. Diese Arbeit ist Teil eines neueren Vorhabens, das sich die automatische Hinderniserkennung und Kollisionsvermeidung zum Ziel gesetzt hat.

Abbildung 1.1 zeigt den Helikopter im Flug und einige technische Bauteile. Basis ist der kommerzielle Modellhubschrauber Genesis der Firma Benda. Der Rotor mit einem Durchmesser von 2,0 Meter wird von einem 2 PS starken Motor angetrieben. Die Maximalgeschwindigkeit beträgt etwa 120 km/h. Der Hubschrauber hat eine Nutzlast von etwa 6 kg, die hauptsächlich für die Sensorik und Rechentechnik verwendet wird. Die Größe des Treibstofftanks[2] und die Kapazität der Akkus für die Elektronik ermöglichen Flüge von über 20 Minuten [TDB04]. Ein größerer Helikopter namens *maxiARTIS* mit höherer Geschwindigkeit, Reichweite und Nutzlast befindet sich derzeit in der Erprobungsphase.

[1] Autonomous Rotorcraft Testbed for Intelligent Systems
[2] Als Treibstoff wird ein Methanol/Öl-Gemisch verwendet.

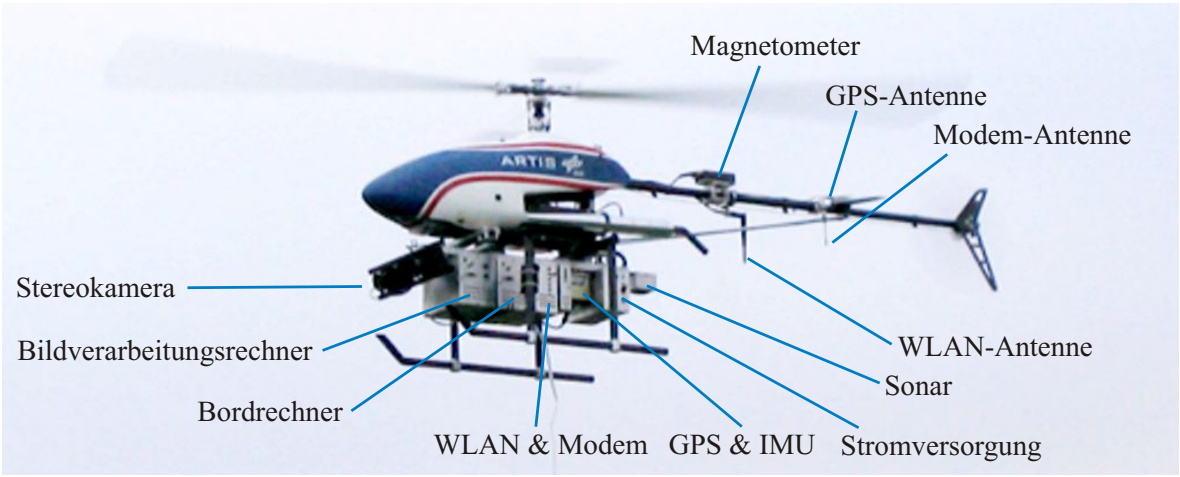


Abbildung 1.1: Der ARTIS-Helikopter des Instituts für Flugsystemtechnik.

Die in dieser Arbeit verwendete Sensorik ist hauptsächlich die Stereokamera an der Front des Helikopters. Sie ist an einen Computer angeschlossen, der ausschließlich für Bildverarbeitungszwecke an Bord ist. Verbindungen zur Bodenstation werden mit Wireless LAN oder über längere Distanzen mit einem Funkmodem hergestellt. Ein lokales Kabelnetzwerk sichert die ständige Kommunikation zwischen Bildverarbeitung und dem Bordrechner für die Flugbefehle. Ziel ist es, dass der Bildverarbeitungsrechner mit Hilfe von Kameras Informationen über Hindernisse oder spezielle Punkte zum Landen oder Verfolgen an den Bordrechner übermittelt, welcher dann selbstständig die Wegplanung übernimmt und entsprechende Flugmanöver ausführt.

1.1.2 Verwandte Projekte

Die Forschung an *VTOL*[3] *UAVs*[4] konzentriert sich hauptsächlich in Einrichtungen der USA. Herausragend ist dabei vor allem das Projekt des *Carnegie Mellon Robotics Institute* in Pittsburgh. Hier wird seit 1991 an einem Helikopter gearbeitet, der vor allem mit Hilfe von Kameras autonom fliegen und landen soll [Ami96]. Beim *Berkeley Aerobot (BEAR)* ist ein Schwerpunkt das autonome Landen auf bekannten Objekten [SSS01]. Ein weiterer autonomer Helikopter ist unter anderem der *GTMax* des *Georgia Institute of Technology* [JS02].

Als europäische Forschungsaktivitäten seien die Projekte der *Autonomous Unmanned Aerial Vehicles Technology Group*, ehemals *WITAS*[5], der Universität Linköping, Schweden [Doh+00]

[3] Vertical Takeoff and Landing
[4] Unmanned Aerial Vehicle
[5] Wallenberg Laboratory for Information Technology and Autonomous Systems

sowie der Helikopter *RESSAC*[6] des *ONERA*[7] in Toulouse [Fab02] genannt. An der Technischen Universität Berlin gibt es ebenfalls ein Forschungsprojekt namens *MARVIN*[8] [Hom+03].

Für die Kollisionsvermeidung sind Versuchsergebnisse des *AVATAR*[9] der *University of Southern California* von besonderer Bedeutung. Es wird eine Kombination verschiedener Kamerasysteme verwendet. Zur Kollisionsvermeidung werden Experimente mit stereobasierter Entfernungsmessung zu frontalen Hindernissen und gleichzeitiger Messung des optischen Flusses an den Seiten durchgeführt, darüber hinaus ist eine omnidirektionale Kamera zur Rundumsicht in Verwendung. Vorteile seien dort vor allem die bessere Navigierbarkeit in Schluchten, d.h. bei Vorhandensein seitlicher Hindernisse und Ausweichmöglichkeiten [HSC+05], [Hra06]. Erfolge in Flugversuchen wurden bereits erreicht [Suk+05].

Als Basissystem hat sich in vielen Projekten der Helikopter *R50* und seit 1997 dessen Nachfolger *RMAX* von Yamaha bewährt. Der *RMAX Type II* hat eine Nutzlast von 31 kg und ermöglicht Flugzeiten von einer Stunde, was im Vergleich zu ARTIS wesentlich höher ist. Diese Hubschraubertypen sind jedoch wesentlich größer und benötigen im Gegensatz zu ARTIS eine Luftfahrtzulassung in Deutschland, was den Forschungseinsatz erschwert.

1.2 Räumliche Wahrnehmung

Die Erkennung von Hindernissen erfordert die dreidimensionale Erfassung des Raumes mit Hilfe von Sensoren. Allerdings liefern Kameras keine Positions- oder Entfernungsdaten bestimmter Objekte, sondern zunächst ein projiziertes Abbild der Umgebung. Erst die Weiterverarbeitung der Bilddaten ermöglicht eine Interpretation der visuellen Information und die Erfassung des Raumes unter bestimmten Voraussetzungen. Die Bilddatenverarbeitung kann prinzipiell an jene Art und Weise angelehnt sein, die auch beim Menschen zu einer räumlichen Erfassung der Umgebung führen. Neben rein physikalischen Gegebenheiten spielen dabei auch bestimmte Annahmen von Objekteigenschaften eine Rolle, die aufgrund von Erfahrungen gemacht werden. Eine Übersicht wird von Schulz zur Wiesch [Sch97] präsentiert.

In der Bildverarbeitung bezeichnet der Begriff *Shape From X* die Rekonstruktion dreidimensionaler Informationen aus Bilddaten [SS00]. Der Eindruck von Tiefe entsteht durch unterschiedliche Faktoren, von denen beispielsweise folgende technisch genutzt werden können:

Fokussierung: Je nach Fokussierung werden nur Objekte in einem bestimmten Entfernungsbereich scharf wahrgenommen, nähere oder weiter entfernte erscheinen zunehmend unscharf. Wird der Fokus so eingestellt, dass ein bestimmtes Objekt scharf abgebildet

[6] Recherche Et Sauvetage par Système Autonome Coopérant
[7] Office Nationale d'Études et de Recherches Aérospatiales
[8] Multi-purpose Aerial Robot Vehicle with Intelligent Navigation
[9] Autonomous Vehicle Aerial Tracking And Reconnaissance

wird, ergibt sich aus dieser Einstellung die Objektentfernung (*Depth from Focus*). Bei Kameras mit hoher Tiefenschärfe, wie es beispielsweise bei Weitwinkelobjektiven mit kurzen Brennweiten der Fall ist, ist die Entfernungsmessung allerdings sehr ungenau. In jedem Fall ist durch die Nachjustierung der Kamera die gleichzeitige Messung unterschiedlich weit entfernter Objekte ebenso unmöglich wie die Auswertung von vorher aufgezeichneten Bildern.

Relative Größe: Durch Erfahrung existiert die Kenntnis von „üblichen“ realen Größen bestimmter Objekte (Menschen, Fahrzeuge, Bäume, u.a.). Durch die tatsächlich wahrgenommene Größe sind Informationen über deren Entfernung möglich, denn hintere Objekte sind auf den erzeugten Bildern kleiner als vordere. Damit sind absolute Entfernungsmessungen nur dann möglich, wenn auch die absolute Größe eines Objektes bekannt ist. Darüber hinaus kann die Änderung der Größe als Bewegung zum Betrachter hin bzw. weg interpretiert werden.

Licht und Schatten: Aufgrund von Erfahrungen entsteht beim Menschen die unbewusste Annahme, dass Licht von oben kommt. Helle, beleuchtete Kanten werden als Oberkante, dunkle, im Schatten befindliche Kanten als Unterkante wahrgenommen. Damit ist es möglich, Löcher von Ausbuchtungen zu unterscheiden, ebenso sind andere Formen rekonstruierbar (*Shape from Shading*). Allerdings wird die Kenntnis oder die Annahme einer Lichtquelle aus einer bestimmten Richtung benötigt. Andernfalls führt dies zu Fehlinterpretationen der Tiefe.

Bewegungs-Parallaxe: Bewegte Objekte scheinen sich um so schneller zu bewegen, je näher sie sich am Betrachter befinden. Gerade bei der Beobachtung aus einem Fahrzeug wird dies deutlich und liefert einen starken Tiefeneindruck (*Shape from Motion*). Dies ist auch die Grundlage der Stereobildauswertung. Zwischen zwei verschiedenen Aufzeichnungsperspektiven hat sich die Welt relativ zur Kamera bewegt, was bei näher befindlichen Objekten zu einem größeren Unterschied bei der Projektion führt als bei entfernteren Objekten (*Binokulare Disparität*).

Es existiert eine Vielzahl von weiteren Möglichkeiten, aus Bildinhalten wie Verdeckungen, Texturen, Kanten und Farben die dreidimensionale Form und die Entfernung von Objekten zu bestimmen [Jäh02, Kap. 8], [SS00, Kap.12]. Diese Verfahren spielen in dieser Arbeit allerdings keine Rolle.

Hier wird mit Hilfe der Stereobildauswertung die dreidimensionale Umgebung rekonstruiert und ausgewertet. Bewegungen sind dazu nicht notwendig – für die Erstellung von Tiefeninformationen genügt die Auswertung eines Stereobildpaares. Darüber hinaus ist noch die abgebildete Größe eines Objektes für die Kamerakalibrierung von Bedeutung. Mit der Kalibrierung wird die Berechnung von absoluten Objektpositionen ermöglicht.

1.3 Stand der Technik

Verschiedene Sensoren bieten noch mehr Möglichkeiten, als die Umgebung rein visuell zu erfassen. Für die Gewinnung von Tiefeninformation zur Kollisionsvermeidung haben sich bisher vor allem *aktive* Sensoren, d.h. solche, die im Gegensatz zu *passiven* Sensoren wie Kameras selbst Energie in Form von Radio- Licht- oder Schallwellen an die Umgebung abgeben, bewährt. In Fahrzeugen sind diese Systeme bereits im Einsatz.

Beisielsweise funktioniert die Abstandsmessung des *Distronic*-Systems von Mercedes-Benz mit Hilfe von Radartechnik – die zusätzliche Auswertung von Kamerainformationen befindet sich in der Erprobungsphase. Hierbei handelt es sich um ein Fahrassistenzsystem, welches den Abstand zum vorhergehenden Fahrzeug misst. Die Information wird zur automatischen Geschwindigkeitsanpassung in Fahrzeugen der Oberklasse genutzt [Böh03, Kap. 2].

In autonomen Fahrzeugen (UGVs[10]) hat sich die *Lidar*[11]-Technik als besonders geeignet erwiesen. Das Team der Universität Stanford verwendet hauptsächlich Lasersensoren zur Erkennung von Hindernissen – mit seinem VW Touareg konnte es 2005 beim *DARPA Grand Challenge*[12] einen Sieg erringen [Mon05]. Andere Teams wie *Sandstorm 2004* verwendeten ebenfalls Laserscanner zur Kollisionsvermeidung, die Erprobung von Stereokameras war jedoch noch nicht erfolgreich [Urm05, S. 79].

Ein wesentlicher Vorteil aktiver Sensorik gegenüber Kameras ist die Funktionsfähigkeit auch bei schlechter Sicht, beispielsweise bei Nacht. Die Reichweiten sind je nach Einsatzbereich akzeptabel – bei Radar bis zu mehreren Kilometern. Bei Lasersensoren liegt die Reichweite je nach Gerät zwischen etwa 30 und wenigen hundert Metern, einige Geräte ermöglichen auch die Entfernungsmessung bis in den Kilometerbereich. Die Reichweite von Sonar ist an der Luft vergleichsweise gering, es lassen sich Abstände von bis zu etwa zwei Metern messen. Dieser Sensor wird daher vor allem für autonomes Landen oder Parken eingesetzt.

Ein großer Nachteil ist der erhöhte Energieverbrauch und die leichtere Detektierbarkeit durch das Empfangen der ausgesendeten Energie. Darüber hinaus spielt bei einem Kleinhubschrauber das Gewicht der mitgeführten Sensorik eine entscheidende Rolle. Radar- oder Lasersensoren von meist mehreren Kilogramm übertreffen die Nutzlast bei weitem, während Kameras bedeutend leichter und damit für diesen Einsatzbereich geeigneter sind.

Allerdings ist die Bildverarbeitung zur Detektion von Hindernissen und zur Kollisionsvermeidung noch nicht so weit fortgeschritten, dass sie in sicherheitsrelevanten Bereichen eingesetzt werden kann – bewährte aktive Sensorik liefert brauchbarere Ergebnisse. Die Erprobung findet

[10] Unmanned Ground Vehicle

[11] Light Detection And Ranging – Messung von Entfernungen mit Hilfe von Laserstrahlen.

[12] Wettbewerb, bei dem autonome Fahrzeuge gegeneinander antreten. Er wird seit 2004 von der DARPA (Defense Advanced Research Projects Agency), dem Forschungszweig des US-Verteidigungsministeriums, veranstaltet.

hingegen in vielen Bereichen statt – sowohl auf dem Gebiet der Fahr- und Flugzeugtechnik, als auch bei meist kleineren Robotern. Es zeigt sich aber, dass die Verwendung von Kameras zu diesem Zweck durchaus möglich ist. Die Erstellung von Tiefenbildern geschieht dabei nicht nur durch die Verwendung von lediglich zwei Kameras. Es werden ebenfalls Multikamerasysteme eingesetzt, die aus drei oder fünf Einzelkameras bestehen [Kan+96].

Die Berechnung von Tiefeninformationen aus Stereobildern ist sehr rechenintensiv, was eine Echtzeitbildverarbeitung nur mit sehr leistungsfähigen Computern ermöglicht. Seit September 2005 wird von der Firma *Videre Design* eine Stereokamera mit integrierter Tiefenbilderzeugung auf einem *FPGA*[13] angeboten. Die Geschwindigkeit bei der Stereobilderzeugung von 30 Hz übertrifft nach Herstellerangaben die Rechenleistung von Personalcomputern des Jahres 2005 bei weitem.

1.4 Vorgehensweise

Die Arbeit gliedert sich in fünf Hauptkapitel. Kapitel 2 erklärt die theoretischen Grundlagen der Kamerageometrie, die für die Stereobildauswertung relevant sind. Die Gewinnung von Objektdaten aus Bildern unter Berücksichtigung der kameraspezifischen Eigenschaften steht dabei im Vordergrund. Es wird ebenfalls erläutert, welche Parameter dazu benötigt und wie sie ermittelt werden. Zusätzlich werden grundlegende Bildverarbeitungsoperationen vorgestellt, die in dieser Arbeit Verwendung finden.

Kapitel 3 stellt die Softwareumgebung vor, die auf dem Bildverarbeitungsrechner des ARTIS zum Einsatz kommt und befasst sich mit den notwendigen Änderungen, die für die Stereobildverarbeitung erforderlich waren. In dieser Umgebung werden aus Stereobildern Tiefeninformationen extrahiert. Grundlegende Verfahrensweisen zur Tiefenbilderzeugung, die Einbindung einer Programmierschnittstelle, die diese Bilder erzeugt und die weitere Optimierung dieser Tiefenbilder werden in Kapitel 4 beschrieben. Mögliche Probleme bleiben dabei nicht unberücksichtigt.

Die Auswertung der Tiefenbilder zur Kollisionsvermeidung mit einem eigenen Ansatz erfolgt in Kapitel 5. Ziel ist die Messung von Abständen zu Hindernissen und das Finden von Ausweichkursen. In Kapitel 6 wird die Kollisionsvermeidung in einer Simulationsumgebung getestet und bewertet.

Das letzte Kapitel fasst die Ergebnisse abschließend zusammen und gibt einen Ausblick auf zukünftige Forschungsarbeiten.

[13] Field Programmable Gate Array – Programmierbarer Logikschaltkreis

Kapitel 2

Grundlagen

2.1 Bilderzeugung mit einer Kamera

Mathematisch betrachtet ist die Erzeugung von digitalen Bildern mit Hilfe von Kameras eine Projektion des dreidimensionalen Raumes auf eine zweidimensionale[1] Ebene. Bei Digitalkameras erfolgt eine Diskretisierung der Ebene in einzelne Bildpunkte (Pixel[2]) und die Diskretisierung der Helligkeiten. Der folgende Abschnitt befasst sich mit den Grundlagen der Projektion unter vereinfachten und realen Bedingungen und stellt die Zusammenhänge zwischen Objekt- und Bildkoordinaten dar.

2.1.1 Zentralperspektive

Die Zentralperspektive beschreibt den Zusammenhang zwischen Objekt- und Bildkoordinaten mit Hilfe der perspektivischen Projektion durch eine Lochkamera. Dieses Modell ist eine Vereinfachung im Vergleich zur Beschreibung von Projektionen bei Aufnahmen mit realen Kameras. Die Lichtstrahlen werden nicht gebrochen. Alle Strahlen zwischen Objekt- und Bildpunkt schneiden sich in einem gemeinsamen Brennpunkt, d.h. die Blendenöffnung ist unendlich klein und es entsteht dadurch keine Unschärfe bei der Abbildung. Für scharfe Bilder ist keine Fokussierung auf eine bestimmte Entfernung erforderlich, umgekehrt ist aber auch keine Entfernungsmessung über den Fokus möglich.

[1] Bei Zeilenkameras erfolgt eine Projektion auf eine Dimension. Diese werden hier nicht betrachtet.
[2] Kunstwort aus den englischen Wörtern *Picture* und *Element*

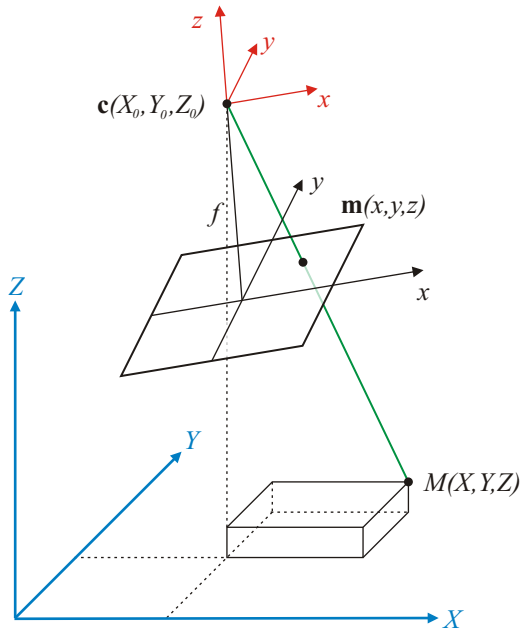


Abbildung 2.1: Zusammenhang zwischen Objekt- und Bildkoordinaten. Der Objektpunkt M wird auf den Punkt $\mathbf{m}$ in der Bildebene abgebildet. Der Brennpunkt der Kamera ist O [Kra97].

Wie in Abbildung 2.1 dargestellt, ist ein Bildpunkt $\mathbf{m}(x, y, z)$ aus dem Objektpunkt M (X, Y, Z) und der Orientierung der Kamera berechenbar. Nach der Gleichung der Zentralperspektive gilt

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + \lambda^{-1} \cdot \mathbf{R}(\omega, \varphi, \kappa) \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix}. \tag{2.1}$$

Die Aufzeichnungsparameter sind eine Skalierung λ, der Bildhauptpunkt (x_0, y_0, z_0), das Kamerazentrum $\mathbf{c}\,(X_0, Y_0, Z_0)$ und die Blickrichtung der Kamera in Form einer Rotationsmatrix $\mathbf{R}$, die hier in Abhängigkeit der drei Rotationswinkel ω, φ und κ angegeben ist. Rotationsmatrizen sind orthogonal, deren Determinante ist 1 [RW97, S. 107]. Sie geben eine Hintereinanderausführung der Rotationen mit den Winkeln ω um die X-, φ um die Y- und κ um die Z-Achse an und werden auch als *XYZ-Eulerwinkel* bezeichnet [MSK+04, S. 42]. Die Ausführung der einzelnen Rotationsschritte ist nicht kommutativ. Andere Reihenfolgen sind prinzipiell möglich, die Ausführungsreihenfolge muss daher stets bekannt sein.

In diesem Fall ist $\mathbf{R}(\omega, \varphi, \kappa) = \mathbf{R}_z(\kappa) \cdot \mathbf{R}_y(\varphi) \cdot \mathbf{R}_x(\omega)$ mit den Rotationen um die einzelnen Achsen[3]

$$\mathbf{R}_x(\omega) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{pmatrix}, \mathbf{R}_y(\varphi) = \begin{pmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{pmatrix} \text{ und}$$

[3]Die Rotationsreihenfolge ist dabei von rechts nach links zu lesen.

$$\mathbf{R}_z(\kappa) = \begin{pmatrix} \cos\kappa & -\sin\kappa & 0 \\ \sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.2}$$

damit ist

$$\mathbf{R}(\omega,\varphi,\kappa) = \begin{pmatrix} \cos\varphi\cos\kappa & \cos\omega\sin\kappa + \sin\omega\sin\varphi\cos\kappa & \sin\omega\sin\kappa - \cos\omega\sin\varphi\cos\kappa \\ -\cos\varphi\sin\kappa & \cos\omega\cos\kappa - \sin\omega\sin\varphi\sin\kappa & \sin\omega\cos\kappa + \cos\omega\sin\varphi\sin\kappa \\ \sin\varphi & -\sin\omega\cos\varphi & \cos\omega\cos\varphi \end{pmatrix}.$$

Die einzelnen Elemente der Rotationsmatrix werden wie folgt bezeichnet:

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}. \tag{2.3}$$

Alternativ ist es auch möglich, die Rotation durch einen Einheitsvektor $\mathbf{v} = (v_x, v_y, v_z)^T$ als Drehachse und einen Drehwinkel α anzugeben. Um auf die Winkelangabe in der vierten Dimension verzichten zu können, kann ein Drehvektor so definiert werden, dass die Richtung die Drehachse und die Länge den Drehwinkel in Radiant angibt. Die *Rodrigues*-Formel ermöglicht die Rotation um einen Vektor ohne Berechnung der Rotationsmatrix [För99]. Darüber hinaus können Rotationen auch mit Hilfe von Quaternionen beschrieben werden [DKL98].

2.1.2 Kollinearitätsgleichungen

Aus der Gleichung der Zentralperspektive lassen sich die *Kollinearitätsgleichungen* herleiten, die den Zusammenhang zwischen dem Objektpunkt M und dem Bildpunkt $\mathbf{m}$ beschreiben [Kra97, S. 14], [Luh00, S. 226ff.].

Die Transformation von Welt- in Bildkoordinaten ist

$$x = x_0 - f \cdot \frac{r_{11}\cdot(X - X_0) + r_{12}\cdot(Y - Y_0) + r_{13}\cdot(Z - Z_0)}{r_{31}\cdot(X - X_0) + r_{32}\cdot(Y - Y_0) + r_{33}\cdot(Z - Z_0)}, \tag{2.4}$$

bzw.

$$y = y_0 - f \cdot \frac{r_{21}\cdot(X - X_0) + r_{22}\cdot(Y - Y_0) + r_{23}\cdot(Z - Z_0)}{r_{31}\cdot(X - X_0) + r_{32}\cdot(Y - Y_0) + r_{33}\cdot(Z - Z_0)}. \tag{2.5}$$

Umstellung der Kollinearitätsgleichungen ergibt beispielsweise folgende Rücktransformation, die bei zusätzlicher Kenntnis der Objektkoordinate Z angewendet werden kann:

$$X = X_0 + (Z - Z_0) \cdot \frac{r_{11}\cdot(x - x_0) + r_{21}\cdot(y - y_0) - r_{31}\cdot f}{r_{13}\cdot(x - x_0) + r_{23}\cdot(y - y_0) - r_{33}\cdot f}, \tag{2.6}$$

bzw.

$$Y = Y_0 + (Z - Z_0) \cdot \frac{r_{12}\cdot(x - x_0) + r_{22}\cdot(y - y_0) - r_{32}\cdot f}{r_{13}\cdot(x - x_0) + r_{23}\cdot(y - y_0) - r_{33}\cdot f}. \tag{2.7}$$

Durch Kenntnis einiger Parameter lassen sich die Punkte $M(X, Y, Z)$ und $\mathbf{m}(x, y)$ ineinander überführen. Es wird zwischen Parametern der inneren und äußeren Orientierung unterschieden:

Innere Orientierung: Sie beschreibt die Eigenschaften der Kamera. Im idealen Lochkameramodell sind dies die Kamerakonstante bzw. Brennweite (f) und der Bildhauptpunkt (x_0, y_0). Bei realen Kameras darüber hinaus noch Parameter, welche die Verzeichnung des Bildes durch die Kameraoptik beschreiben. Die innere Orientierung ermöglicht eine Transformation von Bildpunkten in ein kamerafestes Koordinatensystem und umgekehrt.

Äußere Orientierung: Sie beschreibt die Position des Projektionszentrums $\mathbf{c}(X_0, Y_0, Z_0)$ und Rotation $(\omega, \varphi, \kappa)$, d.h. Blickrichtung der Kamera in einem Weltkoordinatensystem. Die Parameter ermöglichen eine Transformation von Kamera- in Weltkoordinaten und umgekehrt.

Da bei der Projektion eines Objektpunktes auf die Bildebene eine Dimension verloren geht, entsteht ein Informationsverlust. Aus einem einzigen Bild ist es auch mit Hilfe der Kameraparameter nicht möglich, aus einem Bildpunkt den Objektpunkt zu berechnen. Die Umrechnung von Bild- in Objektkoordinaten benötigt daher zusätzliche Informationen.

2.1.3 Homogene und normierte Bildkoordinaten

Unter Verwendung homogener Koordinaten für die Punkte $\mathbf{m} = (x, y)$ und $M = (X, Y, Z)$, d.h. ein Vektor $\mathbf{x} = (x_1, ..., x_n)$ wird durch Erweiterung mit 1 als $\widetilde{\mathbf{x}} = (x_1, ..., x_n, 1)$ dargestellt, ist die Projektion $\widetilde{\mathbf{m}}$ eines Raumpunktes $\widetilde{M}$ auf die Kameraprojektionsebene durch die Gleichung

$$z\,\widetilde{\mathbf{m}} = \mathbf{P}\,\widetilde{M} \tag{2.8}$$

mit der Skalierung z und der 3×4-Abbildungsmatrix $\mathbf{P}$ darstellbar. Translation und Rotation sind dabei im Gegensatz zur vorgestellten Gleichung der Zentralperspektive rein multiplikativ. Die Abbildungsmatrix $\mathbf{P}$ bildet die Raumkoordinaten auf Kamerakoordinaten ab und lässt sich zerlegen in

$$\mathbf{P} = \mathbf{K}[\mathbf{R}\,\mathbf{t}]. \tag{2.9}$$

Die Kameramatrix $\mathbf{K}$ beschreibt die innere Orientierung und die zusammengesetzte Matrix $[\mathbf{R}\,\mathbf{t}]$, die aus der Rotationsmatrix $\mathbf{R}$ und dem Translationsvektor $\mathbf{t}$ besteht, die äußere Orientierung der Kamera.

Die Kameramatrix $\mathbf{K}$ ist

$$\mathbf{K} = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.10}$$

mit den Brennweiten f_x und f_y in x- bzw. y-Richtung, der Schiefe s und dem Bildhauptpunkt (x_0, y_0) der aufgezeichneten Bildpunkte. Die Verwendung von $f_x \neq f_y$ ermöglicht die Transformation in eine Bildebene, in der die Einheiten in x- bzw. y-Richtung verschiedene Größen haben, bei $s \neq 0$ sind die Achsen nicht orthogonal.

Sind die inneren Parameter der Kamera nicht bekannt, lässt sich durch Festlegung der Brennweite $f_x = f_y = 1$, der Schiefe $s = 0$ und des Bildhauptpunktes $(x_0, y_0) = (0, 0)$ die Projektion auf eine Ebene beschreiben, welche parallel zur Bildebene ist. Es ist

$$z \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = [\mathbf{R}\,\mathbf{t}] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2.11}$$

bzw.

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{K} \cdot \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}. \tag{2.12}$$

Die Koordinaten dieser Punkte (x', y') sind im Vergleich zu den eigentlichen Bildpunkten (x, y) verschoben und skaliert.

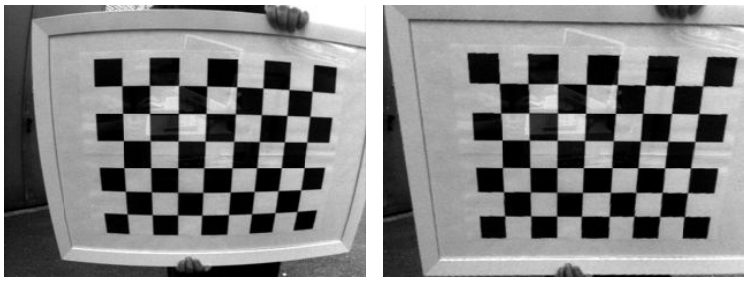
Umgekehrt ist mit Hilfe der inneren Parameter eine Normierung von Bildkoordinaten möglich. Durch Umkehrung der Gleichung 2.12 können aus aufgezeichneten Bildkoordinaten (x, y) normierte Bildkoordinaten (x', y') erstellt werden. Bildpunkte in normierten Koordinaten[4] entsprechen einer Projektion mit der Brennweite 1 und dem Bildhauptpunkt (0,0).

2.1.4 Verzeichnung

Die Verzeichnung der Kamera bedeutet, dass Bildpunkte nicht so auf die Bildebene projiziert werden, wie es die Gleichungen der Zentralperspektive darstellen. Dort wird von einem Lochkameramodell ausgegangen, bei dem Lichtstrahlen nicht gebrochen werden. Bei realen Kameras führen Linsen im Objektiv zur Brechung der Lichtstrahlen, abhängig von Abstand zur optischen Achse, Einfallswinkel und Farbe. Je nach Kamera erscheinen die Bilder mehr oder weniger stark verzerrt. Mit Hilfe von Kameramodellen lässt die Abweichung eines bei der Aufzeichnung entstehenden, verzerrten Bildpunktes vom unverzerrten Bildpunkt mathematisch und kameraspezifisch beschreiben.

Das Verzeichnungsmodell nach Brown [FB86] ist die Grundlage vieler Standardprogramme und wird auch von der verwendeten Software zur Bestimmung der inneren Kameraparameter verwendet. Es beschreibt die Zuordnung vom unverzeichnetem Bildpunkt (x'_u, y'_u)

[4]Sie werden auch als kalibrierte Bildpunkte oder Bildpunkte einer kalibrierten Kamera bezeichnet.



(a) (b)

Abbildung 2.2: Aufgenommenes, verzeichnetes Bild (a) und ein mit Hilfe des vorgestellten Modells daraus berechnetes, unverzeichnetes Bild (b) eines Kalibriermusters.

zum verzeichnetem Bildpunkt (x'_d, y'_d) in normierten Koordinaten mit $x'_d = x'_u + \Delta x'$ bzw. $y'_d = y'_u + \Delta y'$ und unterscheidet drei verschiedene Verzeichnungsarten.

Die Gesamtverzeichnungen $\Delta x'$ und $\Delta y'$ eines Bildpunktes lassen sich unterteilen in

$$\Delta x' = \Delta x'_{\text{sym}} + \Delta x'_{\text{asy}} + \Delta x'_{\text{aff}}, \tag{2.13}$$

bzw.

$$\Delta y' = \Delta y'_{\text{sym}} + \Delta y'_{\text{asy}} + \Delta y'_{\text{aff}}. \tag{2.14}$$

Radiale Verzeichnung entsteht vor allem bei der Brechung der Lichtstrahlen durch die Linse. Mit zunehmendem Abstand von der Linsenmitte werden Objekte kleiner, besonders im im Randbereich des Bildes wird dies deutlich. Diese Art der Verzerrung macht bei gewöhnlichen Kameras den größten Anteil aus. Besonders stark tritt dieser Effekt bei der Verwendung von Weitwinkelobjektiven mit kurzen Brennweiten auf („Fischaugen-Optik“). Der symmetrische Anteil, d.h. die *radial-symmetrische* Verzeichnung wird durch die drei Parameter k_1, k_2, k_3 beschrieben. Es ist

$$\Delta x'_{\text{sym}} = x'_u \cdot (k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2.15}$$

und

$$\Delta y'_{\text{sym}} = y'_u \cdot (k_1 r^2 + k_2 r^4 + k_3 r^6). \tag{2.16}$$

Dabei bezeichnet r mit $r^2 = x'^2_u + y'^2_u$ den Abstand der unverzerrten Koordinaten vom Bildhauptpunkt.

Tangentiale Verzeichnungen resultieren beispielsweise aus Fertigungstoleranzen bei der Platzierung der Linse. Dadurch trifft die optische Achse nicht exakt die Mitte des Bildsensors. Zusammen mit der *radial-asymmetrischen* Verzeichnung und wird dies durch die Parameter p_1 und p_2 beschrieben.

Es ist

$$\Delta x'_{\text{asy}} = 2p_1 x'_u y'_u + p_2(r^2 + 2x'^2_u) \tag{2.17}$$

und

$$\Delta y'_{\text{asy}} = 2p_2 x'_u y'_u + p_1(r^2 + 2y'^2_u). \tag{2.18}$$

Unterschiede zwischen Breite und Länge verschiedener Bildsensorzellen erzeugen eine *Affinität* des Bildes. Zusätzlich können *Scherungen* auftreten, sofern die Elemente nicht exakt orthogonal platziert sind. Beide Faktoren werden mit den Parametern b_1 und b_2 modelliert. Es ist

$$\Delta x'_{\text{aff}} = x'_u b_1 + y'_u b_2 \tag{2.19}$$

und

$$\Delta y'_{\text{aff}} = 0. \tag{2.20}$$

Dieses Verzeichnungsmodell hat sich in vielen Anwendungen bewährt und liefert recht gute Ergebnisse, wobei teilweise auf einige Parameter verzichtet werden kann. Oftmals reichen die Parameter k_1, k_2, p_1 und p_2 für eine akzeptable Genauigkeit aus. Ein geeignetes Verfahren zur Schätzung der Parameter dieses Modells wird von Heikkilä und Silvén [HS97] vorgestellt.

Der Nachteil des Modells ist allerdings, dass eine Invertierung nicht möglich ist. Die effiziente Umrechnung von verzeichneten in unverzeichnete Punkte erfolgt daher durch iterative Schätzung.

Dazu wird um den verzeichneten Punkt (x'_d, y'_d) ein Bereich festgelegt, in dem der unverzeichnete Punkt gesucht wird. Durch Rasterung dieses Bereiches entsteht eine endliche Anzahl von Schätzpunkten, für die die verzeichneten Punkte berechnet werden. Derjenige Schätzpunkt, dessen verzeichneter Punkt den geringsten Abstand zu (x'_d, y'_d) aufweist, dient als Ausgangspunkt für den nächsten Iterationsschritt. Es wird ein neuer Suchbereich um diesen Punkt von kleinerer Größe und Rasterweite zwischen den einzelnen Schätzpunkten festgelegt – der Abstand eines besten verzeichneten Schätzpunktes zum gesuchten Punkt (x'_d, y'_d) ist demzufolge geringer. Ist die gewünschte Genauigkeit durch Unterschreiten eines bestimmten Abstandes erreicht, so bricht die Iteration ab und der entsprechende unverzeichnete Schätzpunkt wird als (x'_u, y'_u) ausgegeben.

Neben diesem vorgestellten Kameramodell existieren weitere, unter anderem das von Tsai [Tsa87], dessen Vorteil die problemlose Invertierbarkeit ist. Godding [God02] und Luhmann [Luh00] verweisen darüber hinaus auf Ansätze, bei denen die Schrägstellung des Sensors zur optischen Achse oder eine Verzeichnung in Abhängigkeit von der Objektentfernung berücksichtigt wird. Kaufmann und Ladstädter [KL05] stellen ein Verfahren zur Beseitigung farbabhängiger Projektionsfehler vor.

2.1.5 Bestimmung der Kameraposition

Die in der Kollisionsvermeidung verwendete äußere Orientierung der Kamera soll eine Zuweisung zwischen Bild- und Objektpunkten in der Art ermöglichen, dass der Abstand der Objektpunkte zum Helikopter in allen drei Dimensionen ermittelt werden kann. Die Position des Helikopters in erdfesten Koordinaten muss dazu nicht bekannt sein.

Um eine Verwechslung mit geodätischen Weltkoordinaten auszuschließen, bezieht sich die äußere Orientierung der Kamera in dieser Arbeit auf ein flugkörperfestes Koordinatensystem[5] und beschreibt das *Boresight Alignment* bzw. *Misalignment* zwischen Kamera und Helikopter [Cra02]. Die Bezugskoordinaten werden im Folgenden als Helikopterkoordinaten mit den Achsen x_f, y_f und z_f bezeichnet.

Ohne Verwendung dieser äußeren Orientierung erfolgt eine Festsetzung von Rotation und Translation gleich null und dementsprechend eine Zuordnung von Punkten in einem kamerafesten Koordinatensystem mit den Achsenbezeichnungen x_c, y_c und z_c.

Da der Punkt des Kamerazentrums nicht mit dem Bezugspunkt des Helikopters übereinstimmen muss, ist eine Translation erforderlich. Die Verschiebung wird mit einem Vektor angegeben, der die Entfernung der Kamera von einem definierten Punkt des Helikopters enthält.

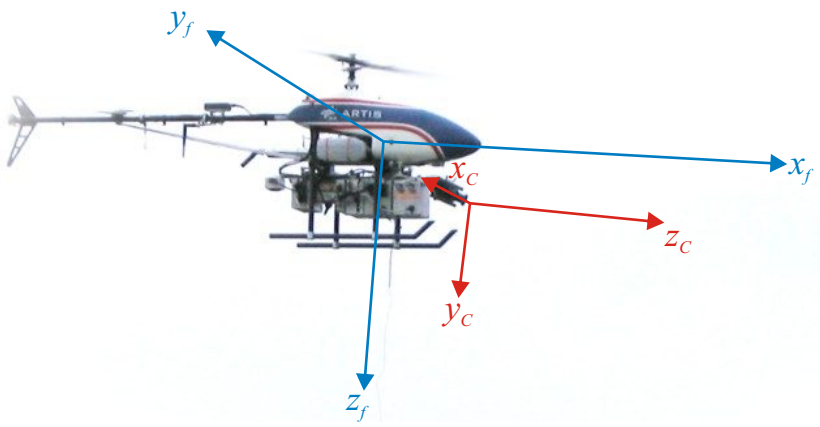


Abbildung 2.3: Kamera- und Helikopterkoordinaten. Der Ursprung des Kamerakoordinatensystems ist das linke Kamerazentrum. Das Helikopterkoordinatensystem ist zu diesem verschoben und rotiert.

Des weiteren sind die Systeme zueinander rotiert. Während in der Kamera die Achsen x_c, y_c und z_c nach rechts, unten, bzw. vorne (Entfernung) zeigen, gelten für den Helikopter andere Achsen. In der Luftfahrt bezeichnen x_f, y_f und z_f aus Sicht des Flugkörpers die Längs(Roll)-,

[5] Als Ursprung des Koordinatensystems wird hier ein Punkt in der Mitte des Helikopters verwendet, um Abstände von Objekten in allen Richtungen gleich werten zu können. In anderen Anwendungen wie bei der Aufzeichnung von Karten, ist es sinnvoller, Bezug auf die Position des GPS-Empfängers, bei der bildgestützten Lagekorrektur Bezug auf das Inertialmesssystem (IMU) zu nehmen.

Quer(Nick)- und Hoch(Gier)achse [Buc02, S. 86]. Koordinaten mit positiven Werten in diesen Richtungen geben also die Entfernungen in die Richtungen nach vorne, rechts, bzw. unten an. Ist die optische Achse der Kamera parallel zur Rollachse, zeigt die Kamera exakt nach vorne. Das Kamerakoordinatensystem ist in diesem Fall um jeweils 90° in z- und x-Richtung gedreht.

Auch wenn in vielen Anwendungen, wie dem bildgestützten Landen [SMS03], die Annahme getroffen wird, dass die Kameraposition am Helikopter exakt bekannt ist, muss bei dieser Anwendung eine Untersuchung stattfinden, inwieweit dies zutrifft. Möglicherweise lässt sich die Lage der Kamera nicht präzise genug ermitteln. Vor allem aber werden bei der Verwendung einer Stereokamera die Bilder nachträglich verdreht, deren Achsen weichen somit von der Montagerichtung der Kamera ab. Gründe und Berechnungsvorschriften werden in Abschnitt 2.2.2 ab Seite 19 erläutert. Aus den Bildern bekannter Objekte lässt sich die Position der Kamera bestimmen.

Ein Algorithmus für die Positionsbestimmung, basierend auf der Arbeit von Sharp et al. [SSS01], ist im ARTIS-Projekt bereits für andere Anwendungen in Verwendung. Dabei wird die Translation und Rotation zu einem bekannten planaren Muster bestimmt, siehe Anhang A.1 ab Seite 100. Ist die Position des Musters vom Helikopter aus gesehen durch Vermessen bekannt und die Position des Musters bezüglich der Kamera aus den Bilddaten errechnet, so lässt sich daraus die Position der Kamera ermitteln. Es ist allerdings zu beachten, dass die Vermessung der Musterposition in Helikopterkoordinaten ebenfalls eine Ungenauigkeit besitzt.

Es seien die Lagebeziehungen zwischen Punkten $\mathbf{p}$ in Kamerakoordinaten $\mathbf{p}_c$ und Muster $\mathbf{p}_m$ errechnet worden, sowie die Beziehung zwischen Helikopter $\mathbf{p}_f$ und Muster $\mathbf{p}_m$ in der Form bekannt. Punkttransformationen von Koordinatensystem a nach Koordinatensystem b seien allgemein durch die Rotation $\mathbf{R}_{ab}$ bzw. Translation $\mathbf{t}_{ab}$ beschrieben, wobei die Indizes m für das Muster-, c für das Kamera- und f für das Helikopterkoordinatensystem verwendet werden.

Es sei

$$\mathbf{p}_f = \mathbf{R}_{mf} \cdot \mathbf{p}_m + \mathbf{t}_{mf} \quad \text{, bzw.} \quad \mathbf{p}_m = \mathbf{R}_{cm} \cdot \mathbf{p}_c + \mathbf{t}_{cm}. \tag{2.21}$$

Durch Einsetzen folgt

$$\begin{aligned} \mathbf{p}_f &= \mathbf{R}_{mf} \cdot (\mathbf{R}_{cm} \cdot \mathbf{p}_c + \mathbf{t}_{cm}) + \mathbf{t}_{mf} \\ \mathbf{p}_f &= \underbrace{\mathbf{R}_{mf}\,\mathbf{R}_{cm}}_{\mathbf{R}_{cf}} \cdot \mathbf{p}_c + \underbrace{\mathbf{R}_{mf}\,\mathbf{t}_{cm} + \mathbf{t}_{mf}}_{\mathbf{t}_{cf}} \\ \mathbf{p}_f &= \mathbf{R}_{cf} \cdot \mathbf{p}_c + \mathbf{t}_{cf}, \end{aligned} \tag{2.22}$$

womit durch $\mathbf{R}_{cf}$ und $\mathbf{t}_{cf}$ eine Transformation von Punkten aus Kamera- in Helikopterkoordinaten beschrieben ist.

Umgekehrt ergibt sich $\mathbf{p}_c = \mathbf{R}_{cf}^{-1} \cdot (\mathbf{p}_f - \mathbf{t}_{cf})$. Da es sich bei Rotationsmatrizen um orthogonale Matrizen handelt, gilt $\mathbf{R}_{cf}^{-1} = \mathbf{R}_{cf}^{T}$. Die Transformation von Helikopter- in Kamerakoordinaten kann demnach mit der transponierten Rotationsmatrix erfolgen, es ist

$$\mathbf{p}_c = \mathbf{R}_{cf}^{T} \cdot (\mathbf{p}_f - \mathbf{t}_{cf}). \tag{2.23}$$

Wie schon erwähnt, müssen Bilder von einem bekannten Kalibriermuster für die Bestimmung der Transformationsparameter aufgezeichnet werden. Deren Verwendung setzt eine konstante Kameraposition während des Fluges voraus, da eine Messung nur am Boden vorgenommen werden kann. Laufende Anpassungen, wie sie bei einer schwenkbaren Kamera vorkommen, müssen daher gesondert betrachtet werden.

Alternativen zu dem verwendeten Algorithmus sind photogrammetrische Verfahren, die auch zur Bestimmung der Kameraposition in erdfesten Koordinaten genutzt werden können. Die Position wird beispielsweise mit der *Direkten Linearen Transformation* oder dem *Location Determination Problem* geschätzt, wobei letzteres nach Döring [Dör05] genauere Werte liefert. Mit Hilfe des *Räumlichen Rückwärtsschnittes* kann iterativ die Genauigkeit verbessert werden.

Diese Verfahren ermöglichen im Gegensatz zu dem hier verwendeten eine Positionsbestimmung auch dann, wenn das untersuchte Muster nicht planar ist. Ebenso ist die nachträgliche Positionsbestimmung aus den Bilddaten möglich, so dass die Kameraposition in diesem Koordinatensystem während eines Fluges nicht konstant sein muss. Einsatzzweck ist vor allem die Vermessung von Gebäuden oder Landschaften, wo eine sehr hohe Genauigkeit erforderlich ist und die Positionsbestimmung nicht über ein ebenes Muster realisiert werden kann [Kra97].

2.2 Bilderzeugung mit zwei Kameras

Wird mit einer einzigen Kamera ein Bild aufgezeichnet, entsteht durch die Projektion ein Informationsverlust durch die Verringerung der Punktdimension. Die Abbildung ist nicht injektiv – ein Bildpunkt kann einem Objektpunkt nicht eindeutig zugeordnet werden. Allerdings ist die Zuordnung zu einer Linie, auf der der Objektpunkt liegen muss, möglich. Mit Hilfe einer zweiten Linie aus einer anderen Perspektive, welche die erste schneidet, lässt sich der Originalpunkt rekonstruieren.

Genau dies wird bei der Auswertung von Stereobildern ausgenutzt. Die Kameras sehen die 3D-Umgebung aus verschiedenen Positionen und liefern dadurch unterschiedliche Projektionen. Durch Kombination beider Bilder und dem Vergleich der Bildpunkte, die das gleiche Objekt darstellen, wird eine Berechnung der dreidimensionalen Koordinaten des Objektpunkts möglich. Insbesondere sind dadurch Tiefeninformationen verfügbar.

2.2.1 Epipolargeometrie

Die Geometrie der Projektion eines Punktes auf zwei Bildebenen wird als *Kern-* oder *Epipolargeometrie* bezeichnet [Luh00, S. 239]. Dabei können die Kameras beliebige Lagen besitzen, also zueinander verschoben und rotiert sein. Abbildung 2.4 zeigt ein Beispiel einer solchen Projektion.

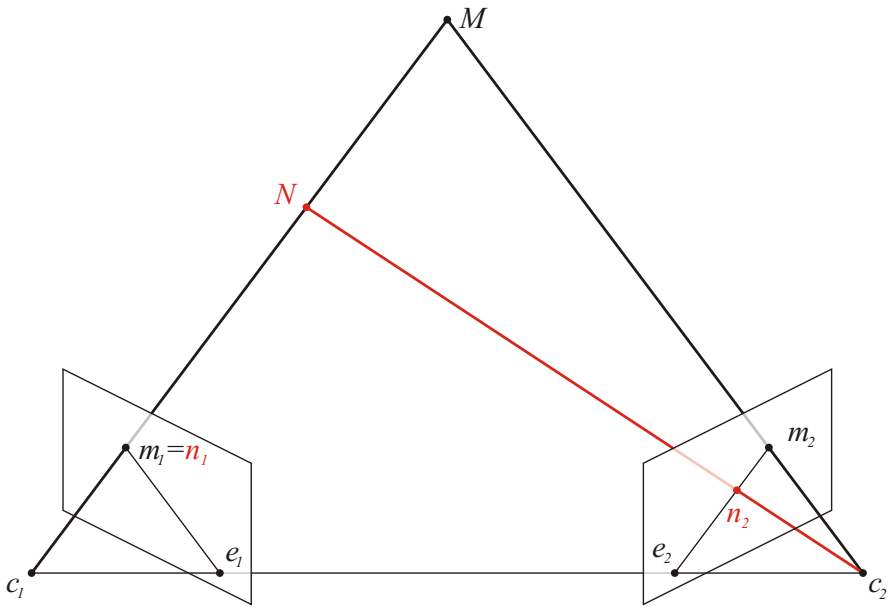


Abbildung 2.4: Epipolargeometrie. Die linke Kamera bildet den Punkt M auf $\mathbf{m}_1$, die rechte Kamera auf $\mathbf{m}_2$ ab. Der Unterschied zwischen M und N ist von der linken Kamera aus nicht erkennbar, die rechte Kamera bildet diesen Punkt auf $\mathbf{n}_2$ ab, welcher auf der Epipolarlinie $\mathbf{e}_2\mathbf{m}_2$ liegt. Die beiden Epipole $\mathbf{e}_1$ und $\mathbf{e}_2$ entstehen durch die Projektion des jeweils anderen Projektionszentrums auf die Bildebenen.

Der Punkt M wird auf die beiden Bildebenen abgebildet. Es fällt auf, dass sich aus Sicht der linken Kamera kein Unterschied zwischen M und einem anderen Punkt N auf der Geraden zwischen M und dem Projektionszentrum $\mathbf{c}_1$ feststellen lässt. Allerdings ist der Unterschied von der anderen Kamera feststellbar. Hier wird dieser Punkt auf einen anderen Bildpunkt $\mathbf{m}_2'$ anstelle von $\mathbf{m}_2$ abgebildet. Allgemein wird die Linie $\mathbf{c}_1M$, welche im ersten Bild auf den Punkt $\mathbf{m}_1$ zusammenfällt, im zweiten Bild auf die Linie $\mathbf{e}_2\mathbf{m}_2 := \mathbf{l}_2$, eine so genannte *Epipolarlinie*, abgebildet. Umgekehrt ist die Epipolarlinie $\mathbf{e}_1\mathbf{m}_1 := \mathbf{l}_1$ das Bild der Linie $\mathbf{c}_2M$ in der ersten Kamera. Die Punkte $\mathbf{e}_1$ bzw. $\mathbf{e}_2$ sind die Schnittpunkte der Linie zwischen den beiden Projektionszentren mit den jeweiligen Bildebenen, genannt *Epipole*. Die Punkte $\mathbf{c}_1$, $\mathbf{c}_2$ und M spannen die *Epipolarebene* des Punktes M auf, alle Punkte dieser Ebene werden auf die gleichen Epipolarlinien abgebildet. Alle Epipolarlinien schneiden den jeweiligen Epipol [HZ00, Kap. 8] [MSK+04, Kap. 5,6].

Die Beziehung zweier korrespondierender Punkte $\mathbf{m}_1$ und $\mathbf{m}_2$ ist durch die Fundamentalmatrix $\mathbf{F}$ beschrieben. Für die beiden Punkte in homogenen Koordinaten $\widetilde{\mathbf{m}}_1$ bzw. $\widetilde{\mathbf{m}}_2$ ist

$$\widetilde{\mathbf{m}}_2^T \mathbf{F} \widetilde{\mathbf{m}}_1 = 0. \tag{2.24}$$

Die Fundamentalmatrix ergibt sich aus der relativen Orientierung, d.h. der Translation und Rotation der Kameras zueinander und den beiden Kameramatrizen. Eine ausführliche Erläuterung findet sich in Anhang A.2, Seite 103. Ebenso definiert die Fundamentalmatrix die den Punkten zugehörigen Epipolarlinien auf der jeweils anderen Bildebene. Es ist

$$\mathbf{l}_2 = \mathbf{F}\widetilde{\mathbf{m}}_1 \quad \text{, bzw.} \quad \mathbf{l}_1 = \mathbf{F}^T \widetilde{\mathbf{m}}_2, \tag{2.25}$$

wobei eine Epipolarlinie $\mathbf{l} = (a, b, c)^T$ mit den Koeffizienten der Geradengleichung $au+bv+c = 0$ dargestellt wird.

Ist die Fundamentalmatrix bekannt, gestaltet sich die Suche nach einem korrespondierenden Punkt in dem anderen Bild wesentlich einfacher – dieser muss zwangsläufig auf der durch die Fundamentalmatrix definierten Epipolarlinie liegen[6]. Der Suchbereich innerhalb der Bildebene wird dadurch von zwei auf eine Dimension eingeschränkt.

Das Prinzip zur Berechnung der Fundamentalmatrix basiert darauf, geeignete Punkte eines Bildes im jeweils anderen Bild, so genannte *Passpunktpaare*, zu finden. Wie auch bei der Aufzeichnung eines Musters bei der Positionsbestimmung sind dabei Ecken im Muster bzw. im jeweiligen Bild, besonders geeignet. Möglichkeiten zur Findung von Ecken werden in Abschnitt 2.4.1 ab Seite 25 genannt.

Die Korrespondenz der jeweiligen Punkte muss bekannt sein – in der Praxis lässt sich dies am besten mit der Aufnahme eines bekannten Musters realisieren, bei denen die Features eindeutig zugeordnet werden können. Die Verwendung spezieller Kalibrierbilder ist nicht erforderlich, da es Verfahren zur Aussortierung falscher Zuordnungen gibt. Da Bildpunkte verrauscht sind und die so errechnete Matrix eine Ungenauigkeit besitzt, lässt sich bei der Verwendung von einer hohen Anzahl von Passpunktpaaren eine größere Genauigkeit erzielen. Es ist naheliegend, ein Muster mit besonders vielen Ecken zu verwenden und mehrere Bilder aufzuzeichnen.

Fehlerhafte Berechnungen entstehen bei ungenauen Passpunkten sowie falschen Punktzuordnungen. Dies ist bei Bildern der Fall, aus denen keine oder nur wenige geeignete Punkte extrahiert werden können. Solche Bilder entstehen bei allzu verschiedenen Blickrichtungen der Kameras, kontrastarmen Motiven oder unscharfen Aufnahmen. Hartley [Har95a] zeigt Beispiele korrekter und fehlerhafter Berechnungen. Bei weniger als sieben Punktpaaren ist die Matrix nicht berechenbar. Algorithmen zur Errechnung der Fundamentalmatrix sind in Anhang A.3 ab Seite 104 erläutert.

[6] Ist die Bildebene parallel zu dieser Epipolarebene, existiert für M keine Epipolarlinie in der anderen Bildebene und somit kein korrespondierender Punkt. Sind die Kamerazentren gleich, existiert für alle Punkte keine Epipolarebene und damit keine Korrespondenz. Durch entsprechende Anordnung der Kameras wird angenommen, dass diese Fälle nicht eintreten.

2.2.2 Rektifikation

Die Rektifikation ist eine Vorverarbeitung der Bilder mit dem Ziel, besonders gut vergleichbare Bilder zu produzieren. Die Bilder werden auf neue Bildebenen projiziert, die bestimmte Eigenschaften erfüllen und für die weitere Verarbeitung eine Vereinfachung sind. Werden rektifizierte Bilder ausgewertet, muss die Veränderung der Projektionsebenen und damit die veränderte Blickrichtung der Kamera berücksichtigt werden.

In diesem Fall wird eine *standard-stereoskopische* Anordnung der beiden Bilder erzeugt. Dabei werden beide Bilder auf eine neue, gemeinsame Bildebene projiziert, welche parallel zur Verbindungslinie der beiden Kamerazentren ist. Die Bildachsen werden so gedreht, dass die x-Achse parallel zu dieser Verbindungslinie und die y-Achse senkrecht zu dieser ist. Die beiden Bildhauptpunkte liegen auf der gleichen y-Koordinate dieser gemeinsamen Ebene.

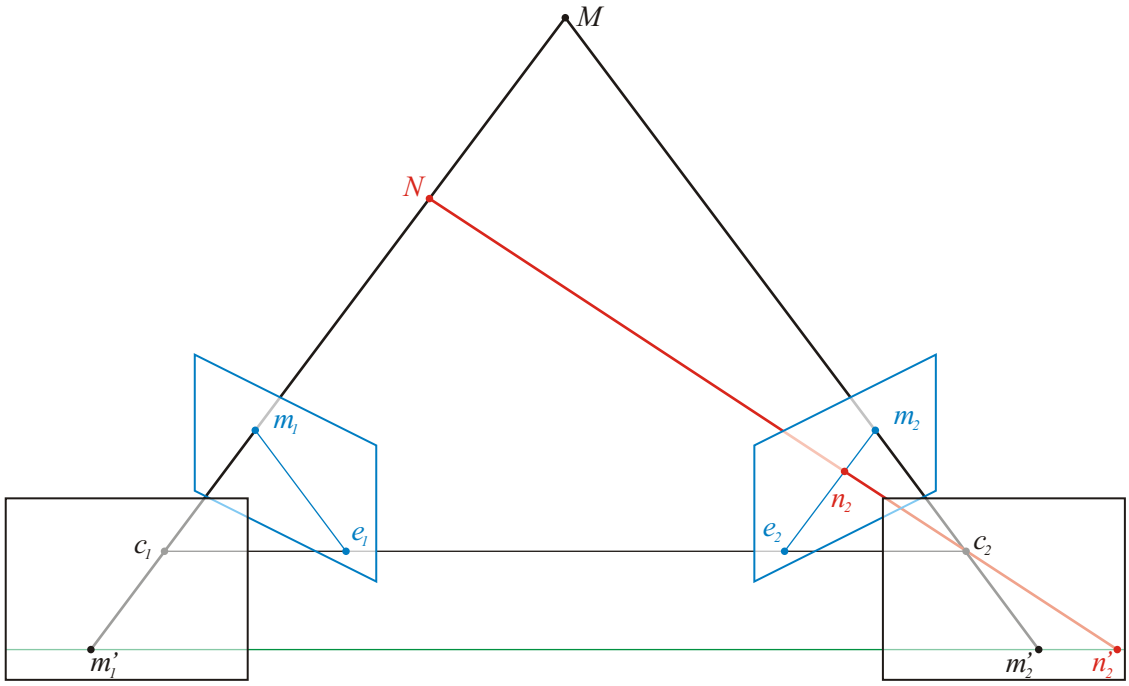


Abbildung 2.5: Rektifikation. Es erfolgt eine Rückprojektion der Bildpunkte der ursprünglichen Bildebenen (blau) auf neue Bilder (schwarz), die auf einer gemeinsamen Ebene liegen. Sie ist parallel zur Verbindungsgeraden der beiden Kamerazentren $\mathbf{c}_1$ und $\mathbf{c}_2$. Damit sind alle Epipolarlinien (grün) parallel zueinander und die Epipole befinden sich im Unendlichen.

Nach Ayache und Hansen [AH88] sei die Rektifizierung der Bildpunkte ausgehend von den Kamerazentren $\mathbf{c}_1$ und $\mathbf{c}_2$ und den ursprünglichen perspektivischen Projektionsmatrizen $\mathbf{P}$ und $\mathbf{Q}$ wie folgt zu berechnen: Es sei

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \tag{2.26}$$

die Projektionsmatrix der linken Kamera mit den Zeilen[7] $\mathbf{p}_j = (p_{j1}, p_{j2}, p_{j3})^T$ (mit $j \in \{1;2;3\}$). Analog sei die Schreibweise der Elemente von $\mathbf{Q}$, der Projektionsmatrix der rechten Kamera, festgelegt. Die Bildpunkte sind $\mathbf{m}_1 = (x_1, y_1)$ bzw. $\mathbf{m}_2 = (x_2, y_2)$. Dann sind die rektifizerten Bildpunkte

$$\begin{pmatrix} u_1 \\ v_1 \\ w_1 \end{pmatrix} = \mathbf{R}_1 \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} u_2 \\ v_2 \\ w_2 \end{pmatrix} = \mathbf{R}_2 \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} \tag{2.27}$$

mit den rektifizierten Bildkoordinaten

$$\mathbf{m}'_1 = \begin{pmatrix} x'_1 \\ y'_1 \end{pmatrix} = \begin{pmatrix} u_1/w_1 \\ v_1/w_1 \end{pmatrix} \quad \text{, bzw.} \quad \mathbf{m}'_2 = \begin{pmatrix} x'_2 \\ y'_2 \end{pmatrix} = \begin{pmatrix} u_2/w_2 \\ v_2/w_2 \end{pmatrix}. \tag{2.28}$$

Die Rektifikationsmatrizen sind

$$\mathbf{R}_1 = \begin{pmatrix} (\mathbf{c}_1 \times \mathbf{c}_2) \times \mathbf{c}_1)^T \\ (\mathbf{c}_1 \times \mathbf{c}_2)^T \\ ((\mathbf{c}_1 - \mathbf{c}_2) \times (\mathbf{c}_1 \times \mathbf{c}_2))^T \end{pmatrix} \begin{pmatrix} \mathbf{p}_2 \times \mathbf{p}_3 & \mathbf{p}_3 \times \mathbf{p}_1 & \mathbf{p}_1 \times \mathbf{p}_2 \end{pmatrix}, \tag{2.29}$$

bzw.

$$\mathbf{R}_2 = \begin{pmatrix} (\mathbf{c}_1 \times \mathbf{c}_2) \times \mathbf{c}_2)^T \\ (\mathbf{c}_1 \times \mathbf{c}_2)^T \\ ((\mathbf{c}_1 - \mathbf{c}_2) \times (\mathbf{c}_1 \times \mathbf{c}_2))^T \end{pmatrix} \begin{pmatrix} \mathbf{q}_2 \times \mathbf{q}_3 & \mathbf{q}_3 \times \mathbf{q}_1 & \mathbf{q}_1 \times \mathbf{q}_2 \end{pmatrix}. \tag{2.30}$$

Für korrespondierende Punkte $\mathbf{m}'_1$ und $\mathbf{m}'_2$ der hierbei entstandenen rektifizierten Bilder gilt

$$y'_1 = y'_2, \tag{2.31}$$

die Punkte besitzen also keinen vertikalen Abstand zueinander.

Dieses Verfahren ist für die Bildrektifikation geeignet, sofern die inneren und äußeren Orientierungen der Kameras bekannt sind. Eine vorhergehende Kalibrierung der Kameras liefert die notwendigen Orientierungsparameter. Die Anwendung eignet sich dann, wenn die Kameras fest installiert sind, also deren Parameter sich nicht ändern und eine Kalibrierung zur Verfügung steht, die die notwendigen Parameter liefert. Fusiello et al. [FTV97] geben eine Erweiterung und Implementation dieser Rektifikationsmethode an.

Eine andere Methode zur Erstellung von rektifizierten Bildern ist die Verwendung von Homographien. Dabei werden die internen Kameraparameter nicht benötigt, allerdings die Lage der Kameras zueinander in Form von der Fundamentalmatrix. Sind zwei Bilder rektifiziert, entsprechen also einer standard-stereoskopischen Anordnung der beiden Kameras, so hat die Fundamentalmatrix $\bar{\mathbf{F}}$ die Form der antisymmetrischen Kreuzproduktmatrix des Vektors $(1,0,0)^T$, also

$$\bar{\mathbf{F}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}. \tag{2.32}$$

[7] nur die jeweils ersten drei Elemente einer Zeile

Werden die ursprünglichen Bildpunkte $\widetilde{\mathbf{m}}_1$ und $\widetilde{\mathbf{m}}_2$ mit den Homographien $\mathbf{H}_1$ bzw. $\mathbf{H}_2$ in der Form

$$\bar{\mathbf{m}}_1 = \mathbf{H}_1 \widetilde{\mathbf{m}}_1 \quad \text{und} \quad \bar{\mathbf{m}}_2 = \mathbf{H}_2 \widetilde{\mathbf{m}}_2 \tag{2.33}$$

rektifiziert, folgt

$$\begin{aligned} \bar{\mathbf{m}}_2^T \bar{\mathbf{F}} \bar{\mathbf{m}}_1 &= 0 && (2.34)\\ (\mathbf{H}_2 \widetilde{\mathbf{m}}_2)^T \bar{\mathbf{F}} \mathbf{H}_1 \widetilde{\mathbf{m}}_1 &= 0 && (2.35)\\ \widetilde{\mathbf{m}}_2^T \mathbf{H}_2^T \bar{\mathbf{F}} \mathbf{H}_1 \widetilde{\mathbf{m}}_1 &= 0, && (2.36) \end{aligned}$$

und es ergibt sich die Faktorisierung der Fundamentalmatrix F in der Form

$$\mathbf{F} = \mathbf{H}_2^T \bar{\mathbf{F}} \mathbf{H}_1. \tag{2.37}$$

Möglichkeiten, diese Homographien zu finden, werden unter anderem von Hartley [Har95b] und Oram [Ora01] gezeigt.

Es ist offensichtlich, dass bei der Umrechnung der Bildkoordinaten nicht alle neu entstehenden Bildpunkte innerhalb der Begrenzung des neuen Bildes sein müssen. Ebenso kann es vorkommen, dass nicht für alle Punkte des neuen Bildes Informationen aus dem unrektifizierten Bild vorliegen, da die entsprechenden Koordinaten außerhalb des Bildbereiches liegen. Diese Bereiche bleiben somit im rektifizierten Bild leer. In der Praxis entsteht dadurch oft ein Informationsverlust im Randbereich des Bildes, wenn die Bildbegrenzung angepasst wird, ein herein- oder herauszoomender Effekt entsteht und die Ränder abgeschnitten werden.

Um den Informationsverlust bzw. die Abweichung von den ursprünglichen Bildern möglichst gering zu halten, sollte die Anordnung der Kameras entsprechend gewählt werden. Dies bedeutet eine Montage mit lediglich horizontalem Abstand, parallelen optischen Achsen und die Verwendung von Kameras mit gleichen Brennweiten. Die Rektifikation ist dann allerdings immer noch notwendig, um Abweichungen durch ungenaue Anbringungsmöglichkeiten und Fertigungstoleranzen ausgleichen zu können.

Die anschließende Diskretisierung der Bildkoordinaten zur Erstellung von Digitalbildern erzeugt ebenso Fehler im resultierenden Bild. Unter anderem entstehen Aliasing-Effekte bei schrägen Kanten, Unschärfen bei der perspektivischen Vergrößerung und Informationsverluste bei der Verkleinerung. Die Qualität der rektifizierten Bilder ist dadurch verschlechtert, bei der Auswertung von Stereobildern ist dies in vielen Fällen jedoch hinnehmbar.

2.2.3 Der Idealfall: Achsparallele Stereogeometrie

Durch Entzerrung und anschließende Rektifikation werden Bilder geschaffen, die der standardstereoskopischen Anordnung entsprechen. Dadurch ist die Berechnung der Projektion stark

vereinfacht. Wie in Abbildung 2.5 erkennbar ist, sind die Epipolarlinien parallel zueinander und ebenfalls parallel zur x-Achse der Bildebenen. Dadurch wird ein Objektpunkt bei beiden Bildern auf die gleiche Zeile projiziert – zwischen den korrespondierenden Bildpunkten entsteht lediglich ein horizontaler Abstand.

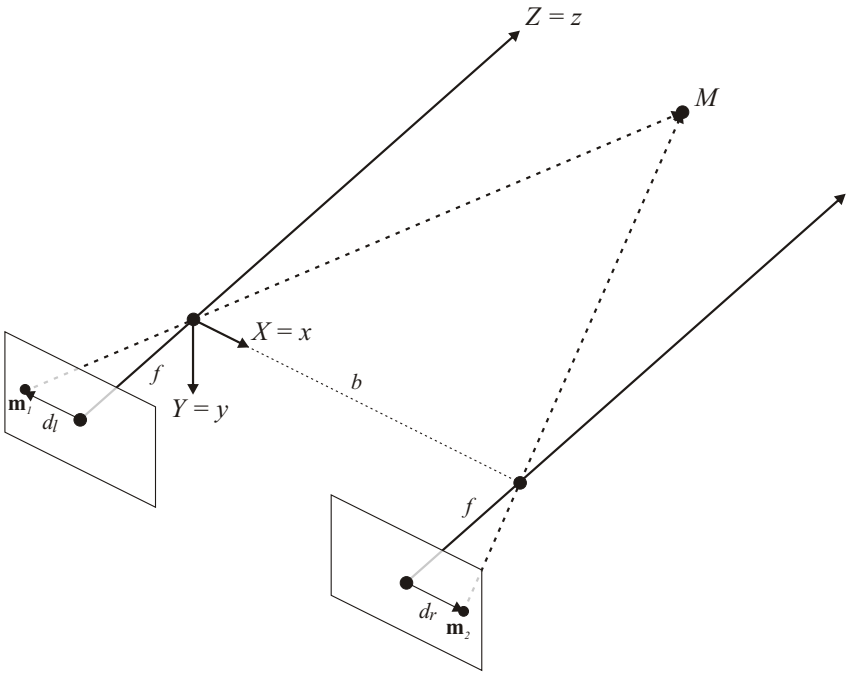


Abbildung 2.6: Achsparalleles Lochkameramodell. Der Punkt $M(X, Y, Z)$ wird auf die beiden Bildebenen, die jeweils den Abstand f zum Brennpunkt besitzen, projiziert. Der horizontale Abstand vom Zentrum beträgt d_l bzw. d_r, d.h. für die Punkte $\mathbf{m}_1(x_1, y_1)$ und $\mathbf{m}_2(x_2, y_2)$ ist $x_1 - x_2 = d_l - d_r$ (Vektoraddition). Da die Kameras keinen vertikalen Abstand zueinander besitzen, ist der Abstand des projizierten Punktes M in dieser Richtung bei beiden Bildern gleich, also $y_1 = y_2$.

Die Projektion eines Punktes $M(X, Y, Z)$ im idealen Lochkameramodell folgt aus der Gleichung der Zentralperspektive. Bezogen auf das kamerafeste Koordinatensystem mit Ursprung im linken Kamerazentrum ist die Rotation gleich null, d.h. die Rotationsmatrix ist die Einheitsmatrix. Die Translation ist ebenfalls null, bis auf die x-Koordinate bei Abbildung in der rechten Kamera. Unter Verwendung homogener Koordinaten ist die Abbildungsmatrix aus Gleichung 2.8 (Seite 10) in diesem Fall

$$\mathbf{P} = \begin{pmatrix} f & 0 & x_0 & b\,f \\ 0 & f & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{2.38}$$

sie ist das Produkt aus Kamera- und Projektionsmatrix. Dabei handelt es sich um Parameter der Bilder, die dem achsparallelen und idealen Lochkameramodell entsprechen. Es bezeichnen f die Brennweite oder Kammerkonstante und x_0 bzw. y_0 die Koordinaten des Bildhauptpunktes. Für die linke (Referenz-) Kamera wird $b = 0$ gesetzt, bei der rechten Kamera ist b die Basis, also der horizontale Abstand der beiden Kameras.

2.3 Bestimmung der Orientierung

Die Kalibrierung der Stereokamera dient zur Ermittlung der inneren und äußeren Parameter jeder Kamera. Hierzu wird die Programmbibliothek *Small Vision System* [SVS] verwendet, dort existiert das Programm *smallvcal* für diesen Zweck [KB04]. Dabei werden bis zu zehn Stereobilder eines bekannten planaren Kalibriermusters aufgezeichnet. In dieser Arbeit wurde ein Schachbrettmuster mit 6×8 inneren Ecken verwendet, pro Stereobild existieren also 48 Passpunktpaare. Der Abstand der Ecken beträgt 54 Millimeter.

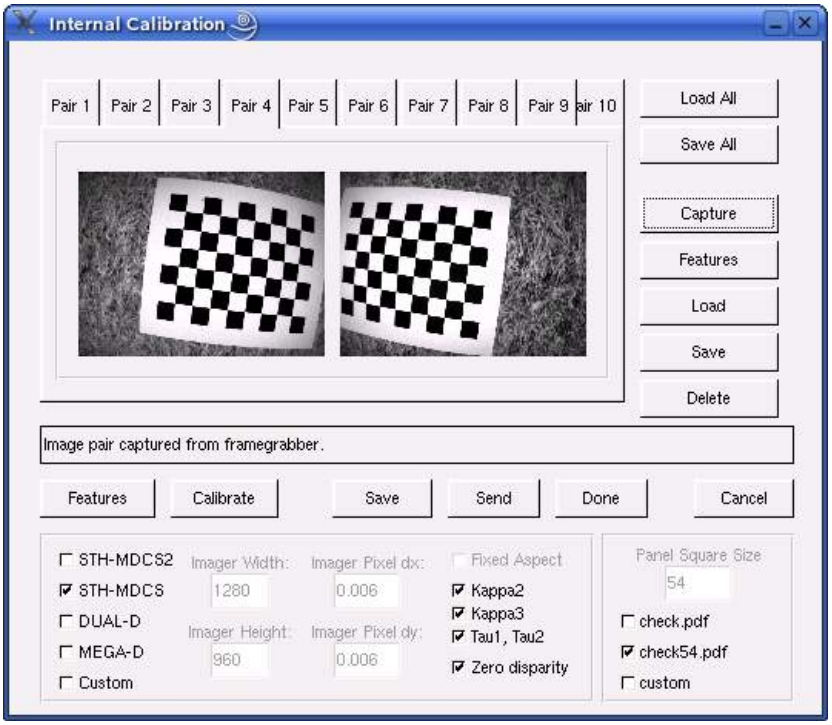


Abbildung 2.7: Das Programm *smallvcal* zur Kalibrierung der Stereokamera. Aus den Positionen der Ecken des Musters werden die inneren Orientierungen der beiden Kameras und die Lage zueinander errechnet.

Der Programmablauf ist folgender:

- Bestimmung der *inneren Parameter* beider Kameras: Brennweiten, Bildhauptpunkt und Verzeichnungsparameter. Es wird das in 2.1.4 vorgestellte Modell verwendet, allerdings ohne die Parameter b_1 und b_2. Zusätzlich kann die Verwendung der verbleibenden Parameter eingeschränkt werden.
- Bestimmung der *relativen Orientierung* der beiden Kameras. Das Koordinatensystem[8] ist so definiert, dass der Ursprung im linken Kamerazentrum liegt. Die Achsen x und y entsprechen den Richtungen einer Bildzeile bzw. Spalte und z der optischen Achse. Die

[8]Dieses Koordinatensystem ist kamerafest und wird auch als Kamerakoordinatensystem bezeichnet.

äußeren Parameter der rechten Kamera in diesem System werden aus den vorhandenen Bilddaten errechnet und geben somit dessen Position und Rotation bezogen auf die linke Kamera an.

- Bestimmung der *Rektifikations-* und *Projektionsmatrizen* für jede Kamera. Die Projektionsmatrizen beziehen sich auf die rektifizierten Bilder.

Das Ergebnis dieser Kalibrierung kann in einer Datei abgelegt und so für die Entzerrung und Rektifikation späterer Bildaufnahmen verwendet werden. Werden keine Änderungen an der Stereokamera vorgenommen, lassen sich Bilder erzeugen, die dem achsparallelen Lochkameramodell entsprechen.

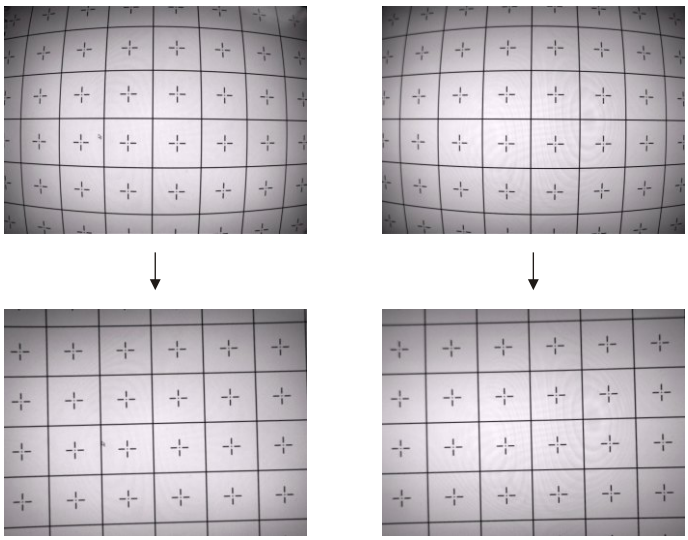


Abbildung 2.8: Beispiel für ein aufgezeichnetes (oben) und idealisiertes Stereobildpaar (unten). Die Verzeichnung der Linsen verursacht die Krümmung der Linien in den oberen beiden Bildern, welche durch die Entzerrung entfernt wird. Die Rektifikation kann aufgrund der Translation und Rotation der Kameras zueinander eine Drehung der Bilder zur Folge haben, wodurch die Linien in den unteren Bildern nicht exakt horizontal bzw. vertikal sind.

Die Bestimmung der Kameraposition erfolgt mit Hilfe eines eigens angelegten Programmes[9] unter Einbeziehung des in Anhang A.1 (Seite 100) vorgestellten Algorithmus. Dieser bestimmt die Lage des Musters bezogen auf die Kamera und rotiert die Koordinaten um 90° um die z- und x-Achse. So lässt sich die äußere Orientierung einer Kamera bezogen auf den Helikopter bestimmen. Hierbei ist lediglich die Bestimmung der Orientierung der linken Kamera erforderlich. Für die Positionsbestimmung wird das rektifizierte Bild verwendet, da das verwendete Stereofilter (Kapitel 4, Seite 41ff.) die Tiefe der Punkte dieses Bildes liefert und aus diesem die Helikopterkoordinaten einzelner Punkte ermittelt werden.

[9]Es handelt sich um ein Filter innerhalb des dip-Frameworks, das noch vorgestellt wird.

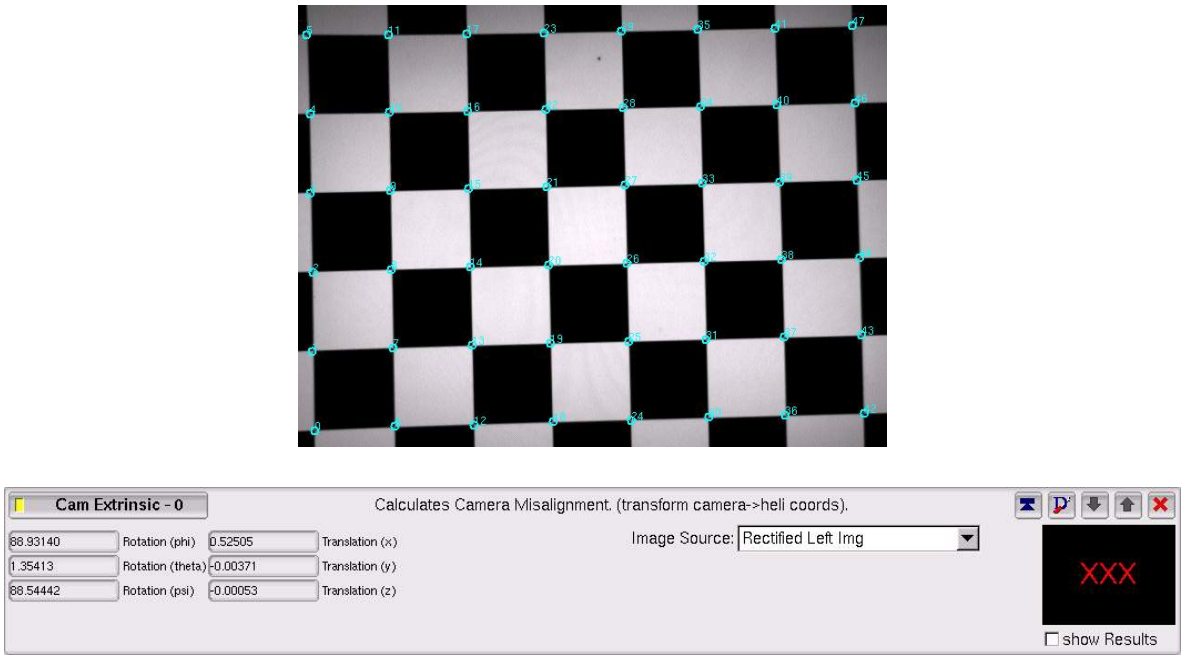


Abbildung 2.9: Bestimmung der Position der Kamera (unten) durch Aufzeichnung eines bekannten Musters (oben), dessen Lage zum Helikopter bekannt ist. Zur Positionsbestimmung werden die Ecken des Musters verwendet, die im Bild blau gekennzeichnet sind. Die Rotationswinkel sind hier mit ϕ, θ und ϕ bezeichnet.

2.4 Bildverarbeitung

Dieser Abschnitt ist ein Exkurs zu einigen Verfahren der digitalen Bildverarbeitung, die im Rahmen dieser Arbeit verwendet werden. Es erfolgt eine Darstellung verschiedener Ansätze, bestimmte Aufgaben der Bildverarbeitung zu bewerkstelligen und die beispielhafte Erläuterung einzelner Algorithmen.

2.4.1 Detektion von Features

Als *Features* werden all jene Punkte eines Bildes bezeichnet, in deren Umgebung sich signifikante Grauwertänderungen in mehr als einer Richtung feststellen lassen. Dies sind vor allem Ecken und Schnittpunkte mehrerer Kanten. Das Ergebnis einer Erkennung von Ecken ist im oberen Bild von Abbildung 2.9 zu sehen.

Solche Features lassen sich unter anderem mit dem *Plessey Corner Detector* von Harris und Stephens [HS88], dem Operator von Förstner und Gülch [FG87], dem von Shi/Tomasi vorgestellten *Feature Tracker* [ST94] oder dem *SUSAN Two Dimensional Feature Detector* von Smith und Brady [SB95] finden. Letzterer sei hier exemplarisch erläutert. Es existieren viele weitere Ansätze zur Findung von Features, eine umfassende Übersicht befindet sich in der Arbeit von Smith und Brady. Dort werden auch die Unterschiede zwischen dem *Plessey*- und

dem *SUSAN*-Algorithmus bei der Erkennung von Ecken aufgezeigt. Darüber hinaus existiert unter anderem von Wang und Brady [WB92] ein Verfahren zur Verbesserung der Genauigkeit vorhandener Eckpunkte auf Subpixelebene, Torr [Tor02, S. 8] stellt dazu ebenfalls eine Möglichkeit vor.

Der SUSAN-Algorithmus untersucht für jeden Pixel in einer Umgebung, ob es sich um einen Eckpunkt handelt. Prinzipiell sind verschiedene Nachbarschaften der Bildpunkte als Umgebung denkbar, vor allem diejenigen, die nahezu kreisförmig sind. Für die Detektion hat sich die 37er[10] Nachbarschaft ($n_{\max} = 37$) bewährt, welche einen Kreis vom Radius 3,4 annähert, siehe Abbildung 2.10.

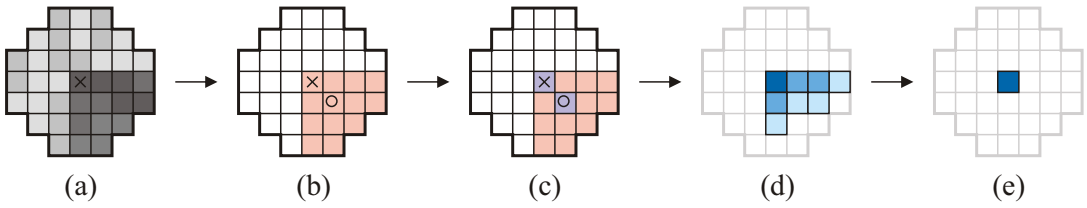


Abbildung 2.10: Funktionsweise der *SUSAN Corner Detection.* Auswahl einer Nachbarschaft um einen Pixel (a), Bestimmung der Region ähnlicher Pixel und dessen Größe (b) und Überprüfung der Linie zwischen Zentrum (×) und Schwerpunkt (○) (c). Resultat ist ein neues Bild, das die Stärke der Ecken repräsentiert (d), zusammengehörige Regionen werden auf den Punkt mit dem Maximalwert reduziert (e). Resultat sind die Eckpunkte des Ausgangsbildes.

In dieser Nachbarschaft wird das USAN[11] bestimmt – das Segment, welches die dem Mittelpunkt $\mathbf{r}_0$ „ähnlichen“ Bildpunkte $\mathbf{r}$ enthält. Eine einfache Möglichkeit ist es, diejenigen Punkte zu der Region zu zählen, dessen absolute Grauwertdifferenz zu $\mathbf{r}_0$ einen bestimmten Schwellwert t erreicht bzw. unterschreitet. Allerdings wird eine unscharfe Zuordnung verwendet, die durch die Gleichung

$$c(\mathbf{r}, \mathbf{r}_0) = e^{-\left(\frac{I(\mathbf{r}) - I(\mathbf{r}_0)}{t}\right)^6} \tag{2.39}$$

mit den Grauwerten $I(\mathbf{r})$ bzw. $I(\mathbf{r}_0)$ der Punkte $\mathbf{r}$ bzw. $\mathbf{r}_0$ beschrieben ist. Dabei wird $c = 1$ als die volle Zugehörigkeit, $c = 0$ als vollständiger Ausschluss eines Bildpunktes zu dem Segment interpretiert. Als Schwellwert ist bei gewöhnlichen 8-Bit-Grauwertbildern ein Wert von $t = 25$ geeignet, bei kontrastärmeren Bildern auch geringere Werte. Allerdings erhöht sich damit auch die Empfindlichkeit gegenüber Rauschen. Die Größe n des USAN in einer Nachbarschaft um $\mathbf{r}_0$ entsteht durch Aufsummieren, es ist

$$n(\mathbf{r}_0) = \sum_{r \in \mathrm{USAN}(\mathbf{r}_0)} c(\mathbf{r}, \mathbf{r}_0). \tag{2.40}$$

[10] inklusive dem Mittelpunkt

[11] Univalue Segment Assimilating Nucleus. *Nucleus* bezeichnet das Zentrum der untersuchten Region, also den Pixel, für den die Eigenschaft untersucht wird, ob es sich um einen Eckpunkt handelt.

Über einen geometrischen Schwellwert g wird die Stärke der Ecke bestimmt, es ist

$$R(\mathbf{r}_0) = \begin{cases} g - n(\mathbf{r}_0), & \text{falls } n(\mathbf{r}_0) < g, \\ 0, & \text{sonst}, \end{cases} \tag{2.41}$$

mit dem Schwellwert $g = n_{\max}/2$. Niedrigere Werte von g ermöglichen die Detektion ausschließlich „schärferer“ Ecken, höhere Werte lassen auch normale Kanten zu. Zur Unterdrückung von Punkten, die aufgrund von Rauschen hier fälschlicherweise als Ecke erkannt werden können, wird überprüft, ob alle Punkte auf der Linie zwischen Zentrum und Schwerpunkt des USAN innerhalb desselben liegen. Ist dem nicht so, wird die Ecke entfernt. Da es vorkommt, dass zu einer eigentlichen Ecke mehrere benachbarte Punkte als solche identifiziert werden, werden in jeder zusammenhängenden Menge von Eckpunkten diejenigen Punkte $\mathbf{r}_0$ unterdrückt, deren Wert $R(\mathbf{r}_0)$ nicht dem Maximum entspricht. Somit wird pro Ecke nur ein Punkt als Eckpunkt klassifiziert.

2.4.2 Vergleichsoperatoren

Für den Vergleich von Ähnlichkeiten zwischen verschiedenen Bildregionen oder abgebildeten Objektpunkten in mehreren Bildern ist es in den wenigsten Fällen ausreichend, einfach die Helligkeiten einzelner Pixel zu vergleichen. Es könnten rein zufällige Ähnlichkeiten auftreten oder tatsächliche Gemeinsamkeiten aufgrund von Rauschen nicht erkannt werden.

Haben zwei Pixel eine ähnliche Helligkeit, lässt dies also keine Schlussfolgerung zu, inwieweit diese Bildpunkte das gleiche reale Objekt repräsentieren oder zu einem ähnlichen Muster gehören. Allerdings werden Objekte im Bild, sofern sie eine bestimmte Größe überschreiten und somit als Objekte identifizierbar sind, von vielen benachbarten Bildpunkten dargestellt. Es ist daher naheliegend, die Umgebung des jeweiligen Bildpunktes für einen Vergleich zu berücksichtigen. Um einen Bildpunkt wird ein Fenster bestimmter Größe ($m \times n$) festgelegt, welches mit einem anderen Fenster gleicher Größe verglichen werden kann (*Block Matching*).

Es existiert eine Vielzahl von Abstandsmaßen, exemplarisch seien hier drei davon beschrieben. Sie werden auch von Lucas und Kanade [LK81], sowie Scharstein und Szeliski [SS01] für den Vergleich von Bildregionen genannt und kommen in vielen Verfahren zur Ermittlung von Bewegungen oder Unterschieden zwischen den Bildern eines Stereobildes zur Anwendung.

Diese Maße für die Ähnlichkeit zwischen den Blöcken p_1 und p_2 sind:

- Summe absoluter Differenzen: Der Betrag der Differenzen der einzelnen Elemente wird aufsummiert. Das Maß für den Unterschied zwischen den Grauwerten der Pixel $p(i, j)$ beider Blöcke (*Sum of Absolute Differences, SAD*) ist:

$$d_{\text{SAD}} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} |p_1(i, j) - p_2(i, j)|. \tag{2.42}$$

- Summe quadratischer Differenzen: Das Quadrat der Differenzen wird aufsummiert. Große Differenzen haben damit einen stärkeren Einfluss. Das Unterschiedsmaß (*Sum of Squared Differences, SSD*) ist:

$$d_{\text{SSD}} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (p_1(i,j) - p_2(i,j))^2. \tag{2.43}$$

- Korrelationskoeffizient: Der Quotient aus Kovarianz und Varianzen mit den Mittelwerten der Blöcke $\bar{p}_1$ und $\bar{p}_2$:

$$r = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \big(p_1(i,j) - \bar{p}_1\big) \cdot \big(p_2(i,j) - \bar{p}_2\big)}{\sqrt{\left(\sum_{i=1}^{m} \sum_{j=1}^{n} \big(p_1(i,j) - \bar{p}_1\big)^2\right) \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \big(p_2(i,j) - \bar{p}_2\big)^2\right)}} \tag{2.44}$$

Eine große Unähnlichkeit zeigt sich bei den Summen durch hohe Werte, beim Korrelationskoeffizient durch Werte nahe null. Kleine Summen und Korrelation nahe 1 signalisieren hohe Ähnlichkeiten. Der Korrelationskoeffizient ermöglicht zusätzlich eine Erfassung von umgekehrten Ähnlichkeiten, die sich in negativen Werten zeigen.

Es ist offensichtlich, dass größere Fenster eine höhere Toleranz gegenüber Bildrauschen besitzen. Allerdings können sehr kleine Objekte damit schlechter verglichen werden, da benachbarte Bildpunkte, die andere Objekte darstellen, nun einen viel größeren Einfluss haben.

2.4.3 Bildpyramiden

Die grundlegende Idee der Bildpyramide ist es, die Auflösung eines Bildes stufenweise zu reduzieren, um die Suche nach bestimmten Segmenten oder Mustern zu beschleunigen. Oftmals sind bestimmte Muster auch in Bildern mit reduzierter Auflösung wie in Abbildung 2.11 erkennbar. Die Mustererkennung erfolgt grob in einer hohen Ebene mit geringer Auflösung und wird anschließend in der darunter liegenden Ebene verfeinert, wobei durch die Vorausschätzung in der höheren Ebene nicht das gesamte Bild durchsucht werden muss. In vielen Anwendungen kann dadurch die Rechenzeit enorm verringert werden.

Die Verringerung der Auflösung verursacht einen Informationsverlust und bedeutet eine Reduzierung der Abtastfrequenz, wodurch Alias-Effekte entstehen können [Jäh02, S. 143]. Die Reduzierung der Auflösung durch einfaches Weglassen bestimmter Bildpunkte ist daher meist nicht geeignet.

Es existieren verschiedene Ansätze, durch Filterung des Bildes bestimmte Informationen bei der Auflösungsreduzierung zu erhalten und den Alias-Effekt zu verringern. Ein einfacher Ansatz zur Erstellung einer Bildpyramide ist es, mit jeder Stufe die Auflösung zu halbieren.

Abbildung 2.11: Beispiel für eine Bildpyramide. Bilder höherer Ebenen haben eine geringere Auflösung (hier jeweils um den Faktor 2 reduziert und zur besseren Darstellung gleich groß dargestellt). Für das Finden von Objekten (z.B. dem Hubschrauber) ist eine reduzierte Auflösung oft ausreichend (z.B. Ebene 2), für die grobe Unterscheidung zwischen Himmel und Landschaft genügt auch ein noch weiter verkleinertes Bild. Allerdings ist in höheren Ebenen ein gesuchtes Objekt oft nicht mehr erkennbar, so dass es sinnvoll ist, die Erstellung der Pyramide auf eine maximale Ebene zu begrenzen oder die Suche nicht in der höchsten Ebene zu beginnen.

Dabei werden jeweils 4 Pixel (in einem 2×2-Fenster) zu einem Bildpunkt der nächsthöheren Ebene zusammengefasst. Der Wert dieses Punktes ergibt sich dann z.B. aus dem Mittelwert der vier Ausgangswerte.

Ebenso ist es möglich, eine Umgebung eines Pixels zu betrachten und alle Punkte dieser Region mit in die Berechnung einfließen zu lassen, wobei sich Regionen durchaus überschneiden können. Typische Beispiele sind die 4er-, die 8er-Nachbarschaft oder auch größere Nachbarschaften. Der Wert des resultierenden Bildpunktes kann wieder der Mittelwert sein.

Gute Resultate liefern Gauß- oder Laplacepyramiden, wie sie unter anderem von Adelson et al. [Ade+84], Gonzalez und Wood [GW02, S. 351ff.] und Jähne [Jäh02, S. 143ff.] vorgestellt werden. Bei der Gaußpyramide bleiben durch Tiefpassfilterung gröbere Flächen erhalten, kleine Strukturen werden durch Glättung unterdrückt. Im Gegensatz dazu werden bei der Laplace-Pyramide durch Bandpassfilterung die Kanten verstärkt.

Ein Bildpunkt einer höheren Ebene repräsentiert damit eine quadratische Region von Bildpunkten der tiefer gelegenen Ebenen. Die Kantenlänge der einzelnen quadratischen Regionen der L-ten Ebene ist bei Halbierung der Auflösung mit jedem Schritt:

$$k(L) = 2^L, \tag{2.45}$$

demnach sind die Abstände zwischen Bildpunkten entsprechend größer. Die zweidimensionalen Regionen, also die Pixel der L-ten Ebene, bestehen aus $r(L) = k^2(L)$ Bildpunkten des Originalbildes, also

$$r(L) = (2^L)^2 = 2^{2L} = 4^L. \tag{2.46}$$

Die Anzahl der Bildpunkte der L-ten Ebene ist:

$$N(L) = \frac{1}{4^L} \cdot N(0). \tag{2.47}$$

Ebene 0 bezeichnet das Ausgangsbild. Es wird deutlich, dass die Anzahl der Bildpunkte exponentiell sinkt und Einzelheiten des Bildes möglicherweise noch erkennbar bleiben. Die Verwendung von Bildpyramiden ist daher sehr effizient. Die Pyramidenhöhe bzw. die Ebene, in der die Suche nach Mustern beginnt, hängt vom Anwendungsgebiet ab, in Abbildung 2.11 wird dies deutlich.

2.4.4 Musterverfolgung

Bei der Musterverfolgung wird davon ausgegangen, dass ein und dasselbe Objekt in verschiedenen Bildern auf ähnliche Weise dargestellt wird. Wird in einem anderen Bild dieses Muster wiedergefunden, handelt es sich möglicherweise um das gleiche Objekt. Mittels *Block Matching* wird das Bild nach Bereichen durchsucht, die dem gesuchten Muster ähnlich sind. Wird eine möglichst „ähnliche“ Region in einem anderen Bild wieder gefunden, so wird sie als „gefunden“ angesehen – es ist eine korrespondierende Region in einem anderen Bild und stellt mit gewisser Wahrscheinlichkeit dasselbe Objekt dar. Anwendung ist unter anderem der Stereobildvergleich zur Errechnung von Tiefenbildern und auch die Erkennung von Bewegungen in einer Bildsequenz.

Bei der Bewegungserkennung werden zwei aufeinander folgende Bilder miteinander verglichen. Wird ein gesuchte Muster in der Nähe wieder gefunden, lässt sich aus dem dadurch entstehenden optischen Fluss die Bewegung des Musters schließen. In der Praxis bedeutet dies, dass einzelne Punkte anvisiert und innerhalb der Bildfolge wiedererkannt werden, um daraus auf die Bewegung und die Geschwindigkeit von Objekten zu schließen.

Abbildung 2.12: Verfolgung eines Musters in einer Bildsequenz. Die grün gekennzeichnete Ecke des Fahrzeugs wird, einmal ausgewählt, in den Folgebildern wiedererkannt.

Bewegungen können nicht unter allen Voraussetzungen erkannt werden. Ist ein Bildbereich unstrukturiert, bewirkt die Bewegung des dargestellten realen Objekts keine Veränderung dieses Ausschnitts. Bei geraden Kanten sind Bewegungen senkrecht zum Helligkeitsgradienten nicht feststellbar, da nur ein bestimmter Bereich um einen Punkt betrachtet wird. Es ist offensichtlich, dass aufgrund dieses *Blendenproblems* nur signifikante Merkmale wie Ecken verfolgt werden können, bei denen Farb- bzw. Helligkeitsverläufe in mehreren Richtungen vorhanden sind. Zusätzlich kann ein *Korrespondenzproblem* auftreten, wenn bestimmte Muster in einem Bild mehrmals vorkommen und zwischen den Bildern die falschen Muster miteinander verknüpft werden [Jäh02, S. 399ff.], [SS00, S. 275ff.].

Bouguet [Bou00] beschreibt einen Algorithmus für eine Musterverfolgung, der in der Bildverarbeitungsbibliothek *OpenCV* enthalten ist und im Rahmen dieser Arbeit verwendet wird. Basis dafür ist der *Feature Tracker* von Kanade, Lucas und Tomasi [LK81], [ST94]. Dieser wird mit einem Pyramidenansatz erweitert, d.h. es erfolgt zunächst eine grobe Schätzung der Bewegung in einem geringer aufgelösten Bild, welche anschließend verfeinert wird. Ist die ungefähre Bewegung bekannt, kann die genauere Suche nach der Musterposition in einem eingeschränkten Bereich erfolgen, woraus sich ein Geschwindigkeitsvorteil ergibt.

Der Vorteil der Methode von Lucas und Kanade ist, dass spezielle Punkte für die Bewegungserkennung ausgewählt werden können. Uninteressante Bildbereiche bleiben unberücksichtigt, was zu einer schnelleren Berechnung führt. Andere, ebenfalls in der Bibliothek enthaltene Verfahren beziehen sich hingegen auf das Gesamtbild. Ein Blockmatching-Verfahren unterteilt das Bild in Bereiche gleicher Größe und errechnet Bewegungsvektoren für jeden Block. Der Algorithmus nach Horn und Schunck [HS81] ermöglicht eine Bewegungsanalyse für jeden Pixel. Diese Verfahren sind vor allem dann interessant, wenn die Kenntnis des optischen Flusses aller Bildbereiche benötigt wird. Eine ausführliche Übersicht sowie der Vergleich verschiedener Verfahren zur Berechnung von Bewegungen befindet sich in der Arbeit von Barron et al. [BFB94].

Kapitel 3

Das dip-Framework

3.1 Übersicht

Bevor mit Hilfe der in Kapitel 2 beschriebenen theoretischen Grundlagen aus Stereokamerabildern Tiefenbilder erzeugt und schließlich ausgewertet werden, wird in diesem Kapitel die Software vorgestellt, die den Rahmen für die verwendeten und entwickelten Algorithmen bildet.

Das *dip*[1]*-Framework* ist eine Software des DLR, die eine weiche Echtzeitverarbeitung von Bildsequenzen ermöglicht und für den mobilen Hubschrauber ARTIS konzipiert ist. Einsatzgebiet ist die Verwendung unterschiedlichster Bildverarbeitungsalgorithmen auf dem Bildverarbeitungsrechner des Helikopters. Das vorhergehende Testen und die Erforschung neuer Methoden am Boden ist ebenfalls möglich.

Konzeptioniert und entwickelt wurde die Software von Goormann [Goo04] und Guth [Gut04] am DLR. Dabei wird die objektorientierte Programmiersprache C++ verwendet. Vorteil z.B. gegenüber Java ist die hohe Hardwarenähe und dadurch höhere Abarbeitungsgeschwindigkeit – allerdings ist das kompilierte Programm nicht auf verschiedenen Betriebsystemen lauffähig. Dieses Manko kann allerdings dadurch ausgeglichen werden, dass lediglich Bibliotheken verwendet werden, die unter Linux und Windows verfügbar sind und dadurch eine Erstellung für diese beiden Systeme möglich ist.

Das Konzept ist dabei so flexibel, dass eine Erweiterung der Software mit neuen Bildverarbeitungsoperationen problemlos durchgeführt werden kann ohne vorhandene Komponenten zu beeinflussen.

[1] Digital Image Processing (digitale Bildverarbeitung)

3.2 Anwendungsumgebung

Das Framework selbst enthält alle wesentlichen Algorithmen zur Bildverarbeitung. Für den Anwender existieren zwei Softwareumgebungen, die das Framework verwenden. Die Interaktion erfolgt über eine grafische Oberfläche (GUI[2]). Beide Anwendungsumgebungen verwenden den gleichen Quellcode des Frameworks für die Datenverarbeitung und es ist dadurch sichergestellt, dass die Funktionalität der Algorithmen in den Umgebungen gleich ist. Abbildung 3.1 zeigt beide Umgebungen.

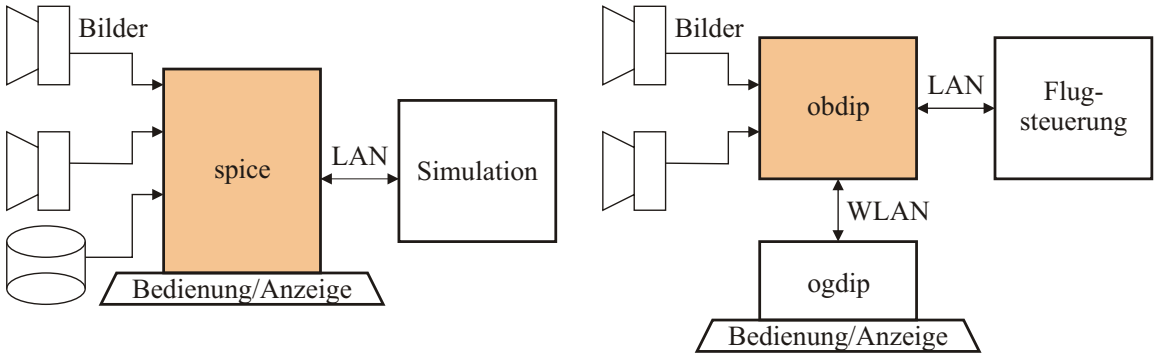


Abbildung 3.1: Das *dip-Framework* für eine Testumgebung (links) und den Flugeinsatz (rechts). Die orange gekennzeichneten Programme enthalten die gleichen Algorithmen zur Bildverarbeitung, Simulation und Flugsteuerung die gleichen Schnittstellen für den Datenaustausch. Das Programm für den Helikoptereinsatz wird vom Boden aus ferngesteuert.

Das Programm *spice*[3] ist die Testumgebung für sämtliche Bildverarbeitungsoperationen. Zusätzlich zu den vorgefertigten Operationen ist hier eine Verkettung von verschiedensten Bildanalysen möglich. Die Netzwerkkommunikation für die Fernsteuerung wird hier simuliert. Darüber hinaus können aufgezeichnete Bildsequenzen geladen werden.

Für den Flugeinsatz wird die zweite Umgebung verwendet, die aus den Programmen *obdip* (on-board dip) und *ogdip* (on-ground dip) besteht. Das *on-board dip* ist die Software, die auf dem Bildverarbeitungsrechner an Bord des Helikopters läuft. Die Bedienung erfolgt ausschließlich über Netzwerk, so dass keine Elemente zur Nutzerinteraktion benötigt werden. Theoretisch ließe sich dieser Teil als Konsolenanwendung implementieren. Allerdings wird es ermöglicht, zur Kontrolle und Aufzeichnung Bilder an die Bodenstation zu senden. Aufgrund der begrenzten Datenübertragungsgeschwindigkeit des Funknetzwerks erfolgt dies analog über eine Videofunkstrecke.

[2]Graphical User Interface: grafische Benutzerschnittstelle
[3]Small Program for Image Computing Experiments

Die Bildverarbeitung auf dem Helikopter wird vom *on-ground dip* ferngesteuert. Hier sind sämtliche Bedienelemente enthalten, die für die Bildverarbeitung während eines Fluges erforderlich sind. Im Wesentlichen können hier die Kameras gesteuert und vorgefertigte Bildanalysen aufgerufen werden.

Für den Test der Bildverarbeitung am Boden, d.h. ohne Helikopter, wird lediglich die *spice*-Komponente benötigt. Diese Arbeit bezieht sich auf diese Komponente, deren grafische Oberfläche ist in Abbildung 3.2 dargestellt.

Abbildung 3.2: Screenshot der *spice*-Bildverarbeitungsumgebung. Im oberen Bereich befindet sich links und rechts jeweils ein Fenster zur Bildanzeige. Dort ist die Darstellung beliebiger geladener und verarbeiteter Bilder möglich. Im unteren Bereich erfolgt die Steuerung der Bildverarbeitungsoperationen.

3.3 Funktionsweise

Die Funktion des Frameworks ist die Verarbeitung von Bildern. Ein *Bild* kann von einem oder mehreren *Filtern* sequentiell bearbeitet werden, wobei ein Filter zunächst beliebiger Art sein kann. Ausgabe ist wiederum mindestens ein Bild und es besteht die Möglichkeit, weitere Ausgaben wie bestimmte Bildeigenschaften zu erzeugen [Goo04, Kap. 5].

3.3.1 Kameras und Bilder

Im Zusammenhang mit dem dip-Framework wird das Wort *Kamera* als eine Bildquelle definiert, unabhängig davon, ob es sich um Bilder aus einer „echten“ Kamera oder um abgespielte Videosequenzen handelt.

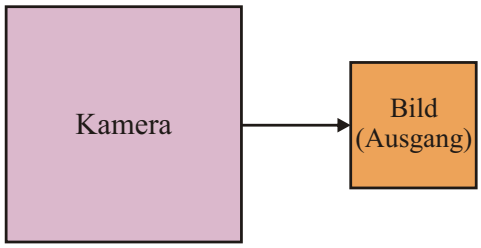


Abbildung 3.3: Kamera erzeugt Bild.

Die Software unterstützt das Laden von Bildern aus folgenden Quellen:

- Digitalkamera mit FireWire-Schnittstelle (IEEE 1394). An den Bus angeschlossene Kameras werden automatisch erkannt und sind nach dem Start des Programms sofort verfügbar, ohne dass sie gesondert geladen werden müssen.
- Bilddaten von einem Datenträger. Die Formate *.avi*[4] als einzelne Videodatei, sowie *.bmp*[5] als Folge von Einzelbildern mit durchnummerierten Dateinamen können ausgewählt werden. In beiden Fällen sollte es sich um verlustfrei komprimierte Bildformate handeln.

Eine Kamera stellt Bilder bereit und sorgt dafür, dass der Inhalt der Bilder entsprechend aktualisiert wird. Das Wort *Bild* bezeichnet im Gegensatz zu Kamera ein konkretes Bildobjekt, das von den Kameras erstellt und weiteren Verarbeitungsschritten verwendet wird.

Bei der Aktualisierung des Bildinhaltes, d.h. wenn ein neues Bild der Sequenz geladen wird oder eine echte Kamera ein neues Bild bereitstellt ändert sich die Referenz auf ein Bild nicht,

[4] Audio-Video-Interleave. Containerformat für Videos. Es unterstützt mehrere Bild- und Tonspuren und die Verwendung von Kompressionsverfahren.
[5] Bitmap. Bilddatenformat, welches für MS Windows und OS/2 entwickelt wurde. Lediglich die Kompression mittels Lauflängenkodierung (RLE) wird unterstützt.

d.h. aus interner Sicht findet keine Änderung des Bildes statt. Spätere Verarbeitungsoperationen benötigen somit keine Kenntnis über die Aktualisierung des Inhaltes. Späteres Abfragen der Bilddaten liefert automatisch den aktuellen Inhalt.

Allerdings ist es manchmal erforderlich, dass eine Information darüber benötigt wird, ob ein Bild tatsächlich aktualisiert wurde. Daher besitzt jedes Bildobjekt einen Zeitstempel, der die Systemzeit enthält, bei der die Kamera die letzte Aktualisierung des Bildinhalts vorgenommen hat. Werden zwei Bilder zu unterschiedlichen Zeiten abgefragt und miteinander verglichen, kann somit eine Aussage darüber getroffen werden, ob sich der Bildinhalt geändert hat, ohne den Inhalt selbst vergleichen zu müssen. Durch die errechenbare Zeitdifferenz lässt sich darüber hinaus feststellen, ob ein Bild der Sequenz übersprungen wurde oder nicht. Dies ist bei der Echtzeitverarbeitung und bei einigen Bildverarbeitungsoperationen von Bedeutung.

Einige Algorithmen benötigen zusätzliche Informationen über die Eigenschaften der Kamera, die das Bild aufgezeichnet hat. Daher erstellt jede Kamera ein Datenobjekt, welches an das ausgegebene Bild übergeben wird. Dieses Objekt enthält spezifische Eigenschaften der Kamera wie die innere Orientierung oder die Position am Hubschrauber. Ebenso stellt dieses Objekt Operationen zur Verfügung, die diese Parameter verwenden, wie beispielsweise die Umrechnung zwischen Bild- und Helikopterkoordinaten.

Nicht alle Bilder werden von Kameras erzeugt. Es ist möglich, Bilder zu kopieren oder Algorithmen auf die Bilder anzuwenden, wodurch weitere Bilder entstehen und die ursprünglichen Bilder erhalten bleiben. Sofern nicht anders angegeben, bekommen die Bilder den Zeitstempel und die Kamerainformationen des Ursprungsbildes – unabhängig von der Laufzeit des Algorithmus oder den Änderungen, die am Bild vorgenommen wurden.

3.3.2 Filter

Die Verarbeitung von Bildern übernehmen die *Filter*. Ein Filter bekommt ein Eingangsbild und wendet auf dieses einen Algorithmus an. Ergebnisse können sowohl als Bild, als auch in Form weiterer Informationen ausgegeben werden.

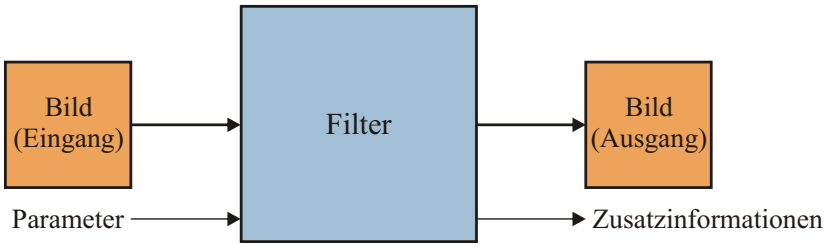


Abbildung 3.4: Filter verarbeitet Bild.

Wird ein Bild ausgegeben, ist es anderen Filtern möglich, dieses als Eingangsbild zu verwenden. Damit ist eine Hintereinanderausführung verschiedener Bildverarbeitungsalgorithmen möglich.

Über die grafische Oberfläche sind die von den Filtern verwendeten Parameter einstellbar, siehe Abbildung 3.2 auf Seite 34. Dies können Schieberegler, Checkboxen und weitere beliebige Eingabefelder sein. Ausgaben der Filter wie resultierende Bilder oder spezielle Bildeigenschaften werden ebenfalls dort angezeigt. Die Speicherung spezieller Daten in Dateien ist zusätzlich möglich.

Beispiele für typische Bildverarbeitungsfilter sind:

- Schwellwertoperatoren, z.B. zur Filterung bestimmter Helligkeiten oder Farben,
- Kantenerkennungsverfahren wie Sobel oder Canny-Filter,
- Hilfsoperatoren zum Kopieren, Skalieren oder Beschneiden von Bildern,
- komplexere Bildanalysen wie der Bewegungserkennung und der Suche nach Mustern.

3.3.3 Commander

Commander sind nutzerdefinierte Zusammenfassungen eines oder mehrerer Filter. Ein Commander beinhaltet die Verarbeitung eines Bildstroms – von der Kamera bis zum endgültigen Ausgangsbild. Es sind feste Commander verfügbar, welche auf den Flugeinsatz optimierte Bildverarbeitungsoperationen ausführen, wie das Verfolgen eines Musters oder das Aufzeichnen von Bildsequenzen.

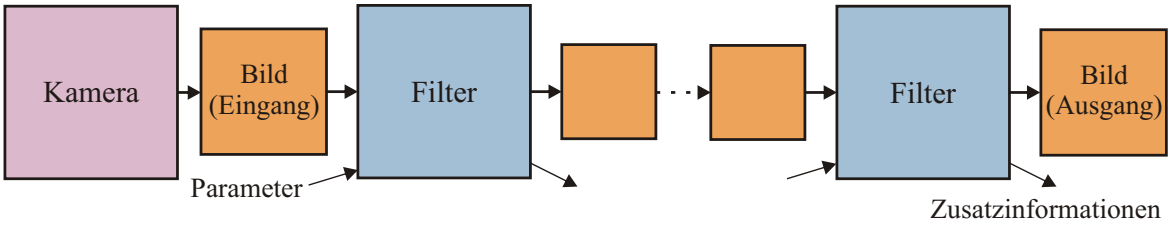


Abbildung 3.5: Commander enthalten mehrere Filter. Als Eingang kann direkt eine Kamera ausgewählt werden.

Der Vorteil dieser vorgegebenen Commander ist, dass die Parameter der darin verwendeten Filter auf den Einsatzzweck des Fluges optimiert sind und nicht bei jedem Programmstart eine umfangreiche Einstellung der Filtereinstellungen erforderlich ist.

Darüber hinaus gibt es den *Experimental* Commander. Dieser für Testzwecke vorgesehen und ermöglicht die Hintereinanderausführung von Filtern. Es können beliebige Filter ausgewählt

und miteinander verkettet werden, wobei der gesamte Funktionsumfang eines jeden Filters verfügbar ist.

3.4 Änderungen für die Auswertung von Stereobildern

Für die Verarbeitung von Stereobildern in dieser Arbeit waren neben der Erstellung von Filtern zur Bildverarbeitung einige Anpassungen der Software erforderlich. Ursprünglich ist dieses System nur dafür vorgesehen worden, eine einzige Kamera anzusteuern und deren Bilder zu verarbeiten. Somit war die Funktionalität auf die Verarbeitung einer Bildquelle beschränkt. Die wesentliche Änderung der Software im Rahmen dieser Arbeit ist die hinzugekommene Fähigkeit, mehrere Kameras gleichzeitig ansteuern zu können.

Die Ansteuerung verschiedener Bildquellen gewährleistet allerdings noch nicht, dass die Bilder zueinander passend sind. Werden beispielsweise zwei Filmsequenzen geladen, erfolgt das Abspielen zunächst unabhängig voneinander. Davon ausgehend, dass eine Multibildsequenz aus mehreren Einzelbildsequenzen besteht, die synchron aufgezeichnet sind, muss beim Abspielen, bzw. weiterem Verarbeiten, dafür gesorgt werden, dass die aktuellen Bilder der einzelnen Sequenzen auch zueinander gehören.

Dazu wird eine Sequenz als *Master* definiert. Sie gibt mit Hilfe des Bildzeitstempels des aktuellen Bildes vor, welche Bilder andere Kameras laden sollen. Ist eine Kamera mit dem Master synchronisiert, so wird bei einer Aktualisierung das Bild geladen, dessen Zeitstempel der vorgegebenen Zeit am nächsten kommt. So wird zwar gewährleistet, dass ein möglichst zeitgleiches Bild gefunden wird, die Zeiten müssen allerdings nicht exakt übereinstimmen. Dieser Fehler beträgt maximal eine halbe Bilddauer. Bei einer Bildwiederholrate von 25 Hz sind dies 20 ms. Eine höhere Genauigkeit ist bei der Synchronisation aufgezeichneter Bildsequenzen nicht möglich. Insbesondere bei der Aufzeichnung schneller Bewegungen liefert die Stereobildauswertung dann falsche Ergebnisse.

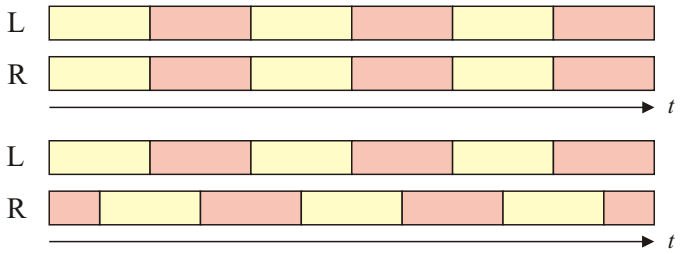


Abbildung 3.6: Synchronisation von zwei Bildsequenzen, die aufeinander folgenden Bilder sind hier abwechselnd gelb bzw. rot dargestellt. Im Idealfall (oben) stimmen die Zeiten der Bilder exakt überein, im schlechtesten Fall (unten) existiert eine Abweichung von einer halben Bilddauer, dieser Fehler kann durch Laden anderer Bilder nicht ausgeglichen werden. Bei höheren Zeitdifferenzen werden Bilder nicht aktualisiert oder übersprungen.

Werden keine aufgezeichneten Bildsequenzen, sondern reale Kameras verwendet, kann durch hardwareseitige Synchronisation eine wesentlich höhere Genauigkeit erzielt werden. Dabei wird nicht wie bei aufgezeichneten Sequenzen ein passendes Bild, z.B. aus einem Puffer, ausgewählt, sondern es erfolgt eine Abstimmung der Sensorbelichtung. Die einzelnen Bilder der verwendeten Stereokamera besitzen nach Angaben des Herstellers einen Zeitverzug von etwa 60 μs [Vid04, S. 19], eigenen Untersuchungen zufolge ist allerdings mit Unterschieden von bis zu 100 μs zu rechnen – diese Zeitdifferenzen gelten dann auch bei geladenen Bildsequenzen, die mit dieser Stereokamera aufgezeichnet wurden. Die Synchronisation erfolgt automatisch über den FireWire-Bus.

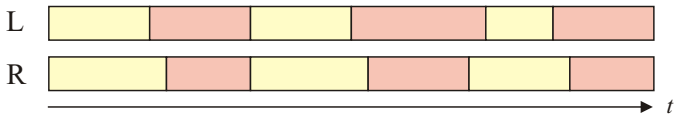


Abbildung 3.7: Hardwareseitige Synchronisation von zwei Kameras. Die Zeiten der aufgenommenen Bilder stimmen nicht exakt überein (Verzögerung), die Bildzeiten können leicht variieren (Jitter). Eine maximale Verzögerung zwischen den beiden Kameras wird allerdings gewährleistet, sie ist wesentlich geringer als die des oben beschriebenen schlechtesten Falls bei aufgezeichneten Bildsequenzen.

3.5 Verwendete Bibliotheken

Das Framework verwendet folgende externe Bibliotheken:

DC1394: Der DC1394-Treiber ist eine Schnittstelle für Linux zum Ansteuern von Kameras, die an den IEEE 1394 (FireWire) Bus angeschlossen sind. Die Schnittstelle stellt verschiedene Funktionen zur Bedienung der Kameras bereit, wie z.B. Start/Stop, Belichtung, Bildwiederholrate, Auflösung, uvm. [DC1394].

FLTK: Das Fast Light Toolkit ist eine Bibliothek zur Erstellung von grafischen Oberflächen. Im Gegensatz zu Bibliotheken wie z.B. MFC[6] für Windows existieren Versionen für verschiedene Betriebssysteme mit gleichen Schnittstellen. [FLTK].

OpenCV: Die Open Source Computer Vision Library ist eine quelltextoffene Schnittstelle von Intel für die digitale Bildverarbeitung. Neben grundlegenden Funktionen wie Laden und Speichern von Bildern und -Sequenzen verschiedenster Formate und dem Erstellen von Zeichenelementen sind verschiedene Algorithmen der Bildanalyse wie z.B. Kantenerkennung, optische Flussberechnung und auch Tiefenberechnung von Stereobildern enthalten [OpenCV].

[6] Microsoft Foundation Classes. Objektorientierte Programmierschnittstelle für die Erstellung von C++-Applikationen unter Windows, u.a. zur Erzeugung von Benutzeroberflächen.

SVS: Das Small Vision System ist eine kommerzielle Software zur Ansteuerung von Kameras, optimiert auf die verwendeten Stereokameras von Videre Design. Neben Funktionen zum Laden von Bildern existiert eine umfangreiche Schnittstelle zur Kalibrierung der Kameras und zur Entzerrung und Rektifizierung der Bilder. Des weiteren ist eine Programmierschnittstelle zur Gewinnung von Tiefeninformationen aus Stereobildern enthalten [SVS].

Kapitel 4

Erzeugung von Tiefenbildern

4.1 Tiefenschätzung

Die Berechnung der Tiefeninformation basiert auf dem Finden von Bildsegmenten, die sich in beiden Bildern befinden und zu demselben aufgezeichneten 3D-Objekt gehören. Anders ausgedrückt, ist die Tiefenschätzung die Unterteilung eines Bildes in Muster und die Suche dieser Muster im jeweils anderen Bild.

4.1.1 Finden von Korrespondenzen

In Abbildung 4.1 ist die Aufzeichnung von zwei Bildern in standard-stereoskopischer Anordnung der Kameras dargestellt. Ist ein Objektpunkt im Blickfeld beider Kameras, so existiert auf jedem Bild ein entsprechender Bildpunkt. In Abschnitt 2.2.3 ab Seite 21 wurde dieses Modell erläutert. Da beide Bildpunkte die gleiche y-Koordinate besitzen, müssen die entsprechenden diskreten Pixel auf der gleichen Zeile liegen. Unterschiede existieren nur noch in einer Dimension. Legt man beide Bildebenen übereinander, ergibt sich ein horizontaler Abstand zwischen den Punkten, genannt *Disparität*.

Aufgabe der Tiefenbildberechnung ist es, für Punkte eines Bildes die korrespondierenden Punkte des anderen Bildes zu finden bzw. diesen Abstand zu den entsprechenden Punkten zu messen. Im Idealfall lassen sich für jeden Punkt eines Bildes zugehörige Punkte im anderen Bild in einem gewissen Abstand finden. Ergebnis ist eine zweidimensionale Funktion $d(x, y)$, welche die Disparität zu jedem Bildpunkt eines der Ausgangsbilder liefert. Wird das linke Bild als Referenzbild gesetzt und nach ähnlichen Mustern im rechten Bild gesucht, sind die Disparitäten positiv[1] – umgekehrt entsprechend negativ.

[1] inklusive Null

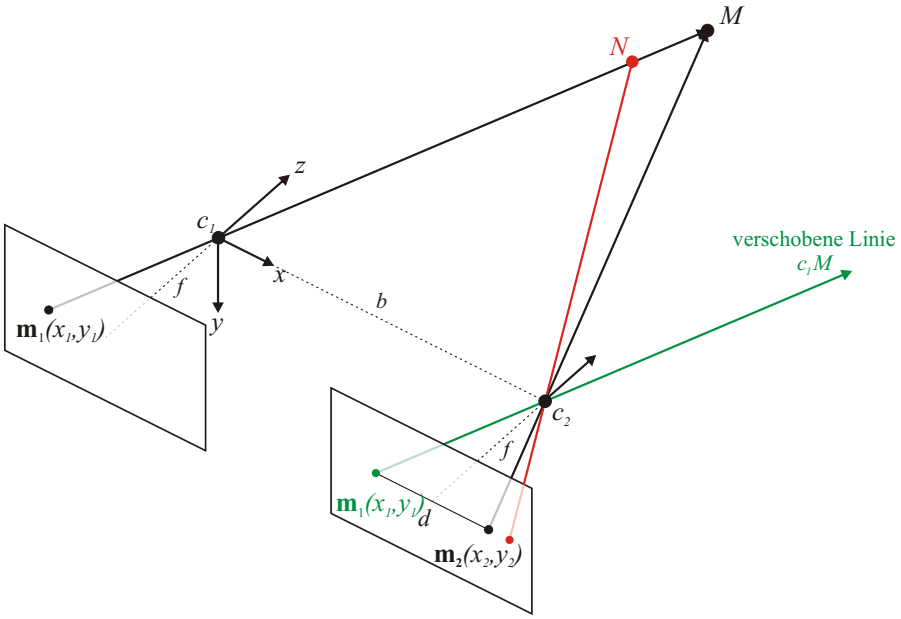


Abbildung 4.1: Stereoskopische Sicht auf zwei ausgewählte Punkte. Während die Punkte M und N in der linken Kamera auf denselben Bildpunkt abgebildet werden und dadurch mit dieser Kamera nicht voneinander unterscheidbar sind, ist durch Zuhilfenahme einer zweiten Kamera mit Abstand b eine eindeutige Zuordnung des Punktes möglich. Die rechte Kamera bildet beide Punkte auf verschiedene Bildkoordinaten ab. Die Disparität d ist der horizontale Abstand beider Bildpunkte und ermöglicht eine Zuordnung vom Bildpunkt nach M oder N.

Im ersteren Fall ist $d(x, y) = d(x_1, y_1)$, zwischen zwei korrespondierenden Bildpunkten $\mathbf{m}_1$ (x_1, y_1) und $\mathbf{m}_2$ (x_2, y_2) gilt dann die Beziehung

$$x_2 = x_1 + d(x_1, y_1) \quad \text{und} \quad y_2 = y_1. \tag{4.1}$$

Alternativ lässt sich auch eine weitere Sicht, beispielsweise zwischen beiden Ausgangsbildern, generieren [SS01].

Wird dieser Abstand d in einem Grauwert g kodiert und für jeden Punkt des Ausgangsbildes der Grauwert an entsprechender Stelle gesetzt, entsteht das Disparitätenbild $g(x, y)$ – eine visuelle Darstellung der Tiefeninformation[2].

Ist die Disparität $d(x, y)$ eines Bildpunktes (x, y) bekannt, so lässt sich mit Hilfe des Strahlensatzes die Tiefe Z errechnen. Es ist

$$Z = \frac{b \cdot f}{d}, \tag{4.2}$$

mit dem Basisabstand b und der Brennweite f [KB05, S. 32]. Damit sind größere Entfernungen messbar, sofern Kameraobjektive mit höheren Brennweiten oder ein größerer Abstand

[2]Die Disparität ist ein reziprokes Maß für die Tiefe bzw. Entfernung, die Begriffe Disparitäten- und Tiefenbild werden hier synonym verwendet.

zwischen beiden Kameras verwendet wird. Soll nur ein bestimmter Entfernungsbereich abgedeckt werden, weil beispielsweise Hindernisse in anderen Entfernungen auszuschließen sind, bietet es sich an, den Suchbereich für die Disparitätenschätzung zu begrenzen.

Aus der Umkehrung der Kollinearitätsgleichungen 2.6 und 2.7 (Seite 9) ergibt sich die Beziehung zwischen einem Bildpunkt (x, y) und dem Objektpunkt (X, Y, Z) im Kamerakoordinatensystem:

$$X = Z \cdot \frac{(x - x_0)}{f} \quad \text{, bzw.} \quad Y = Z \cdot \frac{(y - y_0)}{f}, \tag{4.3}$$

mit dem Bildhauptpunkt (x_0, y_0) und der Brennweite[3] f. Mit Hilfe von Gleichung 4.2 ist so aus einem Punkt des Disparitätenbildes der Objektpunkt rekonstruierbar.

4.1.2 Vorstellung einiger Verfahren

Für Punkte eines Referenzbildes werden die korrespondierenden Punkte im jeweils anderen Bild gesucht. Innerhalb eines bestimmten Bereiches wird mit Hilfe von Abstandsmaßen (siehe Abschnitt 2.4.2, Seite 27) nach ähnlichen Punkten gesucht, wobei die Umgebungen der jeweiligen Punkte berücksichtigt werden. Als korrespondierender Punkt wird der ähnlichste Punkt ausgewählt, aus dessen Position ergibt sich die Disparität. Größere Vergleichsfenster reduzieren das Rauschen im Disparitätenbild auf Kosten der Möglichkeit, kleine Objekte zu finden.

Neben merkmalsbasierten Verfahren wie der von Taylor [Tay03] vorgestellten Methode, einzelne Features auszuwählen und aus deren Korrespondenz die Oberflächen der Objekte zu rekonstruieren, werden vor allem Algorithmen präsentiert, die eine *dichte* Disparitätenkarte erzeugen. Dabei wird versucht, für möglichst jeden Punkt des Referenzbildes die Tiefe zu ermitteln. Eine umfassende Übersicht liefert die Arbeit von Scharstein und Szeliski [SS01].

Neben dem Finden von Disparitäten ist die Optimierung der resultierenden Tiefenkarte von großer Bedeutung. Während sich *lokale* Verfahren darauf beschränken, das Tiefenbild zu glätten, um das Rauschen weiter zu minimieren, erfolgt bei *globalen* Ansätzen eine Bewertung des Gesamtbildes. Dazu wird eine Kostenfunktion erzeugt, welche die in einem Disparitätenbild gespeicherten Unterschiede beider Einzelbilder mit den tatsächlichen Unterschieden für jeden Bildpunkt vergleicht und aufsummiert. Eine weitere Kostenfunktion bewertet die Qualität des Disparitätenbildes hinsichtlich gleichmäßiger Darstellung von Flächen und der Wiedergabe von *Diskontinuitäten* bei Objektkanten. Ziel ist es, die Summe beider Kostenfunktionen zu minimieren – als Kompromiss zwischen exakter Wiedergabe der Korrespondenzen und geringem Rauschen im Tiefenbild [SS01, S. 8f.]. Beispiele dafür sind die Algorithmen von Fua [Fua91] oder Scharstein und Szeliski [SS99].

[3] Die Brennweite zeigt hier in positive z-Richtung, bei den Kollinearitätsgleichungen wird von der entgegengesetzten Richtung ausgegangen [Luh00, S. 118].

Darüber hinaus stellen beispielsweise Birchfield und Tomasi [BT96], sowie Forstmann et al. [For+04] eine Optimierungsmethode mit Hilfe dynamischer Programmierung vor. Wie bei den anderen Verfahren ist der Grundgedanke der, dass sich Disparitäten benachbarter Bildpunkte normalerweise nur geringfügig unterscheiden. Ist die Disparität eines Bildpunktes bekannt, wird davon ausgegangen, dass bei der Darstellung von Flächen die Disparität eines benachbarten Punktes ähnlich ist. Dadurch kann der Suchbereich einschränkt werden und lässt eine zeilenweise Kostenminimierung zu. Die zeilenweise Optimierung verursacht allerdings Fehler in Form von horizontalen Streifen. Nach dem Vergleich in [SS01, S. 45ff.] sei dieses Verfahren allerdings in etwa so schnell wie eine einfache Glättung und damit wesentlich effizienter als eine globale Optimierung.

In dieser Arbeit wird die kommerzielle Software *Small Vision System* (SVS) von Konolige und Beymer [KB05], [SVS] verwendet. Sie ist auf Geschwindigkeit optimiert und verwendet keine globale Optimierung. Als Abstandsmaß für den Vergleich von Bildabschnitten wird nach Hrabar [Hra06, S. 75] die Summe absoluter Differenzen verwendet.

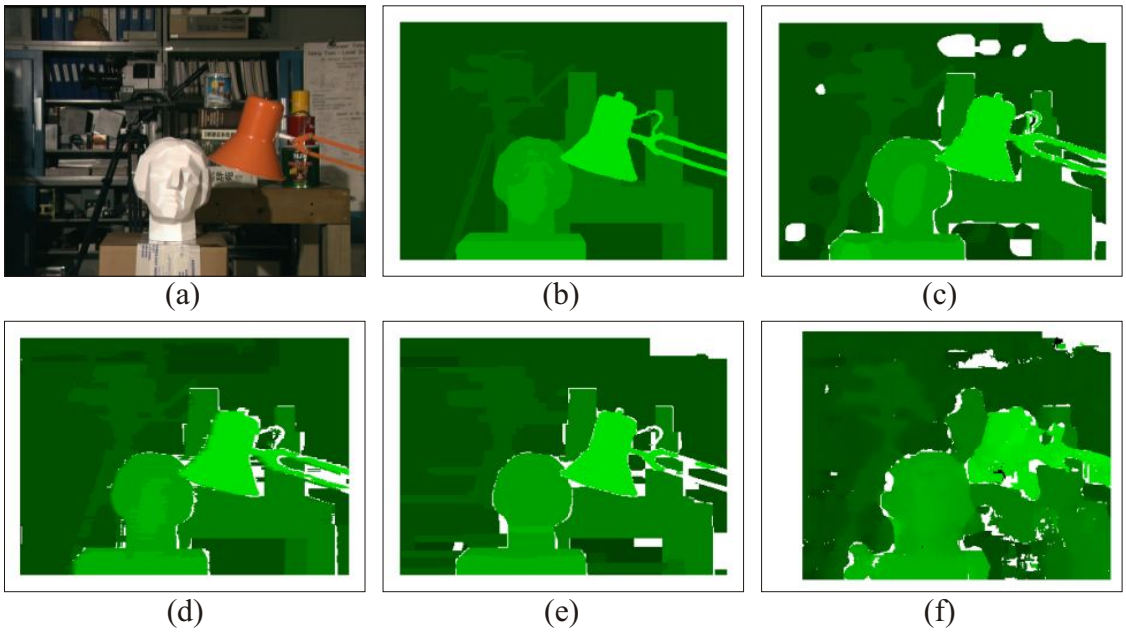


Abbildung 4.2: Linkes Originalbild (a), reale Tiefen (b), die Tiefenbildberechnungen mit den Algorithmen von Scharstein/Szeliski [SS99] (c), Forstmann [For+04] (d), Birchfield [BT96] (e). Zum Vergleich: die verwendete SVS Stereo Engine [SVS] (f).

Abbildung 4.2 zeigt Tiefenbilder unter Verwendung verschiedener Algorithmen. Das verwendete Stereobildpaar und die Bilder (b) bis (e) stammen von der Website von Scharstein und Szeliski [SS03]. Das Vergleichsbild (f) ist mit dem *Small Vision System* erstellt. Zur besseren Veranschaulichung sind diese und alle weiteren in dieser Arbeit vorkommenden Tiefenbilder so eingefärbt, dass die Farbskala zwischen grün und schwarz die Disparität darstellt, wo-

bei hellere Farbtöne größere Disparitäten und damit geringere Distanzen bedeuten. Fehlende Tiefeninformationen werden durch weiße Bildpunkte repräsentiert.

Es hängt stark von den Umgebungsbedingungen und dem Bildinhalt ab, inwieweit die Suche nach ähnlichen Bildabschnitten erfolgreich ist und vor allem korrekte Ergebnisse liefert. Unabhängig von der aufgezeichneten Umgebung existiert Bildrauschen und ein möglicher Unterschied bei der Helligkeit beider Bilder durch verschiedene Belichtungen. Dies reduziert die Ähnlichkeit derjenigen Bildpunkte, die das gleiche darstellen und deren Korrespondenz erkannt werden soll. Darüber hinaus ist es am einfachsten, starke Grauwertänderungen, also Kanten, wiederzufinden. Beim verwendeten Algorithmus werden die Ausgangsbilder daher mit einem *Laplacian of Gaussian*-Filter nach Marr und Hildreth [MH80] vorverarbeitet, es verstärkt Kanten, verringert Rauschen und liefert durch die Annäherung an die 2. Ableitung der Grauwerte keine absoluten Helligkeiten, sondern die Stärke der Helligkeitsänderungen.

Um die Genauigkeit bei der Berechnung der Disparität zu erhöhen, kann entweder mit höherer Bildauflösung aufgezeichnet oder eine Interpolation bei der Schätzung der Tiefe angewendet werden [Wil98, S. 61]. Aufgrund der Unschärfe der Bilder durch die Aufzeichnung und Vorverarbeitung ist die Ähnlichkeit in Abhängigkeit von der Verschiebung meist eine weiche Funktion ohne allzu harte Sprünge. Mittels Interpolation können Disparitäten mit einer Genauigkeit auf Subpixelebene geschätzt werden.

Die Interpolation kann prinzipiell mit beliebigen Funktionen erfolgen, die ein lokales Extremum besitzen. Als einfachste Funktion bietet sich eine Parabel an, die mit Hilfe der kleinsten Fehlerquadrate als kontinuierliche Schätzung von beispielsweise 3 oder 5 Punkten um das vorher gefundene Maximum der Ähnlichkeiten verwendet wird. Der Scheitelpunkt der Parabel liefert dann einen genaueren Wert für die beste Übereinstimmung zweier Bildabschnitte. Die verwendete svs-Bibliothek gibt eine Genauigkeit von $\frac{1}{16}$ Bildpunkten bei der Disparitätenschätzung an [KB05].

4.2 Fehlerkorrektur

Idealerweise existiert für jeden Bereich des linken Bildes ein ähnlicher Bereich im rechten Bild. Es existieren allerdings Situationen, in denen keine Region gefunden werden kann, die als ähnlich zu betrachten ist. Ebenso kann es vorkommen, dass bestimmte Bereiche falsch klassifiziert wurden, d.h. es werden Disparitäten errechnet, wo in in der Realität keine Objekte in entsprechender Entfernung zu finden sind.

Im allgemeinen markiert eine Fehlerkorrektur solche Punkte des Disparitätenbildes als ungültig, in Abbildung 4.2 (Seite 44) sind dies die weiß dargestellten Bereiche. Einige Algorithmen wie der von Birchfield und Tomasi [BT96] ermöglichen auch die nachträgliche Korrektur dieser Disparitäten, der Wert wird aus den Umgebungswerten ermittelt. Dadurch ist das

resultierende Tiefenbild weniger lückenhaft. Blaschek [Bla04] nennt ebenfalls eine Methode zur Füllung der Lücken.

4.2.1 Erkennung und Beseitigung von Unsicherheiten

Eine Ursache für Fehlstellen im Tiefenbild ist fehlende Textur – fehlende Kanten, die als Vergleichsmerkmal dienen können. Bei ebenen Flächen ist dies der Fall. Hierbei korrespondieren verschiedene Regionen gleich stark, so dass eine eindeutige Schätzung der Disparität nicht möglich ist. Ähnliches gilt für sich wiederholende Muster, nur dass hier nicht alle Vergleichsregionen gut mit dem Ausgangsmuster korrespondieren, sondern nur diejenigen, in denen das sich wiederholende Muster vorkommt.

Weiterhin ist es möglich, dass einige Bildabschnitte von einer Kamera erfasst werden, aus der Perspektive der anderen jedoch verdeckt sind (Okklusion). Gehört ein gesuchtes Muster zu einer solchen Region, wird es beim Durchsuchen des anderen Bildes nicht wiedergefunden.

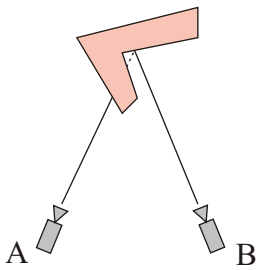


Abbildung 4.3: Beispiel für eine Okklusion. Der Punkt, auf den Kamera B zeigt, ist von Kamera A aus nicht sichtbar, da die Kameras das Bild aus einer unterschiedlichen Perspektive aufzeichnen. Solche Verdeckungen treten vor allem bei Objektkanten auf.

Fehler, die aufgrund von fehlender Textur entstanden sind, zeichnen sich dadurch aus, dass bei verschiedenen Disparitäten, also Verschiebungen des Vergleichsfensters, Gemeinsamkeiten gefunden werden. Es existiert kein Wert, bei dem sich die Ähnlichkeit der Regionen stark von den anderen Werten abhebt. Ein solches Matching ist nur sehr unsicher. Es lässt sich ein Wert für die Qualität eines Matchings festlegen, der umso höher ist, je stärker die Ähnlichkeit der in Wirklichkeit korrespondierenden Regionen im Vergleich zu den anderen Regionen ist. Unterschreitet die Qualität einen bestimmten Schwellwert, so wird die hier geschätzte Disparität nicht akzeptiert.

Erkannt werden können diese und auch die durch Okklusion und Wiederholungen auftretenden Fehler zusätzlich durch den links/rechts-Vergleich. Ein Abschnitt des linken Bildes wird mit den Abschnitten der gleichen Zeile des rechten Bildes vergleichen und so eine Disparität geschätzt. Ergibt der umgekehrte Vergleich zwischen einer Region des rechten Bildes mit einer Zeilenabarbeitung des linken Bildes den gleichen Wert, kann davon ausgegangen werden, dass keine falsche Schätzung aufgrund von Okklusionen erfolgte. Durch Wiederholungen oder

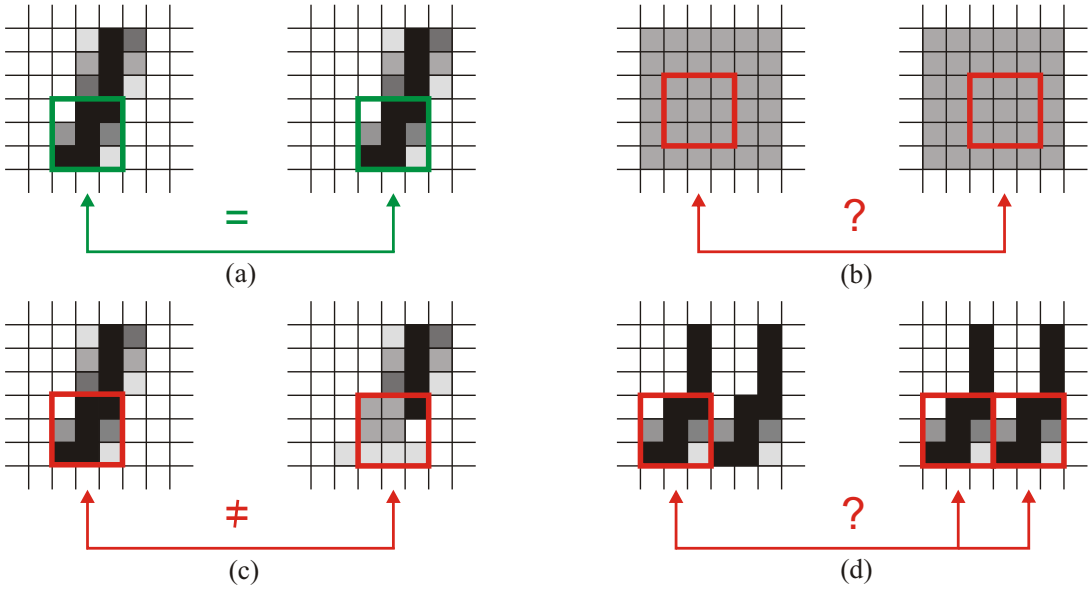


Abbildung 4.4: Verschiedene Beispiele von Matchings. Im Idealfall wird das gesuchte Muster des (hier: jeweils linken) Referenzbildes im anderen Bild wiedergefunden. Die horizontale Verschiebung dieses Musters ist die geschätzte Disparität für den Pixel in der Mitte des Vergleichsfensters (a). Bei gleichmäßigen Flächen ist eine eindeutige Zuordnung nicht möglich (b). Okklusionen sind verantwortlich dafür, dass ein gesuchtes Muster nicht wiedergefunden werden kann (c). Bei sich wiederholenden Mustern ist ebenfalls keine eindeutige Zuordnung und damit keine „sichere“ Disparitätenschätzung möglich (d).

fehlende Textur können ebenso unterschiedliche Werte beim Vergleich zwischen links und rechts bzw. umgekehrt entstehen. Die gefundene Disparität in beiden Bildern ist damit nur im korrekten Fall einzigartig. Allerdings ist es sinnvoll, einen Toleranzbereich einzuführen, wie ihn Blaschek [Bla04, S. 57] vorschlägt. Dies bedeutet, dass die gefundenen Disparitäten nicht exakt (vom Betrag her) gleich sein müssen, deren Differenz jedoch nicht zu hoch werden darf. Andernfalls werden die Disparitäten dieser Bildpunkte ungültig.

Die verwendete Software unterscheidet zwischen *Konfidenz-* und *Einzigartigkeitsfilterung* zur Beseitigung von Disparitäten, die durch eine nur schwach signifikante Punktkorrespondenz bzw. Wiederholungen fehlerhaft sein können [KB05, S. 27].

4.2.2 Beseitigung kleiner Regionen

Obwohl durch die Beseitigung unsicherer Matchings viele Fehlinterpretationen ausgeschlossen werden, zeigt sich in der Praxis, dass dennoch einige Fehlstellen im Disparitätenbild vorhanden sind. Oft sind dies kleine Regionen, die eine hohe Disparität aufweisen und stark von benachbarten Bereichen abweichen. Es ist anzumerken, dass bei einer Filterung bestimmter Regionen stets die Gefahr besteht, sehr kleine reale Objekte fälschlicherweise zu entfernen, was für die Kollisionsvermeidung durchaus ein Problem darstellen kann.

Zur Beseitigung solcher „Zacken“ schlägt Blaschek [Bla04, S. 96ff.] eine Segmentierung des Bildes vor. Im Disparitätenbild werden zusammenhängende Pixel mit gleichem Grauwert zu Segmenten zusammengefasst, allzu kleine Segmente werden entfernt. In [ML98, Abschnitt 4.2] ist die gleiche Idee präsent, allerdings sind geringe Intensitätsunterschiede durchaus erlaubt. Birchfield und Tomasi [BT96] filtern Spalten und Zeilen des Tiefenbildes separat. Fehler seien vor allem dadurch erkennbar, dass sie besonders stark von der Umgebung abweichen – während reale Objekte nur leichte Grauwertverläufe enthalten können. Neben der Beseitigung von kleinen, störenden Elementen könne ebenso eine Filterung dünner Strukturen, die Teil von größeren Bildsegmenten sind, vermieden werden.

Die hier vorgestellte Fehlerbehebung basiert auf der Segmentierung von Blaschek, allerdings muss bei benachbarten Bildpunkten keine Gleichheit vorliegen, damit sie dem gleichen Segment zugeordnet werden. Es genügt eine gewisse „Ähnlichkeit“, d.h. wenn der Grauwertunterschied zwischen einem Pixel und einem Nachbarn einen Schwellwert unterschreitet. Dabei wird das Bild zeilenweise abgetastet und jeder Bildpunkt mit genau den Nachbarn verglichen, die schon abgetastet wurden.

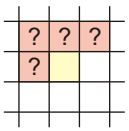


Abbildung 4.5: Untersuchung derjenigen Nachbarn, bei denen schon eine Überprüfung der Segmente stattfand. Bei einer zeilenweisen Abarbeitung des Bildes sind das in einer 8er-Nachbarschaft diejenigen Pixel, die oben-links, oben, oben-rechts und links vom aktuellen Bildpunkt liegen.

Jeder Bildpunkt erhält zunächst einen Verweis auf ein Segment, der zu Beginn leer ist. Bei Abtastung wird dieser Zeiger auf ein neues Segment der Größe 1 gesetzt. Trifft dieser Pixel nun auf einen ähnlichen Bildpunkt eines anderen Segments innerhalb der Nachbarschaft, so wird die Regionengröße um die des aktuellen Punktes erhöht und dessen Zeiger entsprechend umgesetzt. Die von vorangegangenen Filterungen schon als Fehler klassifizierte Pixel werden dabei ausgeschlossen. Sind alle Bildpunkte abgetastet, werden deren Zeiger auf die entsprechenden Segmente überprüft. Ist die Größe des Segments zu klein, auf die ein Pixel verweist, so wird dieser und damit jeder Punkt dieses Segments als ungültig markiert.

Ein alternatives Verfahren, das von Jennings [Jen96] beschriebene, abgewandelte Medianfilter, hat sich als nicht praktikabel erwiesen. Dort wird jeder Pixel in einem 3×3-Fenster mit den Nachbarn verglichen. Bei hinreichend vielen Pixeln (in diesem Fall 40 %), die zu stark von der Mitte abweichen, wird diese als ungültig klassifiziert. Typisches Rauschen lässt sich damit unterdrücken, fehlerhafte Regionen ab einer gewissen Größe jedoch nur durch Vergrößerung des Vergleichsfensters. Korrekte, aber dünne Linien würden dann ebenfalls unterdrückt werden, was jedoch nicht gewünscht ist.

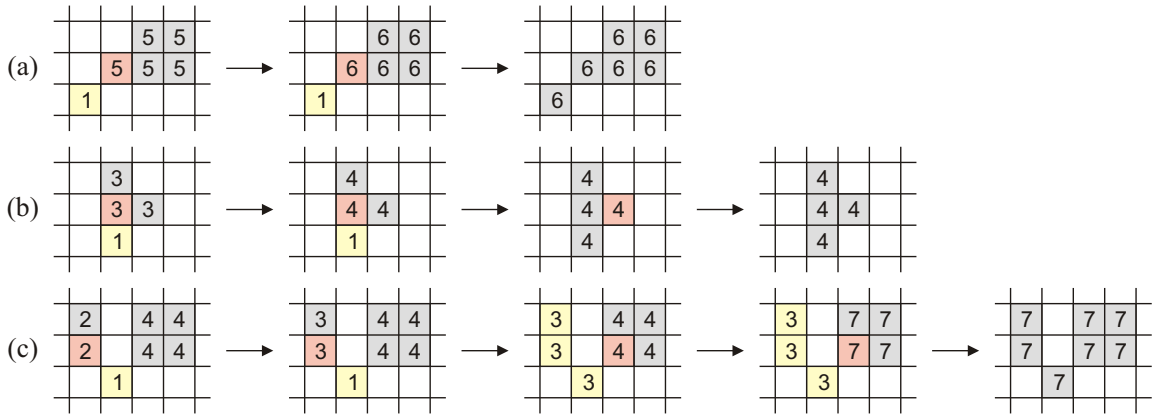


Abbildung 4.6: Berechnung von Segmentgrößen. Zunächst erhält ein getesteter Pixel (gelb) die Segmentgröße 1. Existiert kein Pixel mit ähnlichem Grauwert in der Umgebung, wird der Pixel einer neuen Region zugeordnet und diese Größe zugewiesen. Wird ein passender Pixel (rot) gefunden, so wird die Regionengröße des Pixels zu der passenden addiert und der Pixel dieser Region zugeteilt (a). Es wird vermieden, dass die Pixelgröße mehrmals addiert wird, indem Pixel nach erfolgreichem Zuordnen in der gleichen Region liegen und ein weiterer Vergleich mit Pixeln dieser Region in der Umgebung nicht stattfindet (b). Wird allerdings eine andere Region gefunden, so werden diese zusammengefügt (c).

4.3 Himmelserkennung

Mittels Stereobildvergleich ist es aufgrund der Eigenschaften des Himmels kaum möglich, ihm die Information „weit entfernt“ zuzuordnen. Es ist eher zu erwarten, dass aufgrund der recht ebenen Fläche dort keine Disparitäten geschätzt werden können. Spätere Versuche zeigen dies. Allerdings ist die Information, ob sich in einer bestimmten Bildregion der Himmel befindet, während eines Fluges unabdingbar, um diese Abschnitte als ungefährlich und damit als mögliches Flugziel zu klassifizieren.

Auch wenn der Himmel verschiedenste Farben und wetterbedingt auch Formen annimmt, kann folgende Klassifikation in vielen Fällen den Himmel von anderen Objekten wie dem Boden und sonstigen Hindernissen unterscheiden. Einerseits ist eine hohe Helligkeit im Vergleich zu anderen Objekten vorhanden, andererseits keine scharfen Kanten. Mit einer Farbkamera kann das Vorkommen der normalerweise blauen Färbung mit einbezogen werden. Diese steht allerdings nicht zur Verfügung, eine farbbasierte Himmelserkennung entfällt daher.

Hier wird ein einfaches und vor allem sehr schnelles Verfahren nach Cornall und Egan [CE04] zur Segmentierung eines Bildes in zwei Klassen verwendet. Zwei Bedingungen müssen dafür erfüllt sein: hinreichend hohe Helligkeit und wenig Textur.

Eine gewisse Helligkeit ist genau dann vorhanden, wenn ein bestimmter Grauwert überschritten wird. Das vorgestellte Verfahren verwendet den Schwellwert nach Otsu. Es wird aus dem Bildhistogramm derjenige Schwellwert errechnet, der das Bild möglichst gut in die beiden

Klassen *hell* und *dunkel* unterteilt. Zu Testzwecken wird hier allerdings ein fester Wert verwendet, der vom Benutzer verstellt werden kann.

Die Texturierung erfolgt hier nur durch Vergleich mit dem jeweils vorigen Bildpunkt der gleichen Zeile. Weicht dieser zu stark vom untersuchten Punkt ab, gilt dieser als texturiert. Dies ist bei Kanten der Fall. Andernfalls wird ist die zweite Bedingung für die Himmelsklassifikation erfüllt.

In der Anwendung wird aus dem linken Ausgangsbild ein Binärbild errechnet, welches Aufschluss darüber gibt, ob ein Punkt dieses Bildes zum Himmel gehört oder nicht. Dieses wird mit dem Disparitätenbild verrechnet. Ist für einen Punkt kein gültiger Disparitätswert vorhanden, wird dieser Wert gleich null gesetzt, sofern es sich um einen Punkt des Himmels handelt. Nach Gleichung 4.2 entspricht dies der unendlichen Entfernung. Punkte des Himmels, denen eine Disparität zugeordnet werden kann, bleiben unberücksichtigt.

Abbildung 4.7: Segmentierung eines Bildes (a) in Himmel und Erde. Bildpunkte, die dem Himmel zugeordnet werden, sind weiß dargestellt (b).

Cornall und Egan stellen eine weitere Methode zur Detektion des Himmels vor. Das *k-means Clustering* unterteilt das Bild in k zusammenhängende Bereiche, die möglichst homogen sind und sich von den anderen Bereichen stark unterscheiden. In Anwendungen wie der Horizonterkennung unter normalen Bedingungen liefert das Verfahren schon bei $k = 2$ bessere Ergebnisse als die Verwendung von Schwellwert und Texturierung. Allerdings ist es rechenintensiver und liefert unter besonderen Bedingungen, in denen sich das Bild nicht einfach in Himmel und Erde unterteilen lässt, nur bei höheren Werten für k akzeptable Ergebnisse [CE05].

Darüber hinaus wird von Nechyba et al. [Nec+05] die Segmentierung mit Hilfe von *Diskriminanzanalyse* und *Bayesschen Netzen* präsentiert, die auch unter komplizierten Bedingungen wie Bewölkung oder verschiedenen Farben bei Dämmerung funktionieren soll. Die Rechenzeit ist allerdings für eine Zusatzfunktion der Stereobildanalyse ebenfalls zu hoch.

4.4 Implementation eines Stereofilters

Als Voraussetzung für eine Kollisionsvermeidung wird ein Filter benötigt, das aus einem Stereobild ein Disparitätenbild erstellt. Die Ausgabe dieses Filters kann einem weiteren Filter bereitgestellt werden, welches weitergehende Analysen an der Disparitätenkarte vornimmt. Im Auswahlmenü des *Experimental Commanders* ist dieses Filter durch den Button *„Stereo Disparity“* erzeugbar.

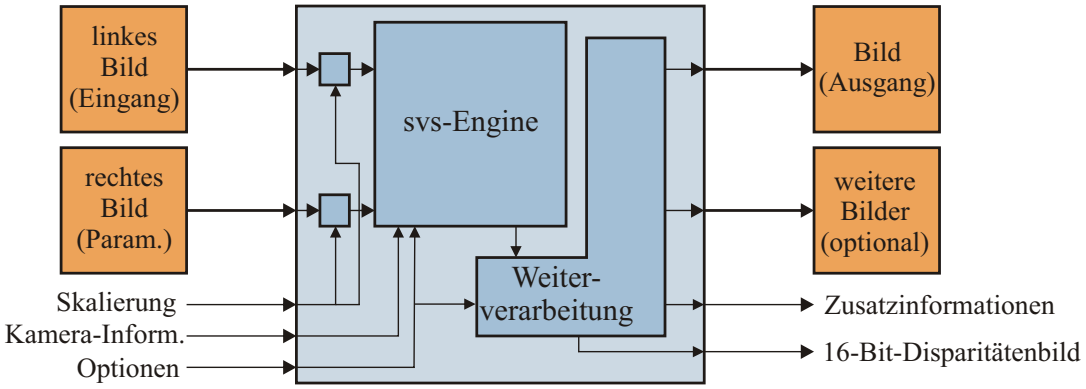


Abbildung 4.8: Aufbau des Filters zur Disparitätenschätzung. Zur Erstellung der 16-Bit-Tiefenkarten wird die svs-Bibliothek verwendet. Nach einer weiteren Filterung wird zusätzlich ein 8-Bit-Bild zur Ausgabe bereitgestellt. Dem Nutzer stehen innerhalb der Filterkette nur die orange dargestellten Bilder zur Verfügung – für die Verwendung der zusätzlichen Informationen wie dem 16-Bit-Bild ist ein direkter Zugriff auf das Filter innerhalb des Programms erforderlich.

Als Eingabe werden dabei zwei Bilder (d.h. Sequenzen) verwendet. Da das Konzept des dip-Framework bei der Filterkette nur ein Eingangsbild zulässt, muss das zweite als zusätzlicher Parameter angegeben werden. Das linke Bild wird vom Commander bereitgestellt – die bilderzeugende Kamera kann unter *„Actual Camera Name“* angegeben werden. Der Commander stellt dem Filter dann das – möglicherweise noch mit vorherigen Filtern versehene – Bild zur Verfügung. Das rechte Bild kann innerhalb des Filterdialogs unter *„Right Image“* ausgewählt werden.

Des weiteren werden Parameter zur Kalibrierung der Kameras benötigt, die aus einer Datei geladen werden. Dabei muss es sich um eine Kalibrierungsdatei der svs-Software handeln. Daten zur Rektifizierung der Bilder und Kameraeigenschaften wie deren Brennweite und Abstand sind dort vorhanden, so dass unter Verwendung von Gleichung 4.2 (Seite 42) die Disparität als konkrete Entfernung interpretiert werden kann.

Innerhalb eines Commanders ist allerdings nur die Verwendung von Grauwertbildern mit einer Farbtiefe von 8 Bit möglich. Da die svs-Software ein Tiefenbild mit 16 Bit pro Bildpunkt errechnet, findet zur Anzeige und für eventuell folgende Filter eine Umrechnung auf 8 Bit

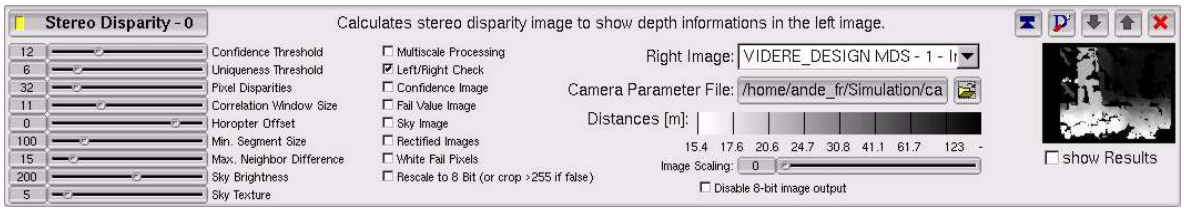


Abbildung 4.9: Die Benutzeroberfläche des Disparitätenfilters. Hier lassen sich alle Eingabe- und Ausgabeparameter festlegen.

statt. Unter *„Distances“* ist eine Skala zu sehen, die dem Nutzer die Information zur Verfügung stellt, welcher Grauwert (im aufbereiteten 8-Bit-Ausgabebild) welcher Entfernung entspricht.

4.4.1 Verwendung des Small Vision Systems

Das Stereofilter des Small Vision Systems produziert vorzeichenbehaftete 16-Bit-Bilder. Dabei haben die Pixel mit den Werten $g_{16}(x, y)$ folgende Kodierung:

- $g_{16}(x, y) \geq 0$: Disparität. Auf 1/16 Subpixel genau. Es ist $g_{16}(x, y) = d(x, y) \cdot 16 + x_{Offset}$. Je größer der Wert, desto näher ist das Objekt mit diesem Pixel des entzerrten und rektifizierten linken Originalbildes vom Betrachtungspunkt aus gesehen. Der negierte *Offset* ist der kleinstmögliche Disparitätswert des gewählten Suchbereiches.
- $g_{16}(x, y) = -1$: Fehler. Es konnte aufgrund fehlender Informationen keine Disparität errechnet werden oder die Filterung hat den Wert nachträglich als ungültig klassifiziert.
- $g_{16}(x, y) = -2$: Fehler. Der rechts/links-Vergleich ergab eine Fehlzuordnung. Ab Version 4.1 der svs-Bibliothek wird dieser Wert nicht mehr verwendet. Der Vergleich ist dort nicht abschaltbar und die ungültigen Disparitäten erhalten ebenfalls den Wert -1.

Für die Anzeige ist die Konvertierung in 8-Bit-Bilder ausreichend, allerdings gehen dadurch Informationen verloren. Vor allem können bei der üblichen Verwendung des gesamten Wertebereiches von $0 \ldots 255$ für Grauwerte die negativen Werte nicht dargestellt werden. Daher wird es Filtern, die auf das Stereofilter zugreifen, ermöglicht, auf das 16-Bit-Bildobjekt zuzugreifen, anstatt wie sonst üblich einfach in der Kette eingehängt zu werden.

Folgende Parameter lassen sich bei dem Stereofilter einstellen. Die Einstellung dieser Werte hat Einfluss auf die von der svs-Bibliothek benötigten Parameter:

- *Confidence Threshold:* Sicherheitsbegrenzung. Mit Hilfe dieses Schwellwertes lassen sich Disparitäten ausblenden, deren Schätzung nur sehr unzuverlässig ist. Ein hoher Wert unterdrückt vor allem Rauschen im Disparitätenbild – vor allem punktweise falsche Disparitätenschätzungen.
- *Uniqueness Threshold:* Einzigartigkeitsbegrenzung. Wird im rechten Bild ein Muster des linken gefunden, und im linken Bild dasselbe Muster im rechten, so müssen die dabei ermittelten Disparitäten einen betragsmäßig ähnlichen Wert aufweisen. Wird dieser Schwellwert auf 0 gesetzt, lässt sich die Überprüfung abschalten. Höhere Werte erfordern eine um so größere Ähnlichkeit der beiden Disparitätswerte, damit sie nicht gefiltert werden.
- *Number of Disparities:* Die Größe des Suchbereiches, in dem nach korrespondierenden Bildelementen gesucht wird.
- *Correlation Window Size:* Die Größe des Vergleichsfensters, welches für die Korrelationsanalyse zwischen den beiden Bildern verwendet wird.
- *Horopter Offset:* Der Suchbereich für die Disparitätenschätzung wird um diesen Wert nach links verschoben, bei negativen Werten entsprechend nach rechts. Positive Werte ermöglichen eine bessere Detektion weiter Distanzen, negative Werte die Erkennung von nahen Objekten.
- *Multiscale Processing:* Ist dieser Knopf aktiviert, wird das Disparitätenbild zusätzlich mit verkleinerten Ausgangsbildern errechnet und mit den Ergebnissen des Originalbildes kombiniert.
- *Left/Right Check:* Dieser Knopf aktiviert den links/rechts Vergleich der svs-Software zur Detektion von Okklusionen. Ab Version 4.1 hat diese Einstellung keine Auswirkungen.
- *Confidence Image:* Es wird ein Sicherheitsbild errechnet und zusätzlich zu dem Disparitätenbild ausgegeben. Die einzelnen Bildpunkte sind ebenfalls auf 16 Bit diskretisiert, zur Ausgabe wird das Bild allerdings auf 8 Bit heruntergerechnet.

4.4.2 Weiterverarbeitung

Die verwendete Bibliothek bietet eine Reihe von Optionen zur Filterung der errechneten Disparitätenbilder. Das in Abschnitt 4.2.2 vorgestellte Verfahren zur Beseitigung von kleinen Segmenten, die sich stark von der Umgebung unterscheiden, existiert dort nicht. Diese Fehlerbehebung wird daher mit einer eigenen Implementation realisiert und verwendet die

16-Bit-Bilddaten, die von der Disparitätenschätzung errechnet wurden. Sie lässt sich über folgende Parameter justieren:

- *Minimal Segment Size:* Die minimale Größe von Regionen. Kleinere Regionen werden ausgeblendet, d.h. als Fehlstelle mit -1 belegt. Auf Bereiche, die größer sind als der eingestellte Wert, hat die Filterung keine Auswirkung.
- *Maximal Neighbor Difference:* Maximale Differenz zwischen Nachbarpixeln. Weicht der Grauwert eines Pixels um diesen Betrag oder weniger von einem benachbarten Bildpunkt ab, werden beide zu der gleichen Region zugeordnet.
- *Sky Brightness:* Helligkeitsschwellwert für die Himmelserkennung. Bildpunkte des Originalbildes mit einem geringeren Wert gelten als zu dunkel.
- *Sky Texture:* Texturschwellwert der Himmelserkennung. Ist die Absolutdifferenz zwischen einem Bildpunkt und seinem Nachbar geringer sind als der angegebene Wert, gilt der Punkt als nicht strukturiert und ist damit ein möglicher Punkt des Himmels. Mit einem Wert von 0 gelten alle Punkte als strukturiert – die Erkennung des Himmels lässt sich auf diese Art abschalten.

Für die Konvertierung der Bilder in 8 Bit zwecks Anzeige und Filterausgabe stehen ebenfalls Optionen zur Verfügung:

- *White Fail Pixels:* Pixel, denen keine Disparität zugeordnet werden kann, werden im 16-Bit-Bild mit negativen Werten belegt – im 8-Bit-Bild müssen diese einem nichtnegativen Wert $f \in [0; 255]$ zugeordnet sein. Hier kann ausgewählt werden, ob das ausgegebene 8-Bit Grauwertbild diese Pixel mit Wert $f = 0$ (schwarz) oder $f = 255$ (weiß) belegen soll, da negative Werte nicht zulässig sind.
- *Rescale to 8 Bit:* Das Stereofilter kann je nach Anzahl der durchsuchenden Disparitäten bis zu $128 \cdot 16 = 2048$ Helligkeitsstufen ausgeben – ein 8-Bit Grauwertbild lässt allerdings nur $2^8 = 256$ verschiedene Werte zu. Ist die Option ausgeschaltet, ist:

$$g_8(x,y) = \begin{cases} f, & \text{falls } g_{16}(x,y) < 0; \\ g_{16}(x,y), & \text{falls } 0 \leq g_{16}(x,y) \leq 255; \\ 255, & \text{sonst.} \end{cases} \tag{4.4}$$

  Ist die Option angewählt, wird das Bild auf den Bereich $[0; 255]$ normiert. Es ist

$$g_8(x,y) = \begin{cases} f, & \text{falls } g_{16}(x,y) < 0; \\ g_{16}(x,y) \cdot \frac{256}{n}, & \text{sonst.} \end{cases} \tag{4.5}$$

  wobei n die Anzahl möglicher Werte für alle $g_{16}(x,y)$ ist. Bei svs entspricht n dem 16fachen der Disparitätenanzahl.

  Bei ausgeschalteter Option erscheint das so entstehende Bild heller und die Entfernungsmessung zu weiter entfernten Objekten ist auf dieser Basis genauer. Detaillierte

Informationen über Objekte im Nahbereich gehen jedoch verloren – als nahe Objekte sind sie allerdings noch identifizierbar.

Weitere Optionen:

- *Fail Value Image:* Ein Binärbild mit allen Fehlpixeln, d.h. mit allen ungültigen Punkten im 16-Bit-Bild, wird zusätzlich ausgegeben. Dies kann sinnvoll sein, da im Ausgabebild keine Informationen über ungültige Werte enthalten sind – aus der zugewiesenen Helligkeit lässt sich dies nicht rückwirkend schließen.
- *Sky Image*: Ausgabe eines Binärbildes, welches die als Himmel erkannten Punkte des rektifizierten linken Eingangsbildes weiß und die restlichen Punkte schwarz darstellt.
- *Rectified Images:* Die rektifizierten und entzerrten Eingangsbilder werden ausgegeben. Dies lässt z.B. einen Vergleich zwischen Bildpunkten im Bild und im Disparitätenbild zu, da sie das gleiche Objekt darstellen.
- *Disable 8-bit image output:* Es wird kein 8-Bit Bild erzeugt. Bei der Verwendung von weiteren Filtern, die lediglich das 16-Bit Bild benutzen und bei denen keine 8-Bit-Visualisierung desselben benötigt wird, ist die Erzeugung dieses Bildes nicht erforderlich. Dies bewirkt einen minimalen Geschwindigkeitsvorteil.
- *Image Scaling:* Wird hier der Wert k angegeben, wird die Auflösung die Eingangsbilder um den Faktor 2^k verringert. Die Ausgabebilder sind damit entsprechend kleiner.

4.4.3 Ausgaben

Das Filter kann folgende Bildausgaben erzeugen, sie sind in Abbildung 4.10 dargestellt:

- Das 16-Bit-Tiefenbild. Eine Anzeige dieses Bildes ist nicht möglich.
- *Stereo Disparity:* Das 8-Bit-Tiefenbild zur Anzeige.
- *Rectified Left / Right Image:* Die rektifizierten und skalierten Eingangsbilder.
- *Sky Detection:* Bild, das die Segmentierung in Himmel (weiß) und Erde (schwarz) darstellt.
- *Stereo Disparity Fails:* Das Bild mit den ungültigen Werten (weiß) des Tiefenbildes.
- *Stereo Confidence:* Ein Bild mit den Konfidenzwerten der Disparitätenschätzung.

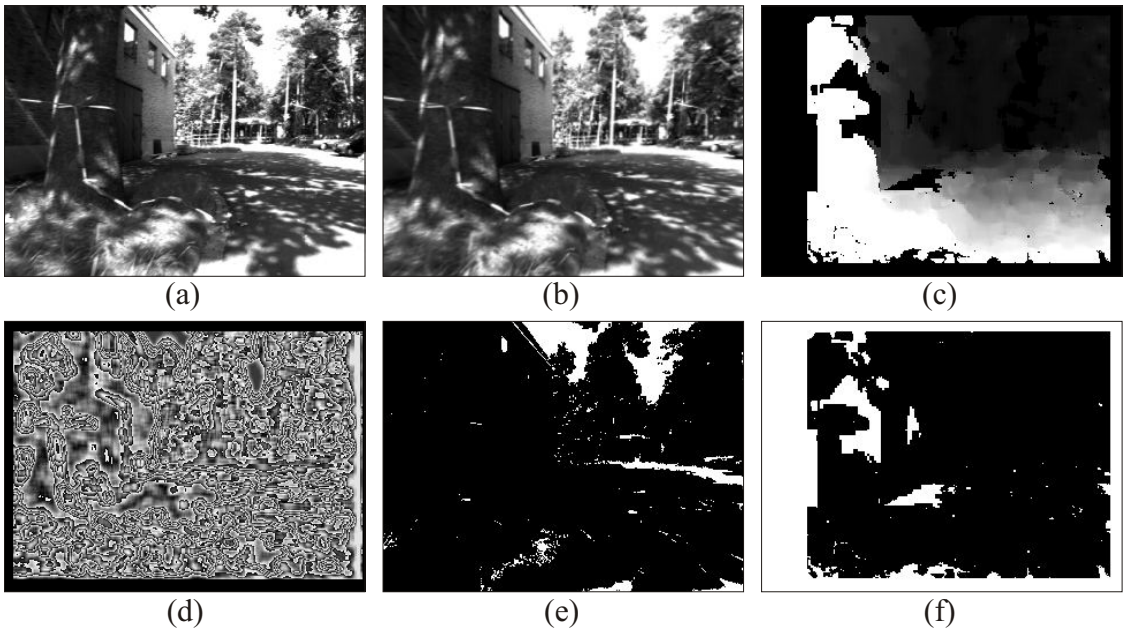


Abbildung 4.10: Eingaben und Ausgaben des Stereofilters. Linkes Eingangsbild (a), rektifiziertes Bild (b), Stereobild zur Ausgabe (c), Konfidenzbild (d), Himmel (e) und ungültige Werte des Tiefenbildes (f).

4.5 Leistungsbewertung

4.5.1 Allgemeines

Versuche haben gezeigt, dass auch bei offensichtlich nur seitlich verschoben angeordneten Kameras eine Kalibrierung zur Herstellung eines achsparallelen Modells zwingend erforderlich ist. Ohne Entzerrung und Rektifikation sind die mit der svs-Software errechneten Disparitätenbilder unbrauchbar. Auch bei nur geringen Änderungen, wie beispielsweise der relativen Lage der Kameras zueinander, muss eine Neukalibrierung erfolgen.

Die Verwendung nachträglich skalierter Kamerabilder mit den gleichen Kalibrierungsinformationen ist hingegen möglich. Bedeutung hat dies beispielsweise, wenn zur möglichst schnellen Bildverarbeitung die Bilder verkleinert werden. Eine Neukalibrierung ist dann nicht erforderlich.

Bei den entstandenen Tiefenbildern fällt ebenfalls auf, dass im Randbereich keine Informationen verfügbar sind. Bei einer getesteten Beispielsequenz sind die Ränder oben 10, rechts 11, unten 11 Bildpunkte breit. Die Breite des linken Randes ist abhängig von der Disparitätenanzahl $9 + n$, dabei ist n die Größe des Suchbereiches. Negative Horoptereinstellungen können diesen Rand verschieben. Das bedeutet vor allem, dass bei einer höheren Anzahl geschätzter Disparitäten der linke Rand breiter und der daraus resultierende Bereich, in dem

prinzipiell Tiefeninformationen verfügbar sind, umso kleiner ist. Bei gleicher Kamerakalibrierung ist die Breite der Ränder unabhängig von der Bildauflösung.

Die Erstellung von Disparitätenbildern erfolgt in Echtzeit, entsprechend schnelle Hardware vorausgesetzt. Tabelle 4.1 listet die Bildwiederholraten des Stereofilters unter verschiedenen Bedingungen auf.

Intel Celeron mit 1,7 GHz Taktfrequenz, 512 MB Arbeitsspeicher und Windows XP:

Anzahl Disparitäten	16	32	64	128
640 × 480, keine Filterung	6,4 Hz	5,6 Hz	3,6 Hz	1,6 Hz
640 × 480, Filterung	6,8 Hz	6,0 Hz	3,7 Hz	1,6 Hz
320 × 240, keine Filterung	25,6 Hz	23,0 Hz	19,5 Hz	15,9 Hz
320 × 240, Filterung	26,9 Hz	24,0 Hz	20,3 Hz	16,3 Hz

Intel Pentium 4 mit 3,0 GHz Taktfrequenz, 512 MB Arbeitsspeicher und SuSE Linux 9.3:

Anzahl Disparitäten	16	32	64	128
640 × 480, keine Filterung	9,3 Hz	8,5 Hz	7,3 Hz	5,9 Hz
640 × 480, Filterung	9,8 Hz	9,0 Hz	7,6 Hz	6,1 Hz
320 × 240, keine Filterung	39,3 Hz	35,8 Hz	31,4 Hz	26,4 Hz
320 × 240, Filterung	41,9 Hz	37,7 Hz	32,8 Hz	27,2 Hz

Tabelle 4.1: Ergebnisse einer Geschwindigkeitsmessung auf verschiedenen Rechnersystemen. Angegeben sind die Bildwiederholraten für mehrere Anzahlen zu durchsuchender Disparitäten bei unterschiedlichen Bildauflösungen mit und ohne der Filterung von Fehlstellen.

4.5.2 Genauigkeit

Aus Gleichung 4.2 folgt, dass die Entfernungsauflösung, d.h. die geringste Differenz ΔZ der Distanz zweier Objekte in einem bestimmten Entfernungsbereich Z umso größer ist, je weiter die Objekte vom Betrachter entfernt sind. Die gemessene Disparität der beiden Punkte unterscheidet sich um den kleinsten messbaren Wert Δd, d.h. der Breite eines Bildpunktes oder Subpixels bei der Verwendung von Interpolation. Bei Verwendung der svs-Bibliothek entspricht dies $\frac{1}{16}$ der Pixelbreite.

Auch wenn sich hier Disparitäten so genau darstellen lassen, wird diese Genauigkeit bei der Auswertung von Stereobildern nicht erreicht. Williamson [Wil98] nennt einen möglichen Wert von $\frac{1}{4}$ Pixel. Nach Messungen von Scharstein und Szeliski [SS01] sei die Genauigkeit noch geringer, eine realistische Genauigkeit von $\frac{1}{2}$ Pixel wird angegeben. Abbildung 4.11 zeigt die Entfernungsauflösung von zwei verschiedenen Kamerasystemen.

Es ergibt sich ein theoretischer Wert von

$$\Delta Z = \frac{Z^2}{b \cdot f} \cdot \Delta d, \tag{4.6}$$

was unter anderem auch bedeutet, dass große Basisabstände und Brennweiten einen positiven Einfluss auf die Entfernungsauflösung haben. Im Fernbereich bei kleinen Disparitätswerten ist die Genauigkeit sehr ungenau, was bei der Auswertung der gemessenen Entfernungen berücksichtigt werden muss [KB05, S. 37]. Bei nachträglicher Verkleinerung der Eingangsbilder erhöht sich die Größe der Pixel, bzw. es verringert sich der in Pixel angegebene Wert der Brennweite. Demzufolge verschlechtert sich die Entfernungsauflösung.

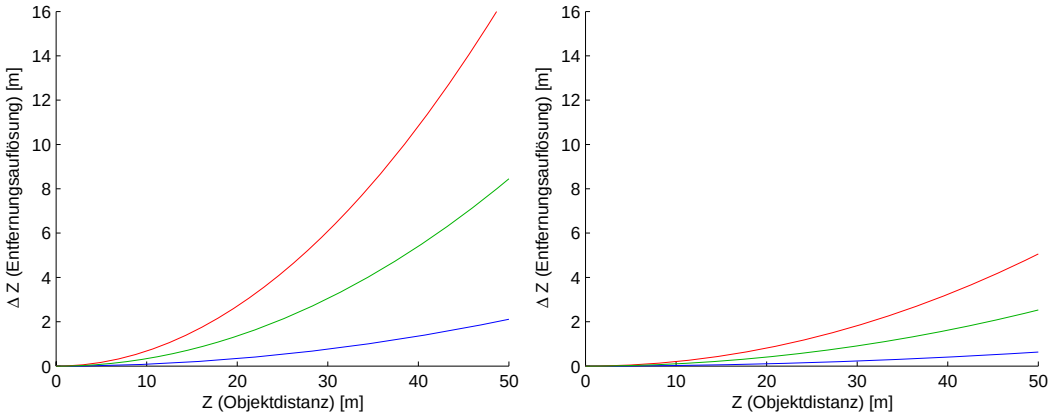


Abbildung 4.11: Entfernungsauflösung ΔZ bei Genauigkeiten Δd von 1/16 (blau), 1/4 (grün) und 1/2 Pixel (rot). Das linke Diagramm zeigt die Auflösung für $b = 150$mm und $f = 510$ Pixel, das rechte Diagramm für $b = 300$mm und $f = 824$ Pixel. Die Parameter stammen von der verwendeten Stereokamera und den ermittelten Werten bei der Simulation in Kapitel 6.

Darüber hinaus existiert durch Unschärfe der Kamera und Rauschen ein Projektionsfehler, d.h. eine statistische Abweichung der tatsächlichen Position eines Bildpunktes von der mit Hilfe der inneren und äußeren Parameter der Kamera theoretisch errechneten Position. Laut Hersteller [KB04, S. 12] sollte dieser Fehler zwischen $0,05$ und $0,1$ Pixel liegen. Bei der Kalibrierung der Kamera für Außenaufnahmen sind allerdings Fehler von bis zu $0,47$ Bildpunkten festgestellt worden.

4.5.3 Messbarer Entfernungsbereich

Der messbare Entfernungsbereich ist nach oben nicht begrenzt, da eine Disparität von 0 rechnerisch eine unendliche Entfernung bedeutet. Durch die Ungenauigkeit bedeutet dieser Disparitätswert allerdings lediglich, dass eine bestimmte Entfernung erreicht bzw. überschritten ist – ab dieser Entfernung ist eine unterschiedliche Projektion bei beiden Bildebenen nicht mehr messbar. Bei der Auswertung der Tiefenbilder sollte daher eine Maximaldistanz $Z_{\max}$ festgelegt werden. Größere Entfernungen werden auf diesen Schwellwert gesetzt oder freie Bereiche nur als frei bis zu diesem Wert angesehen.

Nach unten existiert allerdings eine Grenze. Sehr nahe Objekte werden nur von einer der beiden Kameras erfasst und eine Entfernungsmessung ist damit nicht möglich. Die maximal

messbare Disparität entspricht der Breite des Bildsensors, d.h. die Entfernung die zu diesem Wert gehört ist die kleinste, die gemessen werden kann. Es ist also

$$Z_{\min} = \frac{b \cdot f}{d_{\max}}, \tag{4.7}$$

woraus sich ein Wertebereich von $[Z_{\min}, \infty)$ für die gemessenen Entfernungen ergibt. Mit der Verwendung eines je nach Anwendung zu spezifizierenden Schwellwertes $Z_{\max}$ ist der Wertebereich dann $[Z_{\min}, Z_{\max}]$. In der Anwendung ist auch $d_{\max}$ durch die Größe des Suchbereiches meist geringer als die Sensorbreite.

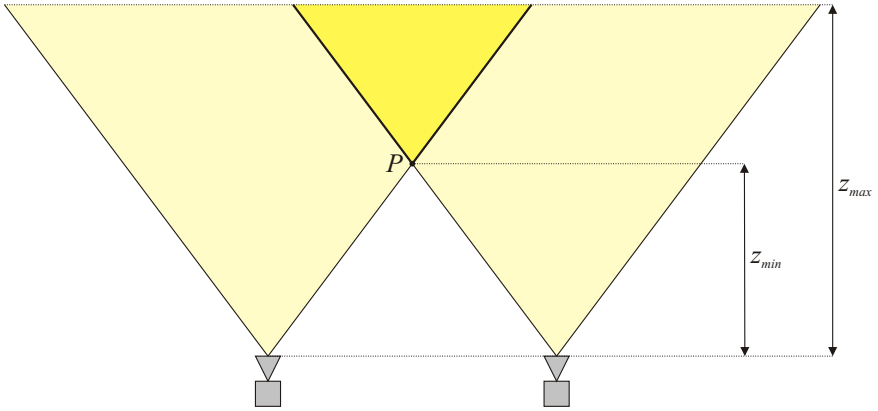


Abbildung 4.12: Sichtbare Bereiche von zwei Kameras. Nur der dunkel gefärbte Bereich ist von beiden Kameras zu sehen, er beginnt ab der Entfernung $Z_{\min}$ bei Punkt P. Dieser wird auf dem linken Bild ganz rechts und auf dem rechten ganz links abgebildet. Die obere Grenze $Z_{\max}$ wird vom Benutzer festgelegt.

Bei der Anwendung zeigt sich jedoch, dass gerade bei der Verwendung von sehr kleinen Horopteroffsets zwar kürzere Entfernungen gemessen werden können, die Erkennung geringer Disparitäten jedoch ausgeschlossen wird, da erst ab einer bestimmten Disparität mit dem Suchen von Korrespondenzen begonnen wird. Bei Außenanwendungen ist allerdings stets mit weiten Entfernungen zu rechnen, wodurch Disparitäten ab null auch geschätzt werden sollten. Findet dies nicht statt, so wie in diesem Fall, werden entfernte Bereiche falsch interpretiert, da höchstens eine falsche Korrespondenz gefunden werden kann. Die dabei entstandenen Bilder sind praktisch unbrauchbar – auf die Verwendung sehr kleiner Horopteroffsets sollte daher verzichtet werden.

Große Werte hingegen können zumindest für eine bessere Visualisierung entfernter Regionen verwendet werden. Für die Auswertung der Disparitätenkarten ergibt sich allerdings kein weiterer Nutzen, denn statt des Suchbereichs $[0; n-1]$ wird nun der Bereich $[0 - \text{Offset}; n - 1 - \text{Offset}]$ für das Finden korrespondierender Bildregionen verwendet. Da im verwendeten Kameramodell keine negativen Disparitäten entstehen können, schränkt ein positiver Offset nur den Suchbereich auf $[0; n - 1 - \text{Offset}]$ ein und bringt keine zusätzlichen Informationen.

Es ist zu empfehlen, die Einstellung bei 0 zu belassen. Die maximal messbare Disparität ist damit durch die Anzahl durchsuchender Disparitäten begrenzt und meist geringer als die Bildbreite.

4.5.4 Untersuchung von Fehlstellen

Wie schon bei der Beschreibung des Stereofilters erwähnt, ist eine Tiefenschätzung texturschwacher bzw. kantenloser Regionen nur schwer möglich. In Abbildung 4.13 ist das linke Bild eines Stereobildpaares und dessen Faltung mit den Masken

$$M_{\text{Sobel}} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}, \quad \text{bzw.} \quad M_{\text{Laplace}} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \tag{4.8}$$

dargestellt. Dies verdeutlicht den Zusammenhang zwischen Regionen schwacher Textur und jenen Regionen, für die keine Tiefenschätzung erfolgen kann.

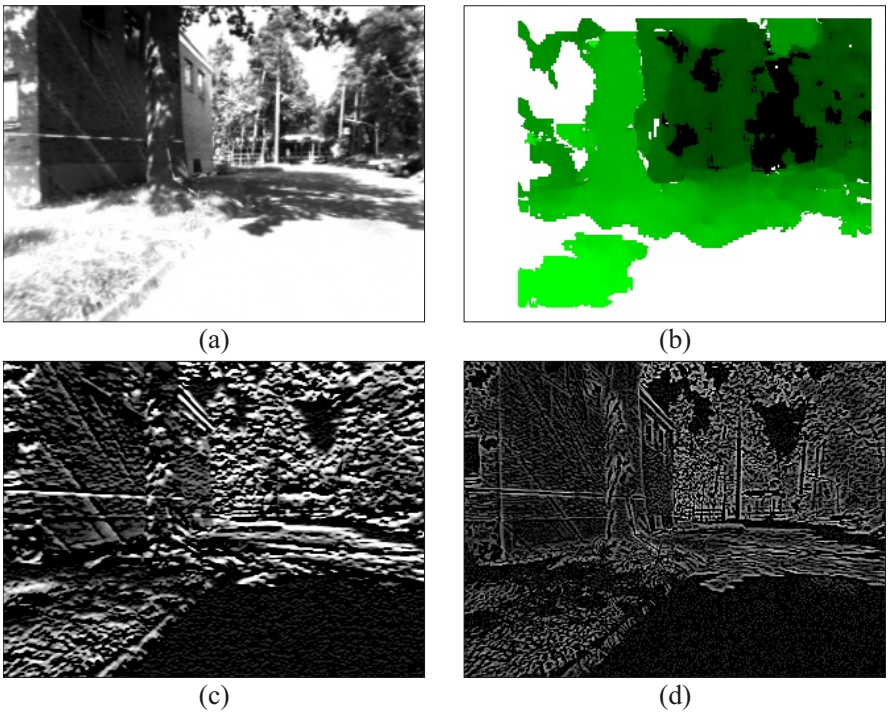


Abbildung 4.13: Untersuchung von Bereichen eines Stereobildpaares (a, nur das linke dargestellt), denen keine Disparität zugeordnet werden kann. Im Tiefenbild (b) sind geringe Entfernungen grün und hohe schwarz dargestellt, fehlerhafte Bereiche weiß. Die beiden unteren Bilder sind Ergebnisse der Ableitungen des Ausgangsbildes mit Hilfe eines horizontalen Sobel- (c) bzw. eines Laplace-Operators (d).

Bei der Betrachtung des Tiefenbildes fällt sofort auf, dass für einige Bereiche keine Tiefe existiert. Vom Randbereich abgesehen sind dies zumeist Stellen, in denen keine Kanten vorhanden

sind. Zur Ermittlung dieser kantenlosen Bereiche wurden auf das Ausgangsbild Ableitungsoperatoren angewendet, welche die Grauwertänderung und damit die Kanten darstellen. In diesem Bild ist vor allem der untere rechte Bereich sehr hell und wenig strukturiert. Dort ändert sich der Grauwert nur wenig, was in den gefilterten Bildern als durchgängig dunkler Bereich zu sehen ist. Die dort dargestellten dunklen Bereiche stimmen in etwa mit jenen im Disparitätenbild überein, bei denen keine Tiefe geschätzt werden kann. Wird zusätzlich Abbildung 4.14 betrachtet, wird deutlich, dass die anderen weißen Bereiche, z.B. links neben dem Baum, durch die Filterung von unzuverlässigen Schätzungen entstanden sind.

Bei anderen getesteten Bildern ergibt sich ähnliches. Es ist also zu erwarten, dass für Bildregionen, in denen keine oder nur wenig Struktur vorhanden ist, keine Tiefenschätzung erfolgen kann. Da die Ursachen für solche Bildregionen können natürlich gegeben sein können, wie es z.B. bei einfarbigen glatten Wänden oder dem Himmel der Fall ist, hilft eine nachträgliche Kontrastverbesserung nicht weiter. Die Erkennung naher und ferner Distanzen muss dort auf andere Weise erfolgen.

Die vorhandenen Tiefeninformationen scheinen allerdings kaum noch fehlerbehaftet zu sein. Mögliche Fehlmessungen werden durch die Anwendung der vorhandenen Filter auf das Bild so gut wie vollständig beseitigt.

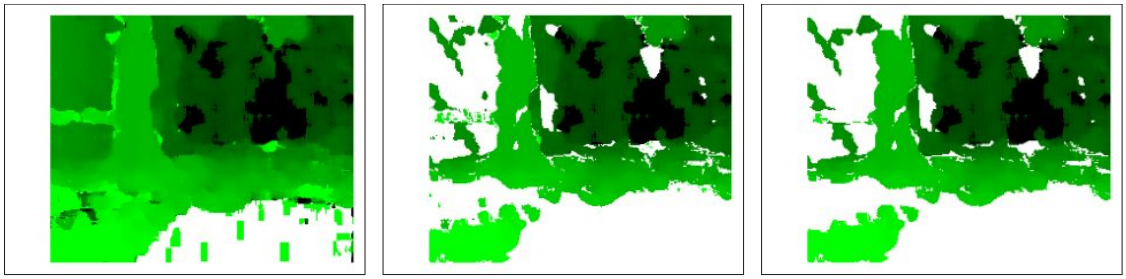


Abbildung 4.14: Tiefenbilder des gleichen Stereobildpaares ohne nachträgliche Filterung (a), mit Filterung von Bereichen geringer Konfidenz und Einzigartigkeit (b) und mit zusätzlicher Filterung von kleinen Segmenten (c).

In Abbildung 4.14 ist zu erkennen, dass bei fehlender Textur, hier unten rechts, keine Schätzung der Tiefe erfolgen kann. Ausnahmen bilden sehr kleine Bereiche wie der Himmel zwischen den Bäumen in der Mitte des oberen rechten Bildviertels. Die anderen fehlerhaften Bereiche entstehen durch die nachträgliche Filterung von Ergebnissen, die sehr wahrscheinlich falsch sein könnten. Dies sind beispielsweise Bereiche mit geringer Texturierung wie der Hauswand links. Das Konfidenzfilter beseitigt solche Bereiche, auch wenn die Gefahr besteht, korrekte Ergebnisse ebenfalls als Fehler anzusehen.

Auffällig sind die falschen Messungen durch Okklusionen. Links neben dem Baum werden sehr nahe Entfernungen vermutet, die in der Realität nicht vorhanden sind. Diese falschen

Messungen können gut beseitigt werden. Im mittleren Bild ist das Ergebnis einer solchen Filterung zu sehen. Übrig bleiben größere Flächen ähnlicher Tiefe, die wahrscheinlich größere Objekte darstellen und kleinere Fehlstellen, die jedoch vom Tiefenfilter fälschlicherweise als sichere Schätzung angesehen werden. Durch die Filterung kleiner Bereiche können diese Stellen ebenfalls beseitigt werden – das Ergebnis ist im rechten Bild zu sehen.

Vergleicht man dieses Disparitätenbild mit dem Ausgangsbild (Abbildung 4.13a), so kann dies durchaus als eine gelungene Messung angesehen werden. Der Baum als Hindernis im linken Bereich (hellgrün) und ein freier, hindernisfreier Bereich (schwarz) im rechten Bereich sind gut erkennbar. Wirklich „falsche“ Entfernungen sind nicht vorhanden, allerdings Bereiche, für die keine Aussage getroffen werden kann. Kleine Lücken im Bild dürften für eine spätere Auswertung unproblematisch sein. Werden sie jedoch allzu groß, wie in diesem Beispiel links oder unten rechts, kann dies durchaus bedeuten, dass ein mögliches Hindernis übersehen wurde.

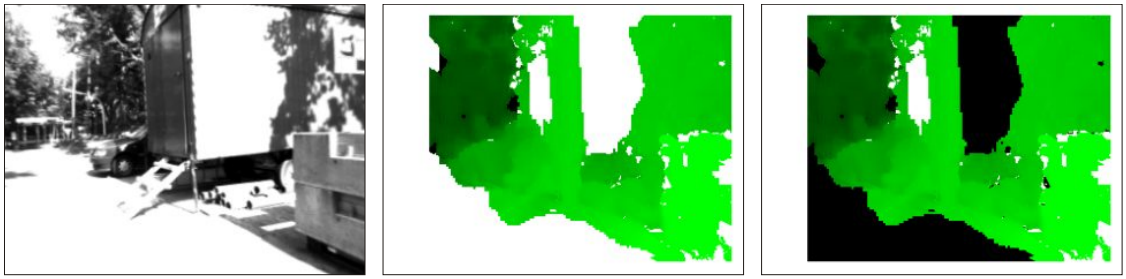


Abbildung 4.15: Linkes Bild des Stereobildpaares einer Außenszene (a), das Tiefenbild ohne (b) und mit Himmelserkennung (c).

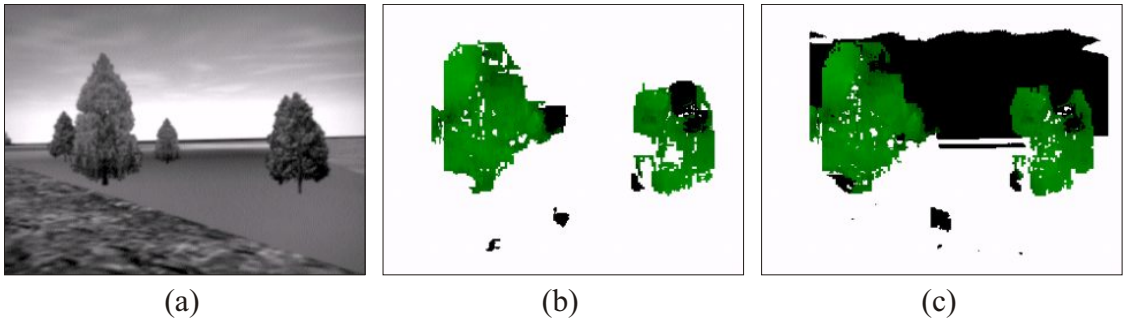


Abbildung 4.16: Linkes Bild des Stereobildpaares einer simulierten Außenszene (a), das Tiefenbild ohne (b) und mit Himmelserkennung (c).

Mit der Himmelserkennung können Regionen als entfernt klassifiziert werden, über die bei der Disparitätenschätzung keine Aussage getroffen werden kann. Mögliche Ergebnisse werden in den Abbildungen 4.15 und 4.16 miteinander verglichen.

Die Erkennung des Himmels in der vorgestellten Form bestimmt alle Bereiche die hell und unstrukturiert sind. Dadurch können auch sonstige Bildabschnitte, die Hindernisse sein können aber eben diese Eigenschaften aufweisen, fälschlicherweise als weit entfernt angesehen werden. Gerade bei sehr hellen bzw. stark belichteten Aufnahmen besteht diese Gefahr. Bei der ebenen Fläche des Bauwagens auf der rechten Seite und der Straße in Abbildung 4.15 wird eine solche Fehlklassifikation deutlich. Die Stärken zeigen sich bei Aufnahmen, bei denen Objekte vergleichsweise dunkel sind. In Abbildung 4.16 aus der Simulationsumgebung werden die Hindernisse des Bildes erkannt, allerdings keine entfernten Bereiche. Mit Hilfe der Himmelserkennung werden die freien Bereiche (schwarz) sichtbar.

Kapitel 5

Kollisionsvermeidung

5.1 Überblick

Dieses Kapitel beschäftigt sich mit der Auswertung von Disparitätenbildern zur Erkennung von Hindernissen und zur Kollisionsvermeidung. Ausgangspunkt für die Implementation ist das dip-Framework, an dem beschriebene Erweiterungen vorgenommen wurden. Die Kamerabilder und das Ergebnis des Filters, das die Tiefenbilder errechnet, werden für die Kollisionsvermeidung verwendet.

5.1.1 Grundlegende Ansätze

Ergebnis der Bildanalyse soll einerseits die Aussage sein, ob sich vor dem Helikopter Hindernisse befinden, die zur Kollision führen könnten. Andererseits sollen Möglichkeiten zum Ausweichen gefunden werden.

Im Tiefenbild sind nahe und entfernte Bereiche verzeichnet, aus Sicht der Kollisionsvermeidung kann dies als Sensorinformation angesehen werden. Es sind für die Auswertung zwei verschiedene grundlegende Ansätze denkbar:

1. Die Klassifikation von Hindernissen. Ein mögliches Resultat sind Objektgrößen und deren Position in einem Weltkoordinatensystem. Es wird eine Karte erstellt und daraus ein Pfad abgeleitet.

2. Die Klassifikation von Bereichen, in denen sich keine Hindernisse befinden und die Ausgabe von Zielpunkten, zu denen ein kollisionsfreier Flug möglich ist.

Insbesondere die Robotik befasst sich ausführlich mit der Thematik, Hindernisinformationen aus Sensordaten zu extrahieren. Dabei hat sich die Erstellung von Umgebungskarten als nützlich erwiesen, da sie durch Integration der Sensordaten über die Zeit erlaubt und dadurch

weniger anfällig für Fehler ist. Beispiele dafür sind Verfahren von Elfes [Elf87], bzw. Borenstein und Koren [BK91]. Dort werden zweidimensionale Karten mit Hilfe von Sonardaten erstellt, die Karten sind gerastert und geben für jeden 2D-Punkt eine Art Sicherheitswert an. Mit einem festzulegenden Schwellwert werden diese Punkte dann als Hindernisse bzw. freie Bereiche gewertet.

Die Verfahren zur Erstellung von Rasterkarten sind prinzipiell auch auf andere Sensoren und dreidimensionale Karten übertragbar, allerdings ist die Datenmenge bei dreidimensionalen Rasterkarten wesentlich höher. Die Verwendung verschiedener Detailstufen oder Polygonen zur Modellierung der Umgebung ist daher sinnvoll. Darüber hinaus muss die Position bzw. bei Integration über die Zeit auch die Bewegung des Sensors bekannt sein. Die Position aus GPS-Daten ist eventuell zu ungenau, die Errechnung der Position mit Hilfe photogrammetrischer Verfahren aus den Bilddaten sehr rechenintensiv.

Es existieren verschiedene Ansätze zur Kollisionsvermeidung bzw. Wegplanung, die Karten mit Hindernissen voraussetzen. Ein Ansatz von Minguez und Montano [MM04] ist das Finden von besonders großen Lücken zwischen den Hindernissen unter Berücksichtigung des Lenkverhaltens. Am DLR wurden graphentheoretische Modelle von Heitzmann-Gabrielli [Hei05] zur Kollisionsvermeidung untersucht. Dabei wird von einem Polygonmodell der Umgebung ausgegangen. Sind genaue und vor allem fehlerfreie Karten vorhanden, können dort kollisionsfreie Wege gefunden werden.

Einen anderen Ansatz wählt Hrabar [Hra06], [HSC+05] für die Kollisionsvermeidung eines unbemannten Helikopters. Wegen der erforderlichen Bildauswertung in Echtzeit wird auf die Positionsbestimmung und die Erstellung von Karten komplett verzichtet. Die Detektion von Hindernissen verwendet nur die aktuelle Tiefeninformation und berücksichtigt keine vorangegangenen Messungen. Allerdings erfolgt eine Fusion der Daten der Stereokamera mit den Daten von zwei seitlich vorhandenen Einzelkameras, die die Entfernung zu Hindernissen mit Hilfe des optischen Flusses bestimmen.

Über Schwellwerte werden nur Hindernisse eines bestimmten Entfernungsbereiches als gefährlich angesehen. Des weiteren werden unten liegende Hindernisse ignoriert, es wird davon ausgegangen, dass der Helikopter darüber hinweg fliegt. Objekte im betrachteten Bereich werden zusammen als ein Hindernis angesehen, das Ausweichen erfolgt dann reaktiv in eine andere Richtung.

Das Verfahren hat jedoch zwei grundlegende Schwachstellen. Mit der Zusammenfassung von allen Objekten zu einem Hindernis wird dem Helikopter die Fähigkeit entzogen, Lücken zwischen mehreren Hindernissen zu erkennen und dort hindurchzufliegen. Durch die Beschneidung des unteren Bildbereiches ist das Ausweichen nach unten nicht mehr möglich.

5.1.2 Eigener Ansatz

Die im Rahmen dieser Arbeit implementierte Kollisionsvermeidung für den ARTIS-Helikopter ist wie der von Hrabar [Hra06] gewählte Ansatz ausschließlich reaktiv. Hintergrund ist vor allem die begrenzte Rechenleistung, wegen der auf eine Aufzeichnung von Karten und der Positionsbestimmung aus Bilddaten verzichtet wird. Die Positions- bzw. Bewegungsdaten des GPS-Sensors werden ebenfalls nicht verwendet, um auch in sicherheitskritischen Situationen wie dem Ausfall dieses Systems ein Umfliegen von Hindernissen zu ermöglichen.

Dabei werden folgende Annahmen gemacht:

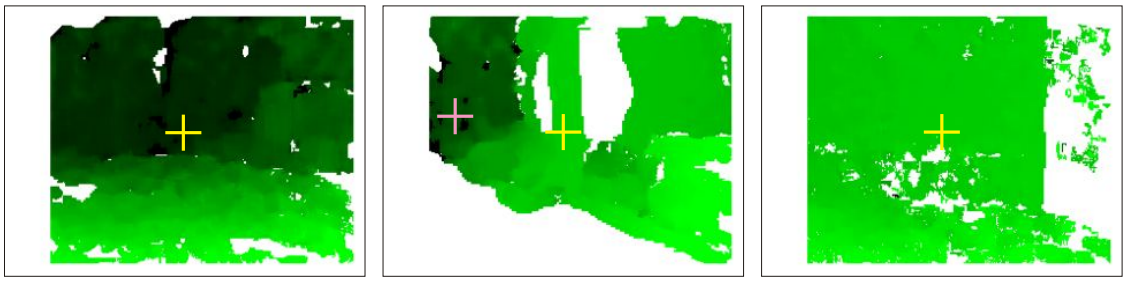
- Die bildbasierte Kollisionsvermeidung funktioniert nur unter eingeschränkten Flugbedingungen. Es ist zwingend notwendig, dass die Flugbahn im Blickfeld der Kamera liegt. Dies bedeutet zum einen, dass der Helikopter nur vorwärts fliegen bzw. als Grenzfall sehr langsamer Geschwindigkeit eine Schwebefluglage einnehmen darf, zum anderen wird zur Vereinfachung ein exakter Geradeausflug in x_f-Richtung angenommen. Die Manövrierbarkeit beschränkt sich dadurch auf die Änderung der Geschwindigkeit, einer horizontalen und einer vertikalen Lenkbewegung.

- Die Stereokamera ist nah am Zentrum des Helikopters in Flugrichtung mit möglichst wenig Rotation angebracht. Die optische Achse (z_c) der linken Kamera entspricht damit in etwa der x_f-Achse des Helikopterkoordinatensystems, geringfügige Abweichungen sind zulässig.

- Aufgrund der Tatsache, dass die Sensorik nur bis zu einer bestimmten Distanz verwertbare Ergebnisse liefert, wird eine maximale Entfernung festgelegt. Entferntere Objekte werden nicht als Hindernis klassifiziert. Die Maximalentfernung muss allerdings so gewählt werden, dass innerhalb dieses Bereiches ein vollständiges Abbremsen oder Ausweichen des Helikopters möglich ist. Andernfalls können Kollisionen mit zu spät erkannten Objekten nicht vermieden werden.

Der hier vorgestellte Ansatz betrachtet zwei verschiedene Arten von Reaktionen, die auch Urmson [Urm05] und Hrabar [Hra06] zur Vermeidung von Kollisionen nennen – *Ausweichen* und *Bremsen*. Das Ausweichen benötigt einen Zielpunkt, zu dem der Helikopter sicher fliegen kann. Zum Bremsen wird lediglich die Distanz zu möglichen Objekten benötigt, die sich in der aktuellen Flugbahn befinden oder dieser zu nahe kommen.

Abbildung 5.1 verdeutlicht, auf welche Art von Tiefeninformation eine bestimmte Reaktion stattfinden muss. Das linke Tiefenbild (a) enthält kein nahes Hindernis in Flugrichtung, somit ist keine Kursänderung notwendig. Im mittleren Bild (b) würde das Beibehalten der Flugrichtung zu einer Kollision führen. Allerdings ist auf der linken Seite ein freier Bereich zu sehen, er ist rot markiert. Die Kursänderung in diese Richtung vermeidet die Kollision mit



(a) (b) (c)

Abbildung 5.1: Beispiele von Tiefenbildern der Umgebung, die verschiedene Reaktionen nach sich ziehen. Der aktuelle Kurs ist mit einem gelben Fadenkreuz markiert, ein alternativer Kurs im mittleren Bild mit einem roten Kreuz.

dem Hindernis in der Mitte. Im rechten Bild (c) fliegt der Helikopter auf eine Wand zu, hier muss ebenfalls eine Bewegungsänderung erfolgen. Da keine hindernisfreien Bereiche im Bild detektiert werden können, muss ein Stopp erfolgen.

Im Idealfall sind alle Informationen, wo sich Hindernisse und hindernisfreie Bereiche befinden, im Tiefenbild enthalten. Dazu werden hier zwei verschiedene Auswertungen des Bildes vorgenommen:

1. Die Entfernungsmessung zu dem Bildbereich, der sich direkt vor dem Helikopter bzw. in Flugrichtung befindet.

2. Die Suche nach alternativen Flugzielen. Zunächst kommen alle dunklen Bildbereiche als Ziel infrage. Allerdings muss für diese Punkte wie bei 1. die Entfernung zu diesem Bereich gemessen werden. Dies ist eine Überprüfung, ob der Helikopter gefahrlos zu einem Zielpunkt fliegen kann.

In beiden Fällen genügt es nicht, lediglich einen Bildpunkt zu betrachten und aus dem Wert dieses Punktes im Tiefenbild die Entfernung auszugeben. Stattdessen wird vom Betrachter aus zu diesem Punkt ein Sicherheitskorridor festgelegt. Dieser muss so groß gewählt werden, dass der Helikopter gefahrlos hindurchfliegen kann. Hier wird ein zylinderförmiger Korridor verwendet, der durch den Radius und die Achse parametrisiert wird.

Für die Punkte des Tiefenbildes wird überprüft, ob deren 3D-Koordinaten innerhalb dieses Korridors liegen. Findet sich kein Punkt in diesem Korridor, so gilt der Bereich als frei. Finden sich dort Punkte, so wird die geringste Entfernung eines Punktes ausgegeben. Es ist offensichtlich, dass die Information, ob ein Punkt innerhalb des Sicherheitskorridors liegt, nicht nur von der Position im Bild, sondern auch von der Entfernung abhängt. Beispielsweise würden auch die am Rand liegenden Bildpunkte in Abbildung 5.2b und c bei geringer Distanz zum Betrachter Objekte innerhalb des Sicherheitskorridors repräsentieren.

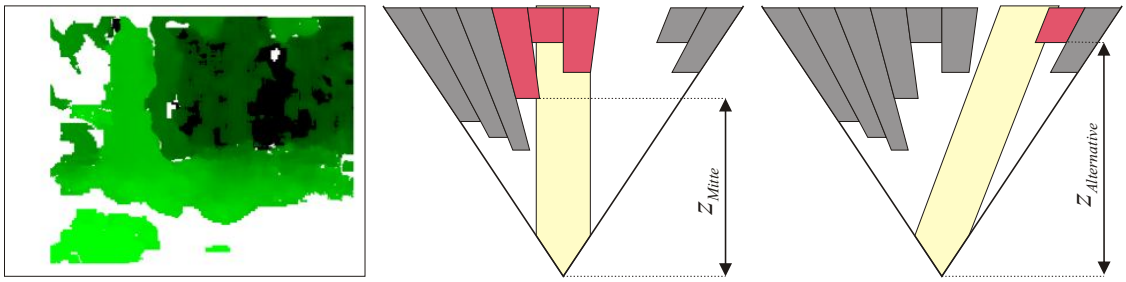


Abbildung 5.2: Grundprinzip der Tiefenbildauswertung. Jeder Punkt des Tiefenbildes (a) repräsentiert ein Objekt in bestimmter Entfernung und Größe. In (b) wird die Entfernung nach vorne gemessen, es wird für alle Punkte überprüft, ob sie Hindernisse innerhalb eines Sicherheitskorridors (gelb) sind. Im rechten Bild (c) passiert das gleiche für alternative Zielpunkte. Die Hindernisse sind dabei rot markiert, die geringste Entfernung eines markierten Hindernisses ist die Entfernung eines Zielpunktes. Die Bilder (b) und (c) sind eine vereinfachte Darstellung der aus dem Tiefenbild rekonstruierten Szene von oben, das Verfahren rekonstruiert die Umgebung in drei Dimensionen.

5.2 Funktionsweise

Dieser Abschnitt beschreibt die Funktionsweise des implementierten Verfahrens zur Kollisionsvermeidung, dessen Idee oben dargestellt ist.

5.2.1 Aufbau einer Bildpyramide

Ein trivialer Ansatz ist es, für jeden Bildpunkt der Tiefenkarte die Entfernung vom Betrachter zu bestimmen, bzw. den Punkt in Kamera- oder Helikopterkoordinaten zu rekonstruieren. Allerdings ist die Überprüfung von allen Pixeln sehr rechenintensiv und verlangsamt die Detektion von Hindernissen deutlich. Es liegt nahe, Tiefenkarten von geringerer Auflösung zu verwenden. Die Information, ob in bestimmten Bereichen mit Hindernissen zu rechnen ist oder nicht, ist in geringer aufgelösten Tiefenkarten, die nach dem hier beschriebenen Verfahren erstellt werden, ebenfalls enthalten.

Für die effiziente Lokalisation von Hindernissen bzw. hindernisfreien Bildabschnitten wird eine Pyramide des Disparitätenbildes erstellt. Dabei werden jeweils 4 Pixel der Ebene L in einer 2×2-Region zu einem Pixel in der nächsthöheren Ebene $(L+1)$ zusammengefasst. Dabei entspricht die unterste (0.) Ebene dem eingehenden Disparitätenbild. Zeilen und Spalten von Bildpunkten seien hier mit u bzw. v bezeichnet, deren Entsprechung in Kamerakoordinaten mit x bzw. y.

Es ist

$$I_0(u, v) = g_{16}(u, v), \tag{5.1}$$

bzw.

$$I_L(u,v) = \max\left\{I_{L-1}(2u,2v), I_{L-1}(2u+1,2v), I_{L-1}(2u,2v+1), I_{L-1}(2u+1,2v+1)\right\} \quad (5.2)$$

für $L > 0$. Nach dieser Methode entsteht in Ebene L ein gültiger Bildpunkt ($I_L(u,v) \geq 0$), sofern mindestens einer der vier Ausgangspunkte der Ebene $L-1$ gültig ist.

Es ist allerdings auch möglich, diesen Punkt als ungültig zu klassifizieren, wenn eine bestimmte Anzahl von fehlerhaften Ausgangspixeln erreicht ist, da es sich in einem solchen Fall um eine generell falsch geschätzte Region handeln könnte. Existieren unter diesen vier Punkten der Ebene $L-1$ mehr ungültige Werte (mit $I_{L-1}(u,v) < 0$) als ein bestimmter Grenzwert, so wird der entsprechende Punkt der darüber liegenden Ebene ebenfalls als ungültig markiert – in diesem Fall wird $I_L(u,v) = -1$ gesetzt.

Bei der Generierung der Bildpyramide kann die maximale Ebene festgelegt werden, wobei die Höhe prinzipiell nur so hoch sein kann, dass die Seitenlängen des Bildes in der Ebene nicht mehr durch 2 dividiert werden können. Beispielsweise lassen sich aus einem Bild der Größe 640×480 maximal fünf Ebenen erstellen, das Bild der höchsten Ebene hat dann die Größe von 20×15 Bildpunkten.

Die Verwendung des Maximum-Operators für die Erstellung eines Punktes einer höheren Ebene hat den Grund, dass dadurch aus einem Bildpunkt der Ebene L ersichtlich ist, dass in dieser Region, d.h. den Pixeln der Ebene $L-1$ und auch allen Ebenen darunter, keine Bildpunkte höheren Wertes befindlich sind. Dadurch kann in einer hohen Ebene eine Mindestabschätzung aller Hindernisse in diesem Bereich getroffen werden.

Die ansonsten übliche Verwendung von beispielsweise gewichteten Mittelwerten bei der Generierung von geringer aufgelösten Bildern (siehe Abschnitt 2.4.3 auf Seite 28) ist hierbei ungeeignet. Die Information, dass in einer Region alle Objekte im Durchschnitt eine bestimmte Entfernung besitzen, ermöglicht eben keine Aussage darüber, ob diese Region grundsätzlich von der Hindernislokalisation auszuschließen ist oder nicht.

5.2.2 Detektion von Hindernissen

Grundsätzlich stellt ein Punkt des Tiefenbildes ein Objekt dar, das sich in einer bestimmten Entfernung zum Betrachter befindet. Dabei muss festgestellt werden, ob dieser Punkt als Hindernis zu betrachten ist.

Einem Punkt des Tiefenbildes mit einem bestimmten Grauwert lässt sich nach Gleichung 4.2 (Seite 42) eine Entfernung im Kamerakoordinatensystem zuordnen. Darüber hinaus kann die Größe eines Objektes mit der Größe von einem Pixel Breite bzw. Höhe ermittelt werden. Ein Punkt des Tiefenbildes repräsentiert damit ein flaches Objekt, von dem nur die vordere

Fläche zu sehen ist. Von dieser Fläche sind allerdings Position, Höhe, Breite und Entfernung vom Betrachter bekannt.

Diese Fläche ist parallel zur xy-Ebene des Kamerakoordinatensystems. Die Ebene, in der sich die Fläche befindet, wird im Folgenden *Entfernungsebene* genannt. Alle Punkte des Tiefenbildes mit gleichem Wert, also gleicher Entfernung zum Betrachter, repräsentieren Objekte bzw. Flächen dieser Ebene.

Wie schon erwähnt, erfolgt die Feststellung, ob ein Punkt ein Hindernis darstellt oder nicht, mit Hilfe eines Sicherheitskorridors. Dies erfolgt mit Hilfe des 3D-Punktes in der Entfernungsebene und der Schnittmenge dieser Ebene mit dem Sicherheitskorridor.

5.2.2.1 Festlegung eines Sicherheitskorridors

Bezogen auf eine Achse, die eine näherungsweise Darstellung der aktuellen bzw. einer möglichen Flugbahn darstellt, sind Hindernisse all jene Objekte, die einen bestimmten Mindestabstand zu dieser Achse unterschreiten. Dadurch wird ein Zylinder um diese Achse aufgespannt – ein Sicherheitskorridor, durch die der Helikopter kollisionsfrei fliegen kann, sofern sich in ihm keine Objekte befinden. Alle Objekte außerhalb dieses Zylinders sind bezogen auf diese Flugbahn keine Hindernisse. Der Grundflächenradius des Zylinders muss je nach Größe des Helikopters und eines gewünschten Sicherheitsabstandes zu Hindernissen entsprechend festgelegt werden.

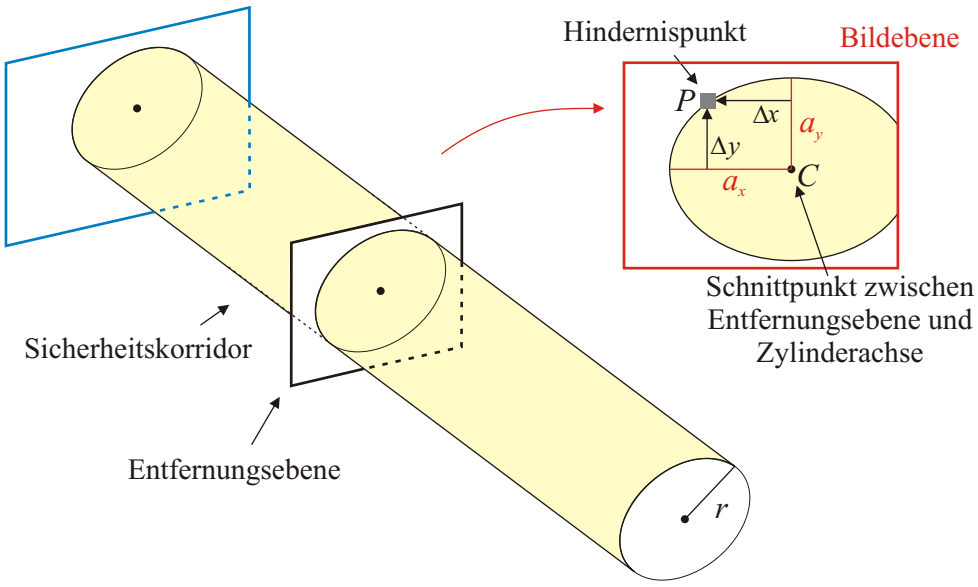


Abbildung 5.3: Sicherheitskorridor. Dieser besitzt einen kreisförmigen Querschnitt und geht vom Flugkörper aus in beliebige Richtungen. Der so entstehende Zylinder schneidet die Entfernungsebene allgemein in einer Ellipse. Diejenigen Bildpunkte dieser Entfernungsebene, die sich innerhalb dieser Ellipse befinden, stellen Objekte innerhalb des Korridors dar und sind bezogen auf diesen Korridor damit Hindernisse. Bei weiter entfernteren Ebenen (blau) ist die Ellipse in Bildkoordinaten kleiner.

Schneidet der Zylinder mit Grundflächenradius r eine Entfernungsebene senkrecht, so ergibt sich auf dieser Ebene ein „Sicherheitskreis“ mit dem Radius r. Schneiden sich die Ebene und die Achse nicht senkrecht, so wird der Kreis zu einer (größeren) Ellipse, deren Halbachsen hier mit a_x und a_y bezeichnet werden [Wre99].

Die Längen der Halbachsen a_x und a_y der Ellipse ergeben sich aus dem Radius des Sicherheitskorridors und den Schnittwinkeln:

$$a_x = \frac{r}{\cos \alpha_x}, \text{ bzw. } a_y = \frac{r}{\cos \alpha_y}. \tag{5.3}$$

Die Winkel ergeben sich aus der Koordinate des Schnittpunktes $C(x, y, z)$ der Zylinderachse mit der Entfernungsebene. Es ist

$$\alpha_x = \text{atan2}(z, x), \text{ bzw. } \alpha_y = \text{atan2}(z, y). \tag{5.4}$$

Die Funktion $\text{atan2}(z, x)$ ist wie folgt definiert:

$$\text{atan2}(z, x) = \begin{cases} \tan^{-1}(\frac{x}{z}), & \text{falls } z > 0; \\ \tan^{-1}(\frac{x}{z}) + 180°, & \text{falls } z < 0 \wedge x \geq 0; \\ \tan^{-1}(\frac{x}{z}) - 180°, & \text{falls } z < 0 \wedge x < 0; \\ 90°, & \text{falls } z = 0 \wedge x > 0; \\ -90°, & \text{falls } z = 0 \wedge x < 0; \\ \text{nicht definiert}, & \text{falls } z = x = 0. \end{cases} \tag{5.5}$$

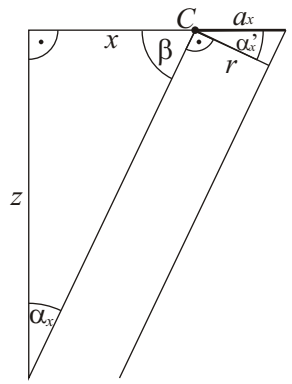


Abbildung 5.4: Berechnung des Schnittwinkels, Draufsicht von oben. Es ist $\alpha_x + 90° + \beta = 180°$ (Innenwinkelsumme des Dreiecks) und $\alpha_x' + 90° + \beta = 180°$ (Nebenwinkel), es folgt $\alpha_x + \beta + 90° = \alpha_x' + \beta + 90°$ und somit $\alpha_x = \alpha_x'$. Im rechtwinkligen Dreieck gilt für die Ankathete r: $\cos \alpha_x = a_x / r$, woraus Gleichung 5.3 folgt. Für den zweiten Winkel α_y in der anderen Dimension analog.

Der Wert eines Bildpunktes gibt an, in welcher Entfernungsebene sich dieser befindet. Für diese Entfernung ist sowohl die Größe eines Pixels bekannt, als auch der Schnittpunkt zwischen Entfernungsebene und Zylinderachse. Bei der Untersuchung der frontalen Entfernung eines Hindernisses entspricht die Zylinderachse der x-Achse des Helikopterkoordinatensystems. Bei

der Untersuchung eines alternativen Zielpunktes definiert dieser Zielpunkt den Schnittpunkte zwischen Zylinderachse und Entfernungsebene – dessen Bildkoordinaten sind in allen Entfernungsebenen gleich.

Der zu untersuchende Punkt mit den Abständen Δx und Δy vom Zentrum der Ellipse in Kamerakoordinaten liegt innerhalb der selben, wenn

$$\frac{\Delta x^2}{a_x^2} + \frac{\Delta y^2}{a_y^2} \leq 1 \tag{5.6}$$

nach der Ellipsengleichung [RW97, S. 81] gilt.

5.2.2.2 Entfernungsmessung im Tiefenbild

Zur flugbahnbezogenen Hindernisfindung ist die kleinste Entfernung eines Objektes gesucht, das sich innerhalb des entsprechenden Zylinders befindet. Bezogen auf das Tiefenbild ist dies ein Punkt mit größtem Grauwert, dessen Kamerakoordinaten einen Punkt innerhalb des Zylinders darstellen. Ist ein Startwert für diesen Grauwert gefunden, z.B. der Bildpunkt auf dem Mittelpunkt des Kreises bzw. der Ellipse, ist die Entfernung von Hindernissen innerhalb dieses Korridors kleiner oder gleich der diesem Bildpunkt entprechenden Entfernung.

Pixel des Tiefenbildes mit geringerem Grauwert sind damit uninteressant. Sie stellen Objekte mit größerer Entfernung als ein bereits gefundenes Hindernis dar und müssen nicht weiter betrachtet werden.

Die Implementation ist folgendermaßen aufgebaut: Für einen Bildpunkt, der ein potentielles Hindernis darstellt, wird innerhalb der Entfernungsebene dieses Punktes festgestellt, ob es sich tatsächlich um ein Hindernis handelt. Für diese Ebene wird der Schnittpunkt mit der Zylinderachse berechnet. Liegt der Punkt innerhalb der Ellipse, handelt es sich um ein Hindernis innerhalb des Flugbereichs (Zylinders). Allzu große Entfernungen werden mit Hilfe eines Schwellwertes ignoriert.

Für eine schnelle Verarbeitung wird zunächst das Bild mit der geringsten Auflösung verwendet. Sind Punkte dieses Bildes komplett außerhalb des Sicherheitskorridors, kann es auch in dieser Bildregion des am höchsten aufgelösten Bildes keinen Hindernispunkt geben. Sind Punkte komplett innerhalb des Bereiches, ist ebenso ersichtlich, dass sich im Bereich ein Hindernis befinden muss – eine Betrachtung höher aufgelöster Unterregionen ist auch hier nicht erforderlich. Schneidet die Begrenzung des Zylinders die Bildregion, müssen höher aufgelöste Bilder, also tiefere Pyramidenebenen, untersucht werden. Das Verfahren setzt voraus, dass sich die Kamera innerhalb des Sicherheitskorridors befindet.

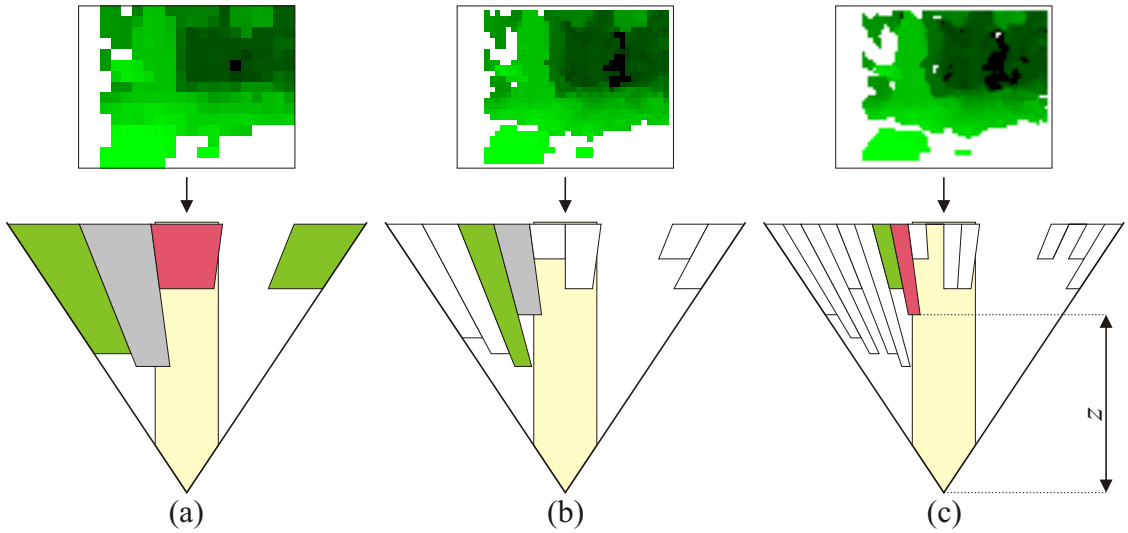


Abbildung 5.5: Pyramidenbasierte Erkennung von Hindernispunkten. In einem gering aufgelösten Tiefenbild (a) repräsentieren die Bildpunkte Flächen einer bestimmten Größe. Liegen diese Flächen vollständig innerhalb (rot) oder außerhalb (grün) des Sicherheitskorridors (gelb), so ist es eindeutig, ob es sich um ein Hindernis handelt. Schneidet der Korridor die Flächen (grau), wird das Hindernis in einer tieferen Pyramidenebene (b) gesucht. Bei der untersten Ebene (c) sind Flächen nicht mehr unterteilbar, deren Mittelpunkt bestimmt, ob die dargestellten Objekte innerhalb oder außerhalb des Sicherheitsbereiches liegen. Mit z ist hier die geringste Entfernung zu einem Hindernis dieses Korridors gekennzeichnet.

Bei der Überprüfung der Punkte der Pyramidenebene L, d.h. der Flächen, die die Punkte darstellen, geschieht folgendes:

- Liegt ein Flächenelement der Ebene L vollständig außerhalb des Korridors, wird es ignoriert.
- Liegt ein Element der Ebene L teilweise innerhalb des Korridors, werden alle vier Teilflächen der Ebene $L-1$, aus der sich das Element zusammensetzt, überprüft.
- Liegt ein Element der Ebene L vollständig innerhalb des Korridors, wird die Entfernung die der Disparität zugeordnet ist, als Objektentfernung ausgegeben.

Punkte der Ebene 0 können nicht weiter unterteilt werden und liegen entweder vollständig innerhalb oder außerhalb.

Abbildung 5.5 verdeutlicht die Suche nach Hindernissen in der Pyramide des Tiefenbildes. Dabei wird für jeden Bildpunkt, d.h. jede Objektfläche nur die jeweilige Entfernungsebene betrachtet. Gesucht ist damit, ob eine Fläche innerhalb der Ellipse liegt, die die Schnittmenge zwischen Entfernungsebene und Sicherheitskorridor bildet.

Für die Feststellung, ob eine Fläche wie bei der Darstellung in Abbildung 5.6 innerhalb oder außerhalb der Ellipse liegt bzw. die Ellipse schneidet, genügt die Messung des Abstandes des

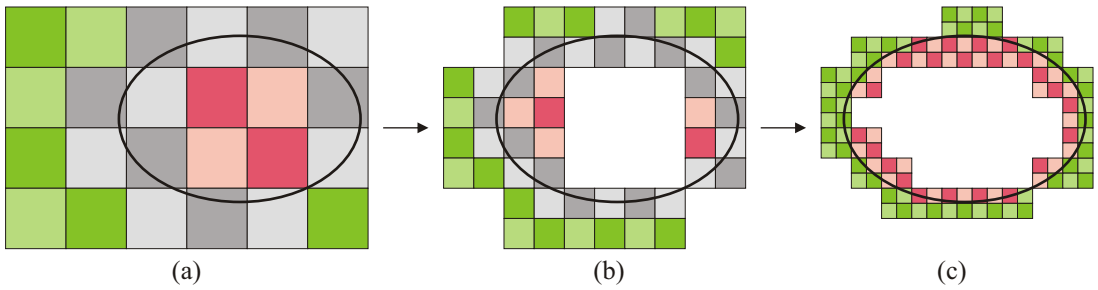


Abbildung 5.6: Erkennung von Hindernissen innerhalb einer Entfernungsebene. In der Darstellung besitzen alle Bildpunkte den gleichen Wert, repräsentieren also Objekte in gleicher Entfernung und befinden sich somit in dieser Ebene. Es ist in (a) für alle grün gekennzeichneten Punkte eindeutig, dass sie außerhalb der Ellipse liegen. Alle roten Punkte liegen eindeutig innerhalb. Bei den grauen Punkten muss eine höher aufgelöste Pyramidenebene (b) verwendet werden. In der untersten Pyramidenebene (c) ist die Zugehörigkeit durch Verwendung des Flächenmittelpunkts eindeutig.

Flächenmittelpunkts zur Ellipsenmitte nicht. Stattdessen werden Punkte der Fläche betrachtet, die am nächsten oder am weitesten entfernt vom Ellipsenmittelpunkt liegen. Abbildung 5.7 veranschaulicht dies.

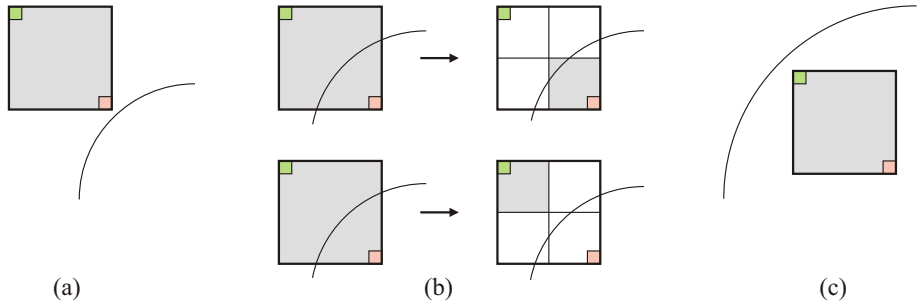


Abbildung 5.7: Unterteilung von Regionen, die als Punkte in einer Bildpyramide dargestellt sind. Sind der nächste (rot) und damit auch der entfernteste Punkt (grün) außerhalb der Ellipse, die den Sicherheitsabstand darstellt, ist kein Hindernis vorhanden (a). Liegt der nächste Punkt innerhalb, kann erst in tieferen Ebenen festgestellt werden, ob sich ein Hindernis dort befindet (b, oben) oder eben nicht (b, unten). Liegt allerdings der entfernteste Punkt ebenfalls innerhalb, so existiert in jedem Falle ein Hindernispunkt, der in diesem Bereich liegt (c).

Dazu müssen für jeden Punkt einer höheren Pyramidenebene, d.h. jede Fläche von Tiefenbildpunkten, jene Punkte des hochaufgelösten Tiefenbildes ermittelt werden, die sich innerhalb dieser Fläche am nächsten oder am weitesten entfernt vom Ellipsenmittelpunkt befinden. Bei der Errechnung der Distanz (in Pixeln) eines Bildpunktes $C(u_c, v_c)$ (der Ebene 0) zu einem Bildpunkt der Ebene L, also einer quadratischen Region der Seitenlänge 2^L in Ebene 0, wird jeweils die Distanz zum nächstgelegenen Punkt der Ebene 0 $P(u_p, v_p)$ innerhalb dieser Region berechnet. Die Regionen haben als Einzelpunkte in der Ebene L die Koordinaten $(u_{c,L}, v_{c,L})$ und $(u_{p,L}, v_{p,L})$ – in Ebene 0 entsprechen die oberen linken Ecken $(u_{c,0}, v_{c,0})$ den Koordinaten $u_{c,0} = 2^L \cdot u_{c,L}$, $v_{c,0} = 2^L \cdot v_{c,L}$, usw. .

Liegt beispielsweise die Region oben-rechts vom anderen Messpunkt, ist der nächstgelegene Punkt der der unten-links liegenden Ecke dieser Region. Liegt die Region auf der gleichen Zeile und rechts davon, so ist es der linke Punkt der Region, der auf der gleichen Zeile wie der Vergleichspunkt liegt (Abbildung 5.8).

Die Koordinaten der Punkte der Region, die dem Punkt C in Ebene 0 am nächsten sind (in Abbildung 5.7 und 5.8 rot dargestellt) lassen sich wie folgt ermitteln:

$$u_{p,0} = \begin{cases} 2^L \cdot u_{p,L} + 2^L - 1, & \text{falls } u_{p,L} < u_{c,L}; \\ u_{c,0}, & \text{falls } u_{p,L} = u_{c,L}; \\ 2^L \cdot u_{p,L}, & \text{falls } u_{p,L} > u_{c,L}, \end{cases} \tag{5.7}$$

bzw.

$$v_{p,0} = \begin{cases} 2^L \cdot v_{p,L} + 2^L - 1, & \text{falls } v_{p,L} < v_{c,L}; \\ v_{c,0}, & \text{falls } v_{p,L} = v_{c,L}; \\ 2^L \cdot v_{p,L}, & \text{falls } v_{p,L} > v_{c,L}. \end{cases} \tag{5.8}$$

Die Koordinaten der Punkte, welche vom Punkt C am weitesten entfernt sind, liegen stets in den Ecken der Regionen. Ist die Region von Punkt P linksoben von C, so ist die linke obere Ecke am weitesten entfernt, für die anderen drei Ecken analog. Ist eine Koordinate jedoch gleich, muss die Ecke genommen, werden, die zu Punkt C (in Ebene 0) den größten Abstand besitzt, das ist z.B. eine untere Ecke, falls C in der oberen Hälfte der Region liegt. Die Differenz zwischen der oberen Kante der Region und der v-Koordinate des Punktes C ist dort kleiner als die halbe Ausdehnung der Region.

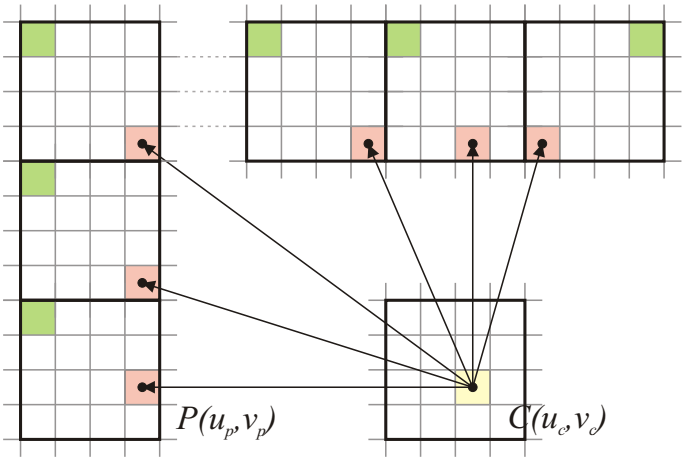


Abbildung 5.8: Berechnung der Distanz zu Regionen in einem geringer aufgelösten Bild der Bildpyramide. Liegt die aus mehreren Pixeln bestehende Region beispielsweise links-oben vom gesuchten Punkt, wird die Distanz zum Pixel in der rechten unteren Ecke verwendet, bei anderen Ecken analog (rot). Ist eine Koordinate gleich, erfolgt die Distanzmessung über die Berechnung der anderen Koordinate, d.h. liegt die Region links vom Ausgangspunkt (v-Abstand = 0), wird die u-Koordinate eines rechten Pixels zur Messung benutzt (rot).

Diese Ecken sind in Abbildung 5.7 bzw. 5.8 grün dargestellt und lassen sich wie folgt ermitteln:

$$u_{p,0} = \begin{cases} 2^L \cdot u_{p,L}, & \text{falls } u_{p,L} < u_{c,L} \vee (u_{p,L} = u_{c,L} \wedge u_{c,L} \cdot 2^L - u_{c,0} < 2^{L-1}); \\ 2^L \cdot u_{p,L} + 2^L - 1, & \text{sonst,} \end{cases} \tag{5.9}$$

bzw.

$$v_{p,0} = \begin{cases} 2^L \cdot v_{p,L}, & \text{falls } v_{p,L} < v_{c,L} \vee (v_{p,L} = v_{c,L} \wedge v_{c,L} \cdot 2^L - v_{c,0} < 2^{L-1}); \\ 2^L \cdot v_{p,L} + 2^L - 1, & \text{sonst.} \end{cases} \tag{5.10}$$

Als Distanzen in Bildkoordinaten ergeben sich

$$\Delta u = |u_{c,0} - u_{p,0}| \quad \text{, bzw.} \quad \Delta v = |v_{c,0} - v_{p,0}|. \tag{5.11}$$

Der reale Abstand zwischen den Punkten P und C ergibt sich aus der Größe, d.h. Höhe und Breite eines Bildpunktes. Der variiert je nach Entfernungsebene. Zur Ermittlung der Distanz wird diejenige Entfernungsebene verwendet, in der sich Punkt P befindet. Mit dieser Größe lässt sich die reale Entfernung bestimmen.

Die Distanzen in Kamerakoordinaten sind

$$\Delta x = \Delta u \cdot \frac{z}{f} \quad \text{, bzw.} \quad \Delta y = \Delta v \cdot \frac{z}{f}; \tag{5.12}$$

mit der Breite eines Pixels, die $\frac{z}{f}$ durch Einsetzen des Wertes 1 für $(x - x_0)$ in Gleichung 4.3 (Seite 43) beträgt. Es bezeichnen z die aus dem Grauwert des Pixels $g_{16}(u_{p,0}, v_{p,0})$ resultierende Entfernung und f die Brennweite in Pixeln. Zur Verbesserung der Laufzeit werden die Zuordnungen zwischen dem Wert eines Punktes im Tiefenbild, der davon abhängigen Entfernung und der Größe eines 1 Pixel großen Objektes dieser Entfernung in Lookup-Tabellen festgehalten.

Somit lässt sich überprüfen, ob der nächste bzw. der entfernteste Punkt einer Region einen bestimmten Abstand zum Schnittpunkt mit der Achse besitzt – also ob ein Punkt einer höheren Pyramidenebene vollständig außerhalb, teilweise innerhalb oder vollständig innerhalb des Sicherheitsbereichs liegt.

Auf diese Art wird für alle Punkte des Tiefenbildes untersucht, ob sie Hindernisse innerhalb des festgelegten Korridors darstellen. Die geringste Entfernung eines Objektes innerhalb dieses Korridors wird somit als Entfernung ausgegeben. Praktisch bedeutet dies, dass sich innerhalb des Korridors das nächste Hindernis in dieser Entfernung befindet. Entlang der Achse des Zylinders ist ein kollisionsfreier Flug bis hin zu dieser Entfernung möglich.

5.2.3 Frontale Entfernung

Nach oben beschriebenem Verfahren wird zunächst die kleinste Entfernung gemessen, die ein Objekt besitzt, das sich direkt vor dem Flugkörper befindet. Ist kein Hindernis in diesem Bereich, so ist ein problemloses Weiterfliegen möglich. Wird eine nahe Entfernung gemessen, so muss gebremst oder der Kurs geändert werden.

Dazu wird die Entfernung der Hindernisse gemessen, die innerhalb eines Sicherheitsabstandes zur x-Achse des Helikopterkoordinatensystems liegen. Berechnet wird die Entfernungsebene, in der sich das nächste Hindernis innerhalb des Korridors befindet, Ausgabe ist die Entfernung x_f des Ursprungs S dieser Ebene in Helikopterkoordinaten als Näherung für die tatsächliche Entfernung eines Hindernisses. In Abbildung 5.9 ist das Kamerakoordinatensystem um $\varphi = 5°$ gedreht, bei einem Radius des Sicherheitskorridors von $r = 2$ m ist die maximale Ungenauigkeit daher $\Delta x = \tan\varphi \cdot r \approx 0,17$ m, unabhängig von der absoluten Hindernisentfernung. In der Praxis sind die Rotationswinkel geringer und damit auch Δx. Dieser Wert wird bei Hindernissen am Rand des Korridors (T) erreicht und ist bei großen Entfernungen weitaus geringer als die Entfernungsauflösung der Disparitätenschätzung.

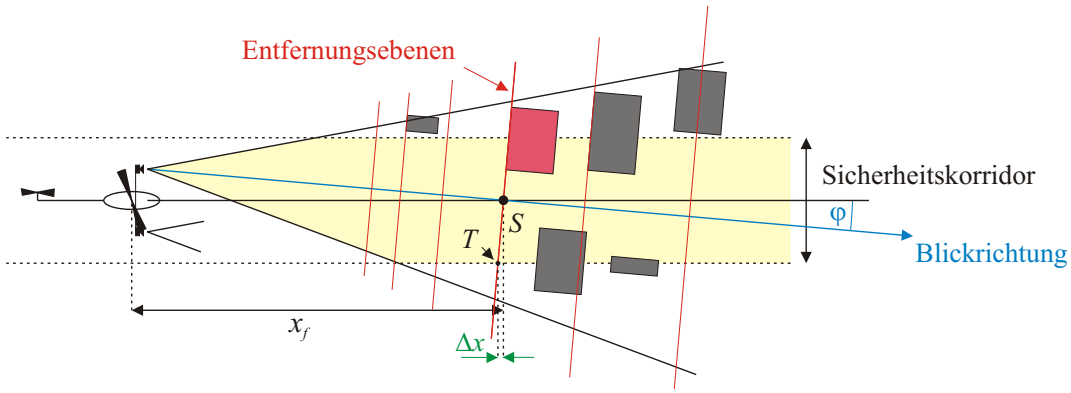


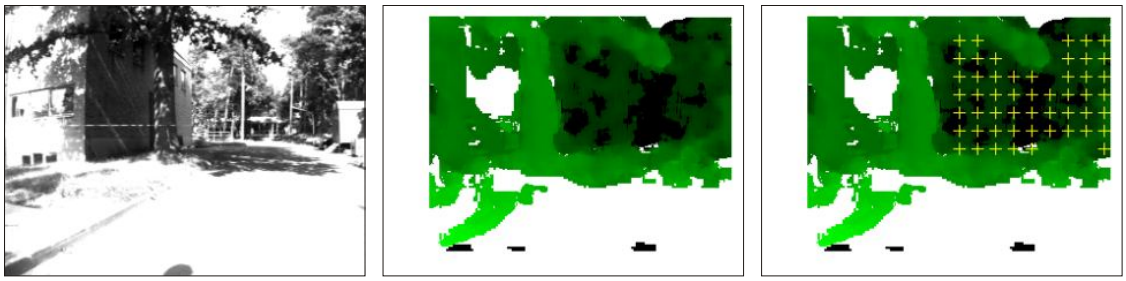
Abbildung 5.9: Messung der frontalen Entfernung zum Flugobjekt (Draufsicht). Alle Objekte, die außerhalb des Sicherheitsbereichs (gelb) liegen, sind uninteressant. Die geringste Entfernung x_f zu einem Objekt innerhalb des Korridors ist die Distanz, bis zu der ein kollisionsfreier Flug in dieser Richtung möglich ist. Sie ist damit relevant für die Entscheidung, ob die Flugrichtung beibehalten werden kann.

Anwendung für die Messung der frontalen Entfernung ist eine ständige Kontrolle, ob der eingeschlagene Kurs auch kollisionsfrei ist. Teile des Sicherheitskorridores können bei geringen Entfernungen außerhalb des sichtbaren Bereiches liegen, wodurch dort liegende Hindernisse nicht erkannt werden. Bei Änderung des Kurses und damit auch der Sicht können diese dann erkannt werden. Dazu kann je nach Abstand zum nächsten Hindernis eine Geschwindigkeitsempfehlung gegeben oder im Zweifelsfall auch angehalten werden.

5.2.4 Finden von Ausweichmöglichkeiten

Die Suche nach Ausweichmöglichkeiten ist zunächst durch das Finden von Bildabschnitten gekennzeichnet, in denen sich keine Hindernisse befinden. Im Disparitätenbild sind das all jene, die dunkel und möglichst weit von hellen Bereichen entfernt sind.

Gesucht werden diese Bereiche über die schon vorhandene Bildpyramide. In der höchsten Ebene werden die Punkte der Helligkeit nach sortiert. Eine bestimmte Anzahl der dunkelsten Bereiche wird dann näher betrachtet.



(a) (b) (c)

Abbildung 5.10: Linkes Bild eines Stereobildpaares (a) und zugehöriges Tiefenbild (b). In Bild (c) sind alle dunklen Bereiche und damit mögliche Ziele gelb markiert. Für jeden Markierungspunkt wird die Entfernung zu Hindernissen in einem Sicherheitsbereich um diesen Punkt bestimmt.

Zunächst wird die dunkelste Stelle innerhalb eines jeden Bereiches gesucht, die Punkte sind in Abbildung 5.10 gelb dargestellt. Für jeden Punkt wird nun der Abstand zum Helikopter mit Hilfe des Sicherheitskorridors bestimmt. In diesem Fall wird ein Zylinder betrachtet, dessen Achse im Gegensatz zur frontalen Entfernungsmessung das Kamerazentrum schneidet. Vorteilhaft ist die schnellere Erkennung von Hindernissen in der Entfernungsebene, da der Schnittpunkt in jeder Ebene die gleichen Bildkoordinaten besitzt. Da die Kamera nur gering zum Ursprung des Helikopterkoordinatensystems verschoben ist, kann dieser Korridor als gute Näherung zu einem Korridor betrachtet werden, der durch das Helikopterzentrum und einen weit entfernten Zielpunkt geht.

Wie in Abbildung 5.11 dargestellt, ist die Ausgabe der Punkt $S(x_f, y_f, z_f)$ im helikopterfesten Koordinatensystem. Bis zu diesem Punkt kann ein kollisionsfreier Flug erfolgen. Es kann in den Filtereinstellungen festgelegt werden, wie viele Punkte maximal zur Messung der Entfernung letztendlich herangezogen werden und damit mögliche Zielpunkte darstellen.

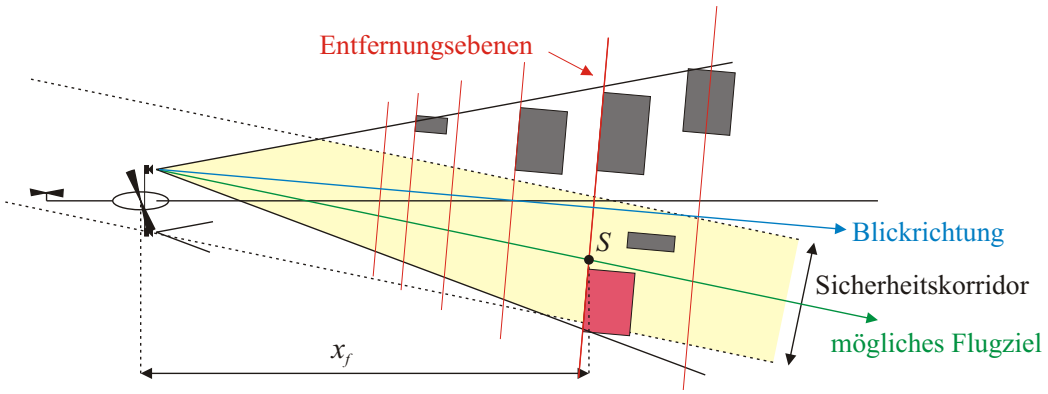


Abbildung 5.11: Messung von Entfernungen zu alternativen Zielpunkten. Zu bestimmten Bildpunkten werden die Entfernungen gemessen, die innerhalb des Sicherheitskorridors liegen, bzw. im Bild innerhalb der Ellipse, die die Entfernungsebene und den zylinderförmigen Korridor schneidet. Helle Punkte im Disparitätenbild werden von vornherein ausgeschlossen, da dort in jedem Fall Hindernisse zu erwarten sind.

5.2.5 Kosten und Auswahl eines Ziels

Der Algorithmus zum Finden von Ausweichmöglichkeiten liefert lediglich alternative Zielpunkte und deren Koordinaten. Hier ist eine Möglichkeit angegeben, aus dieser Menge von Zielpunkten, deren Helikopterkoordinaten bekannt sind, den besten Punkt herauszusuchen.

Für die Auswahl eines geeigneten Ziels aus den gefundenen Punkten wird folgendes verlangt:

- Die Entfernung zu einem Hindernis innerhalb des Korridors soll möglichst hoch sein. Das bedeutet, Regionen mit nahen Hindernissen sollen keine Ziele darstellen.
- Es soll so wenig wie möglich gesteuert werden. Sind mehrere Regionen ohne bzw. nur mit entfernten Hindernissen erkannt worden, soll diejenige ausgewählt werden, die am wenigsten Bahnabweichungen bzw. Lenkmanöver nach sich zieht.
- Die Möglichkeit einer Richtungsanweisung kann gegeben werden, um eine Steuerung in eine bestimmte Richtung zu erzielen.

Modelliert wird dies durch eine Kostenfunktion, die für jeden der gefundenen Zielpunkte $S(x, y, z)$ einen Wert berechnet. Abhängig ist dieser von der Entfernung zum Hindernis und einer Abweichung zu einem gewünschten Wegpunkt. Die Linie zwischen Helikopter und Wegpunkt, d.h. eine gewünschte Flugbahn, sei hier in Form von Winkeln in horizontaler (γ_y) bzw. vertikaler Richtung (γ_z) angegeben. Beim gewünschten Geradeausflug sind diese Winkel gleich null. Die Möglichkeit zur Angabe von Winkeln lässt eine Bevorzugung von anderen Richtungen zu.

Eine einfache Möglichkeit ist ein lineares Kostenmodell mit der Kostenfunktion

$$K(x, \Delta\alpha_y, \Delta\alpha_z) = -k_x \cdot x + k_y \cdot \Delta\alpha_y + k_z \cdot \Delta\alpha_z. \tag{5.13}$$

Dabei bezeichnen k_x, k_y und k_z die Gewichtungen für die einzelnen Kostenfaktoren x-Entfernung als Länge und y- bzw. z-Abweichung in Winkeln vom Idealwinkel, der durch eine Zielkoordinate gegeben ist. Diese Abweichungen sind

$$\Delta\alpha_y = |\text{atan2}(x, y) - \gamma_y| \quad \text{, bzw.} \quad \Delta\alpha_z = |\text{atan2}(x, z) - \gamma_z|. \tag{5.14}$$

mit der Funktion atan2 nach Gleichung 5.5 (Seite 71).

Die negative Gewichtung der Entfernung x bevorzugt weit entfernte Bereiche. Bei gleicher Entfernung haben Punkte mit geringen Bahnänderungen geringere Kosten. Die Möglichkeit, Bahnabweichungen in horizontaler und vertikaler Richtung unterschiedlich zu werten, erlaubt beispielsweise eine Bevorzugung von vertikalem Ausweichen gegenüber horizontalem. Dies ist sinnvoll, da Flugkörper eine verschiedene Wendigkeit in der Nick- bzw. Gierachse besitzen.

Am Ende wird derjenige Punkt S als Ziel ausgewählt, der die geringsten Kosten besitzt.

5.2.6 Fixierung eines Punktes

Bei der Untersuchung von Tiefenbildsequenzen wurde festgestellt, dass die Koordinaten des vorgeschlagenen Zielpunktes zum Ausweichen zwischen zwei aufeinanderfolgenden Bildern sehr stark schwanken. Wird ein Punkt anvisiert und verfolgt (siehe Abschnitt 2.4.4, Seite 30), stellt der in den Folgebildern wiedergefundene Punkt das gleiche Objekt dar, wodurch es zu geringeren Schwankungen kommt.

Da die Texturierung des Tiefenbildes sehr schwach ist, lässt sich ein Punkt aufgrund des Blendenproblems schwieriger wiederfinden als im Originalbild. Daher wird ein gefundener Zielpunkt im linken Kamerabild verfolgt. Da das Kamerabild nicht entzerrt und rektifiziert ist, wird die Position des zu verfolgenden Punktes entsprechend umgerechnet. Mit Hilfe des Sicherheitskorridors wird ähnlich wie in Abschnitt 5.2.4 verifiziert, ob dieser verfolgte Punkt eine geeignete Kursänderung darstellt.

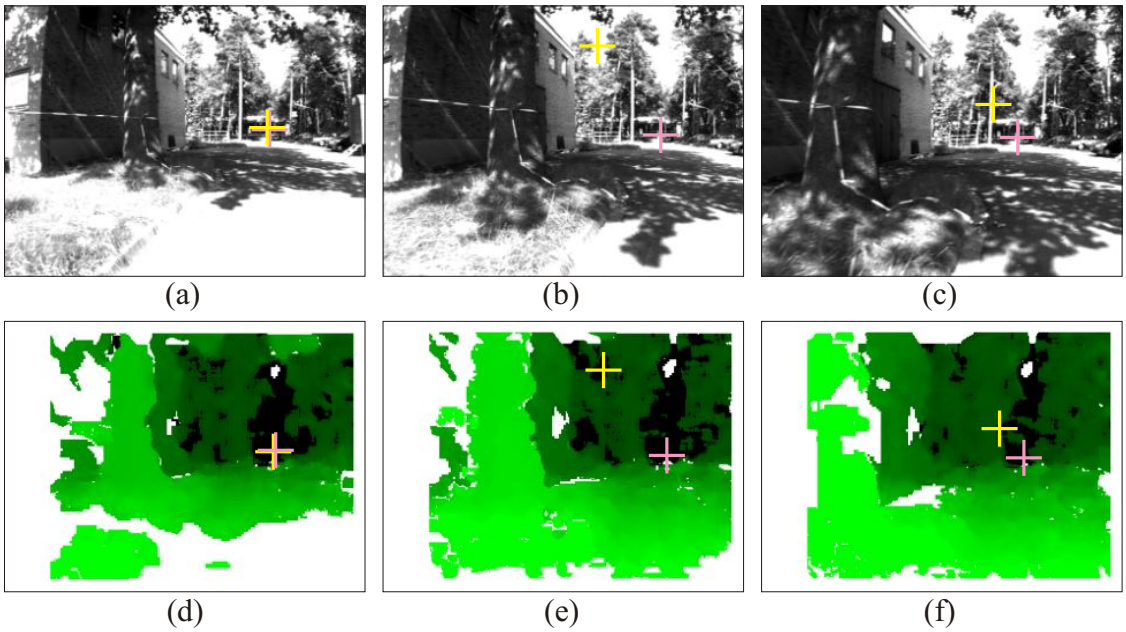


Abbildung 5.12: Aufeinanderfolgende Bilder (a-c) und die zugehörigen Tiefenbilder (d-f) dieser Sequenz. Der vorgeschlagene Ausweichpunkt (gelb) „springt“ hin und her. Der rot dargestellte Punkt wird im Originalbild verfolgt und springt dadurch nicht.

5.3 Steuerung des Helikopters

Die Analyse eines Tiefenbildes ermittelt die Distanz zu Hindernissen auf dem aktuellen Kurs und liefert Zielkoordinaten zum Ausweichen. Die Steuerungslogik entscheidet mit Hilfe dieser Informationen, welches Flugmanöver ausgeführt wird.

Aufgrund von Bildrauschen schwanken die Entfernungs- und Zielinformationen auch wenn sich die Kamera bzw. der Helikopter nicht bewegt. Um starke Ausreißer bei Steuerungsanweisungen zu ignorieren, erfolgt eine Glättung mit Hilfe des Medians über die letzten drei Werte.

Die für die Steuerung zum Zeitpunkt t verwendete frontale Entfernung x_{Front} ist

$$x_{\text{Front}} = \text{median}\{x_t, x_{t-1}, x_{t-2}\} \tag{5.15}$$

mit der zum jeweiligen Zeitpunkt t gemessenen frontalen Entfernung x_t.

Der alternative Zielpunkt $S_{\text{Alt}}(x, y, z)$ zum Ausweichen besitzt die Koordinaten eines der letzten drei möglichen Ziele. Der ausgewählte Punkt erfüllt die Bedingung

$$S_{\text{Alt}} = S \in \{S_t, S_{t-1}, S_{t-2}\} \,|\, x = \text{median}\{x_t, x_{t-1}, x_{t-2}\}. \tag{5.16}$$

Dabei ist $S_t = (x_t, y_t, z_t)$, sofern verfügbar, der verfolgte Punkt zum Zeitpunkt t, ansonsten der Punkt mit den geringsten Kosten.

Darüber hinaus wird ein Schwellwert x_{Lenkung} festgelegt, ab welcher Distanz zum Helikopter ein Hindernis als gefährlich anzusehen ist und ein Ausweich- bzw. Bremsmanöver folgen muss. Hindernisse in größerer Entfernung werden als ungefährlich eingestuft und es wird ermöglicht, auch in diese Richtung zu steuern. Das Flussdiagramm in Abbildung 5.13 verdeutlicht den schematischen Ablauf der Kollisionsvermeidung. Letztendlich werden Kommandos zum Steuern in eine bestimmte Richtung oder zum Bremsen an den Bordrechner des Helikopters gesendet.

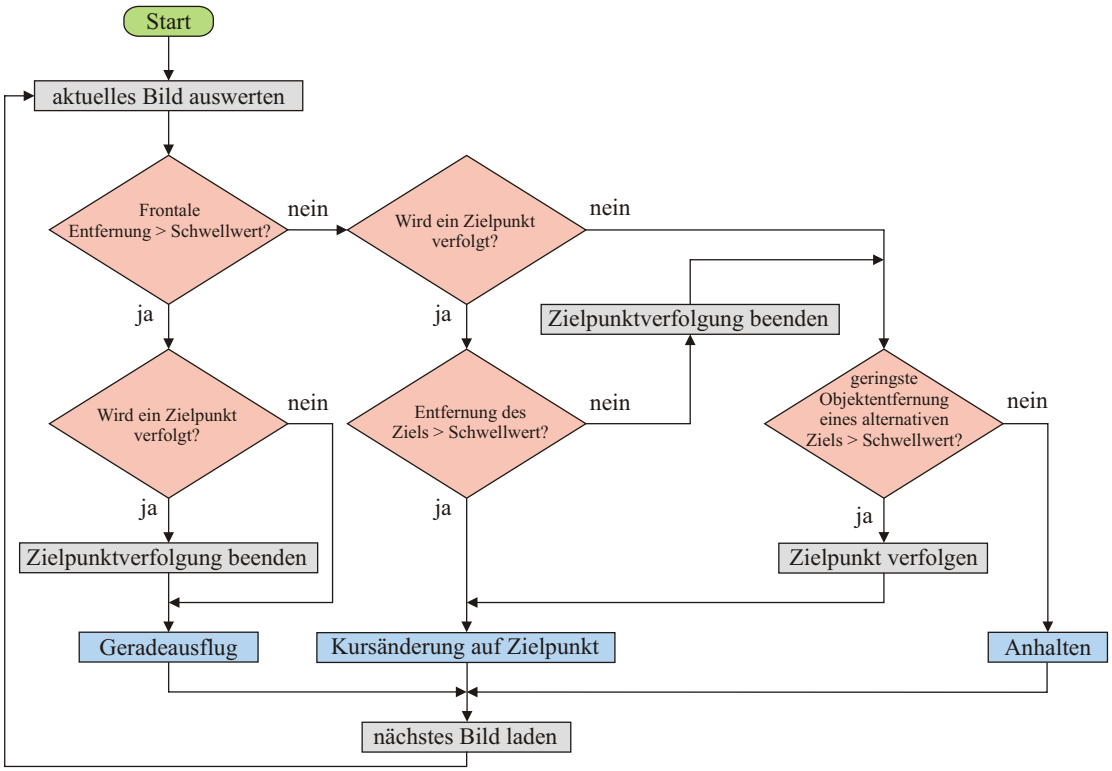


Abbildung 5.13: Flussdiagramm des Verfahrens zur Kollisionsvermeidung. Ist kein Hindernis auf dem aktuellen Kurs, wird dieser beibehalten. Andernfalls wird ein Ausweichkurs angesteuert, wobei bevorzugt der mediangefilterte verfolgte Punkt S_{Alt} das Ziel definiert. Befinden sich auf allen möglichen Ausweichkursen Hindernisse in zu geringer Entfernung, soll der Helikopter anhalten. Die Kommandos, welche an die Helikoptersteuerung weitergegeben werden, sind blau gekennzeichnet.

Der genannte Schwellwert x_{Lenkung} muss so gewählt werden, dass der Helikopter innerhalb dieser Distanz vollständig abbremsen kann. Ermittelt werden kann eine minimale Ausweichdistanz $\hat{x}_{\text{Lenkung}}$ über den Bremsweg bei einer klassischen Bewegungsgleichung[1]. Von Hrabar [Hra06] wird die vorgestellte Distanz als minimale Sichtweite verwendet, für die eindeutige

[1]Zur Sicherheit wird davon ausgegangen, dass innerhalb der Distanz zum Lenken noch vollständig gebremst werden kann. Unter bestimmten Voraussetzungen, wie dem ständigen Vorhandensein oder der Annahme von Ausweichmöglichkeiten, die nur wenige Kursänderungen nach sich ziehen, kann die Hindernisdistanz wesentlich geringer sein. Andernfalls wäre beispielsweise das Autofahren mit hohen Geschwindigkeiten unmöglich.

Klassifikation von hindernisfreien Bereichen zum Ausweichen muss die Sichtweite allerdings höher sein.

Es ist

$$\hat{x}_{\text{Lenkung}} = x + v_0(2t_c + t_a) + \frac{v_0^2}{2a} \tag{5.17}$$

mit dem Abstand zwischen Ursprung der Helikopterkoordinaten und der Spitze des Rotorkreises bzw. einem Sicherheitsabstand x, der Fluggeschwindigkeit v_0, Verzögerungszeiten durch die Datenverarbeitung t_c bzw. Trägheit der Flugsteuerung t_a und der Bremsbeschleunigung a. Abbildung 5.14 veranschaulicht den Zusammenhang zwischen Fluggeschwindigkeit und der minimalen Distanz von Hindernissen für kollisionsfreie Kursänderungen.

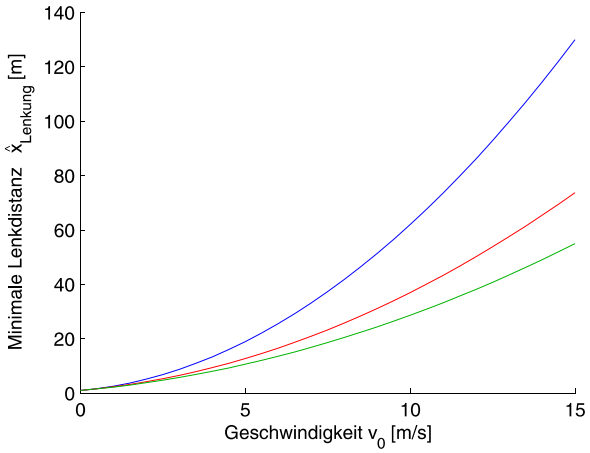


Abbildung 5.14: Minimale Ausweichdistanz $\hat{x}_{\text{Lenkung}}$ bei $x = 1\,m$ und den Beschleunigungswerten $a = 1\,m/s^2$ (blau), $a = 2\,m/s^2$ (rot) und $a = 3\,m/s^2$ (grün). Die Werte der Verzögerungszeiten seien mit $t_c = 0,3\,s$ und $t_a = 0,5\,s$ realistische Werte für einen Helikopterflug.

5.4 Einbindung des Filters in das Framework

Das Verfahren zur Kollisionsvermeidung wurde ebenfalls als Filter für das dip-Framework implementiert. Das Disparitätenfilter ist darin vollständig enthalten. Eingang sind wieder zwei Kamerabilder, als Standardausgabe wird das vom Disparitätenfilter erstellte 8-Bit-Grauwertbild verwendet, welches selbst unangetastet bleibt.

Die Berechnung von Entfernungen und Ausweichmöglichkeiten erfolgt aus dem 16-Bit-Disparitätenbild und einigen Metainformationen wie Pixelgrößen, der Beziehung zwischen Grauwert und Entfernung und der Kameraorientierung. Der Auswahlpunkt *„Collision Avoidance“* ruft dieses Filter auf.

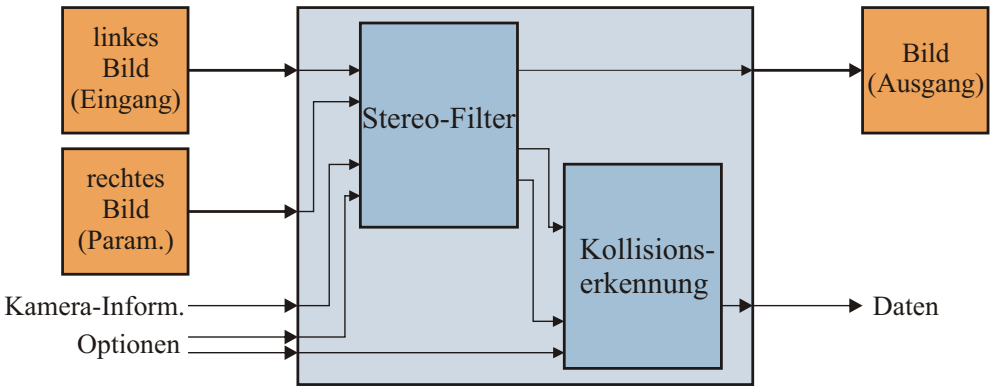


Abbildung 5.15: Aufbau des Filters zur Kollisionsvermeidung. Er enthält das Stereofilter. Das 8-Bit-Disparitätenbild des Stereofilters wird zur Anzeige ausgegeben, jedoch nicht verwendet. Die Kollisionsvermeidung verwendet das 16-Bit-Tiefenbild und Eigenschaften desselben wie der Zuordnung zwischen Disparität und Entfernung.

Erforderliche Eingaben sind zunächst die gleichen, die auch das Disparitätenfilter verwendet: linkes Bild, rechtes Bild und Kalibrierungsdatei. Zudem sollte sichergestellt werden, dass für die Kameras selbst Orientierungsinformationen verfügbar sind.

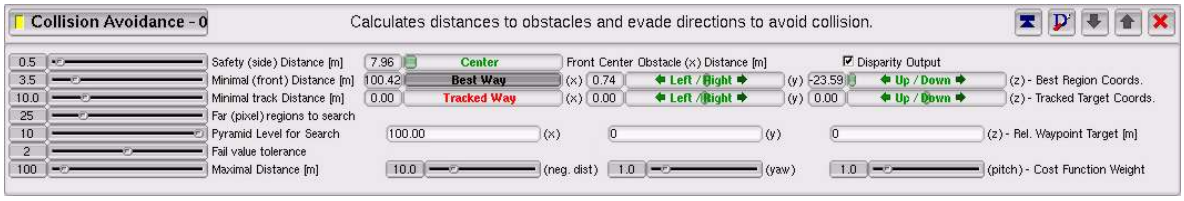


Abbildung 5.16: Die Benutzeroberfläche des Kollisionsfilters zur Festlegung der Parameter und zur Anzeige der Entfernungen.

Als Ausgabe wird kein neues Bild generiert – dieses Filter ist lediglich eine Bildanalyse. Ausgegeben werden die Entfernung des nächsten Hindernisses entlang der Flugkörperachse, die Position eines Punktes, bei dem am wenigsten Hindernisse zu erwarten sind und die Position eines verfolgten Punktes. Weiterhin können folgende Punkte visualisiert werden:

- Der Bildpunkt, der den Schnittpunkt der x_f-Achse und der Entfernungsebene mit der größtmöglichen Distanz darstellt,
- alle gefundenen dunklen Regionen, d.h. Zielpunkte, die auf Hindernisfreiheit überprüft werden,
- das Ziel mit den geringsten Kosten,
- der verfolgte Zielpunkt.

Das Filter lässt folgende Optionen zu:

- *Safety side distance:* Der Radius des Sicherheitskorridors. Prinzipiell sind beliebige positive Werte denkbar – sinnvoll sind solche, die größer als das Flugobjekt sind und noch einen Minimalabstand zu den Hindernissen zulassen.
- *Minimal front distance:* Minimaldistanz. Werden sehr nahe Punkte unterhalb dieses Wertes innerhalb eines Korridors gefunden, so bricht die Berechnung weiterer Punkte in diesem Korridor ab. Sinn und Zweck ist, dass Punkte unterhalb dieses Wertes generell als zu nah zu betrachten sind, so dass kein Ausweichen in diese Richtung möglich ist und bei frontaler Entfernung längst ein Bremsen oder Ausweichen hätte erfolgen müssen. Die Berechnung näherer Hindernisse bringt keinen Nutzen und würde Rechenleistung vergeuden.
- *Minimal track distance:* Minimaldistanz für die Verfolgung eines Zielpunktes. Nähere Distanzen kommen als Zielpunkt nicht in Frage – nur „sichere“ Punkte ohne nahe Hindernisse sollen verfolgt und ausgegeben werden, da diese Richtung geflogen werden soll. Darüber hinaus wird die Verfolgung eines Punktes abgebrochen, wenn die Distanz unter diesen Wert fällt.
- *Far pixel regions to search:* Anzahl der dunklen Regionen, bei denen die entsprechenden Korridore auf ihre Hindernisse überprüft werden. Kleine Anzahlen beschleunigen die Berechnung, lassen aber auch zu, dass bestimmte geeignete Ziele übersehen werden könnten.
- *Pyramid Level for Search:* Die Pyramidenebene, bei der die Suche nach dunklen Regionen begonnen wird.
- *Fail value tolerance:* Fehlertoleranz bei der Erzeugung der Bildpyramide. Existieren mehr Fehler als dort angegeben bei der Reduzierung von jeweils vier Pixeln zu einem, so wird dieser Punkt als Fehler klassifiziert.
- *Maximal Distance:* Maximaldistanz bei der Berechnung von Entfernungen, da deren Schätzung im weit entfernten Bereich nur sehr unzuverlässig ist. Werden keine Hindernisse gefunden, wird dieser Wert ausgegeben.
- *Relative Waypoint Target:* Angabe eines Wegpunktes (x, y, z) in Helikopterkoordinaten zur Steuerung des Helikopters in eine bestimmte Richtung, falls mehrere hindernisfreie Zielpunkte gefunden werden.
- *Cost Function Weights:* Wichtungsparameter für die Kostenfunktion bei der Bestimmung eines besten Zielpunktes. Gewichtet werden die Entfernung des Ziels *(negative distance)* und die horizontale *(yaw)* bzw. vertikale Lenkbewegung *(pitch)*.

Ausgegeben und auf der Benutzeroberfläche angezeigt werden:

- *Front Center Obstacle:* Die Distanz (x) zu Hindernissen direkt vor dem Helikopter.
- *Best Region Coordinates:* Der Zielpunkt (x, y, z) mit den geringsten Kosten.
- *Tracked Target Coordinates:* Der verfolgte Zielpunkt (x, y, z).

Alle Werte sind in helikopterfesten Koordinaten. Sie werden mit jedem neuen Bild aktualisiert und können von anderen Filtern und Anwendungen verwendet werden.

Für die Steuerung des Helikopters können folgende Einstellungen vorgenommen werden:

- *Steering Distance:* Die Maximaldistanz x_{Lenkung} für Lenkmanöver. Werden Hindernisse mit geringerer Entfernung erkannt, werden Ausweich- bzw. Bremsmanöver eingeleitet.
- *Send MCP*[2] *Message:* Ermöglicht das Senden von Steuerungsnachrichten an den Helikopter. Die Frequenz, mit der aktualisierte Kommandos gesendet werden, kann festgelegt werden. Sendevorgänge für Geradeausflug- bzw. Ausweichmanöver können separat gesteuert werden, beispielsweise um das Netzwerk beim Nichtvorhandensein von Hindernissen zu entlasten.

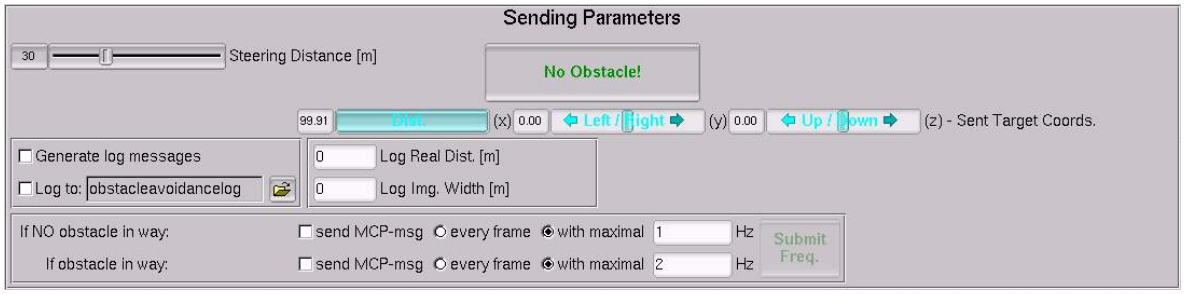


Abbildung 5.17: Die Benutzeroberfläche des Kollisionsfilters zur Steuerung des Helikopters. Angezeigt werden der aktuelle Status (kein Hindernis, Ausweichkurs, Bremsen) und die aktuell gesendete Zielkoordinate. Darüber hinaus ist die Speicherung der Daten möglich.

[2]Master Control Program. Flugsteuerungssoftware auf dem Bordrechner des Helikopters.

Kapitel 6

Test und Bewertung

6.1 Versuchsziel

Dieses Kapitel beschreibt die Erprobung der implementierten Kollisionsvermeidung in einer Simulationsumgebung. Ziel der Simulation ist der Test von Hard- und Software für die Eignung in der Realität, in diesem Fall im Helikopterflug. Die Verwendung gleicher Geräte und Schnittstellen, sowie die möglichst realitätsgetreue Modellierung von Flugeigenschaften ermöglicht die Untersuchung vieler Einzelkomponenten am Boden. Allerdings muss eine Überprüfung stattfinden, inwieweit die Simulationsergebnisse auf einen Flug übertragbar und was die Ursachen für eventuelle Unterschiede sind.

6.2 Aufbau der Simulationsumgebung

Es handelt sich um eine Hardware-in-the-Loop-Simulation, d.h. es es wird mit Rechnersystemen, Protokollen und Sensoren gearbeitet, die auch im realen Flug verwendet werden. Dabei bilden die Komponenten einen Regelkreis und beeinflussen sich gegenseitig. Abbildung 6.1 zeigt den Aufbau.

Die entwickelte Kollisionsvermeidung läuft auf einem Bildverarbeitungsrechner. Ausgabe ist ein Punkt im helikopterfesten Koordinatensystem als Flugziel. Die Koordinaten können einen Geradeausflug, einen Ausweichkurs oder ein Abbremsen auf die aktuelle Position bedeuten.

Wird ein Ziel an die Simulation übermittelt, so nimmt der Helikopter Kurs auf diese Koordinate. Diese Daten werden per *UDP*[1] an den Bordrechner des ARTIS-Helikopters geschickt, der auch bei Flugversuchen verwendet wird. In diesem Fall ist der Bordrechner allerdings nicht

[1] User Datagram Protocol. Dieses Netzwerkprotokoll ist verbindungslos, was eine hohe Geschwindigkeit ohne Kollisionen ermöglicht. Allerdings erfolgt im Gegensatz zu TCP (Transfer Control Protocol) keine Überprüfung der erfolgreichen Übertragung, wodurch es vorkommen kann, dass Datenpakete unbemerkt verloren gehen oder nicht in der korrekten Reihenfolge ankommen.

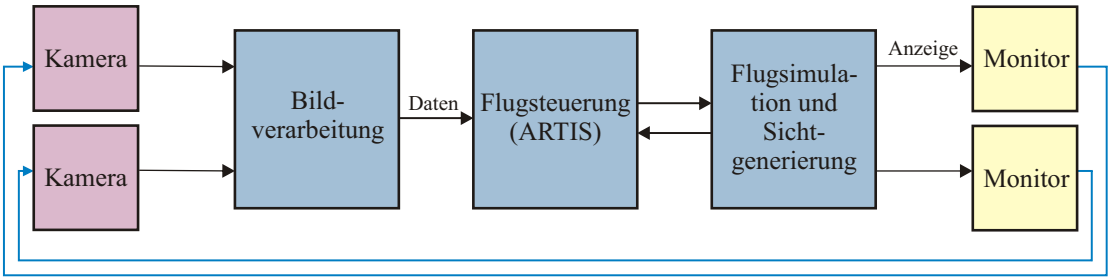


Abbildung 6.1: Schematischer Aufbau der Hardware-in-the-Loop-Simulation. Die Kollisionsvermeidung ist die Bildverarbeitung des dip-Frameworks. Die Flugsteuerung läuft auf dem ARTIS-Bordrechner. Simulation und Sichtgenerierung ist ein separates Rechnernetzwerk. Die Bildverarbeitung und die Flugsteuerung besitzen keinerlei Informationen darüber, ob es sich um einen Flug oder eine Simulation mit künstlich erzeugten Sensordaten handelt.

an Sensoren, sondern an die Simulationsumgebung angeschlossen. Dort wird die aus Messungen bekannte Flugdynamik des Helikopters berücksichtigt und es erfolgt eine Emulation der Sensoren, die der Bordrechner für einen Flug benötigt.

Die Visualisierung der Simulation erfolgt über eine eigene Sichtumgebung. Position und Lage des Helikopters sind bekannt, relativ dazu wird die Position und Lage von zwei Kameras angegeben. Es werden zwei Sichten generiert und auf Bildschirmen angezeigt, die jeweils den Blickpunkt einer Kamera darstellen. Bezogen auf den Helikopter kann jede Sicht beliebig rotiert oder verschoben werden. Die beiden Monitorbilder werden von Kameras abgefilmt und deren Bilder für die Stereobildverarbeitung verwendet. Die Bildschirme sind an eine gemeinsame Grafikkarte angeschlossen, was eine gleichzeitige Bildgenerierung garantiert. Die Synchronisation der Kameras erfolgt über den FireWire-Bus.

Die Kameras sind so über den Bildschirmen platziert, dass sie bei gleichem Bildschirminhalt möglichst ähnliche Bilder liefern. Ermöglicht wird dies durch die Messung der Höhe der Kamera über dem Bildschirm und der manuellen Justierung der Kameras. Die Gestelle erlauben die Montage der Kameras in beliebiger Position, siehe Abbildung 6.2. Die Ähnlichkeit der Kamerabilder lässt sich mit Differenzbildern feststellen. Folgende Einstellungen werden verwendet:

- Bildschirme: Auflösung 1600×1200 Bildpunkte, 60 Hz Bildwiederholrate,
- Kameras: Auflösung 640×480 Bildpunkte, 15 Hz Bildwiederholrate.

Die Sichten werden so generiert, dass deren Blick in Richtung der x_f-Achse des Helikopters zeigt. Der Ursprung der linken Sicht entspricht dem Ursprung des Helikopterkoordinatensystems, der Ursprung der rechten Sicht ist zu diesem Punkt um 0,3 m in y_f-Richtung verschoben.

Bei der Erzeugung von Bildern durch 3D-Rendering und deren Aufzeichnung existieren allerdings einige Probleme, die bei der Aufzeichnung realer Szenen nicht vorkommen:

Abbildung 6.2: Laboraufbau der Sichtsimulation. Es erfolgt eine Anzeige von zwei Sichten auf LCD-Flachbildschirme. Die Kameras sind so positioniert, dass der Rand des Bildschirme nicht zu sehen ist und die Bilder nicht verwackeln. Schutz vor Außenlicht wird über dunkle Abdeckungen realisiert, die hier nicht zu sehen sind.

- Da die aufzuzeichnenden Bilder diskretisiert sind, treten Rundungen und damit Aliasing-Effekte an den Kanten von Objekten auf. Der Vergleich zwischen den aufgezeichneten Kanten wird durch die Treppenabstufung ungenauer.

- Auch exakt horizontale oder vertikale Kanten sind auf den nächsten Bildpunkt gerundet. Bei den verwendeten Auflösungen nehmen 640 Kamerabildpunkte maximal 1600 Bildschirmpunkte auf, also maximal 2,5 Bildschirmpunkte pro Pixel. Mit dieser Auflösung kann somit nur eine Genauigkeit von höchstens 1/2,5 Subpixel bei der Disparitätenschätzung erfolgen – anstatt den theoretisch möglichen 1/16. In der Praxis ist dies noch weniger, da nicht alle 1600 Bildschirmpunkte im Blickfeld der Kamera liegen. Nachträgliche Verkleinerung der Kamerabilder erhöht hier die mögliche Genauigkeit der Interpolation.

- Ebenso werden durch die Rasterung vor allem entfernter Objekte nicht alle Bildpunkte der verwendeten Texturen angezeigt. Unterscheiden sich die angezeigten Punkte zwischen der linken und rechten Sicht, findet keine Zuordnung der angezeigten Bereiche statt und es kann trotz gleicher und strukturierter Objekte keine passende Disparität geschätzt werden.

- Die Bildwiederholrate der Kameras ist geringer als die der Bildschirme. Bei der Aufzeichnung von Bewegungen kommt es zur Überlagerung mehrerer Bilder – es besteht die Gefahr, dass die zusätzlichen Kanten fehlerhafte Korrespondenzen hervorrufen. Bewegungsunschärfe entsteht nicht.

- Generell werden in der Simulation sich wiederholende Strukturen, z.B. beim Boden, verwendet. Die Regelmäßigkeiten können Fehlschätzungen der Disparität erzeugen.
- Sowohl die Bildschirme, als auch die Kamerasensoren bestehen aus diskreten Matrizen mit entsprechenden Begrenzungen der einzelnen Elemente. Die Überlagerung der unterschiedlichen Rasterungen führt zu sichtbaren Moiré-Effekten. Vor allem bei dunklen Bildern wird dies in der Mitte deutlich. Die damit entstehenden Kanten können den Vergleich der Bilder beeinträchtigen.

Bei den verwendeten LCD-Flachbildschirmen findet im Gegensatz zu Monitoren mit Kathodenstrahlröhren kein zeilenweiser Bildaufbau statt. Es entsteht kein Flackern bei der Bildaufzeichnung.

6.3 Versuchsdurchführung

6.3.1 Erzeugte Bilddaten und Kalibrierung

Für die erforderliche Kalibrierung der Stereokamera wird ein bekanntes Schachbrettmuster in der Sichtsimulation erzeugt. Es wird ermöglicht, das Muster zu bewegen, wodurch beliebige Ansichten aufgenommen werden können. Wie in Abschnitt 2.3 ab Seite 23 vorgestellt, werden bis zu zehn verschiedene Ansichten zur Gewinnung der inneren Kameraparameter und der relativen Lage zueinander erzeugt. Zur Kalibrierung werden die von den Kameras erzeugten Bilder verwendet, die bei der Aufzeichnung der Monitorbilder entstehen.

Dabei sind nicht alle Bilder geeignet – ist das Muster zu weit entfernt und damit zu klein, können die Eckpunkte nicht mehr eindeutig zugeordnet werden. Bei der Kalibrierung sind die gemessenen Pixelfehler zwischen 0,08 und 0,13. Dieses Ergebnis ist wesentlich besser als die Kalibrierung in realer Umgebung und erreicht fast die vom Hersteller angegebene Genauigkeit von 0,05 bis 0,1 Bildpunkten.

Folgende relative Kameraorientierung wurde ermittelt:

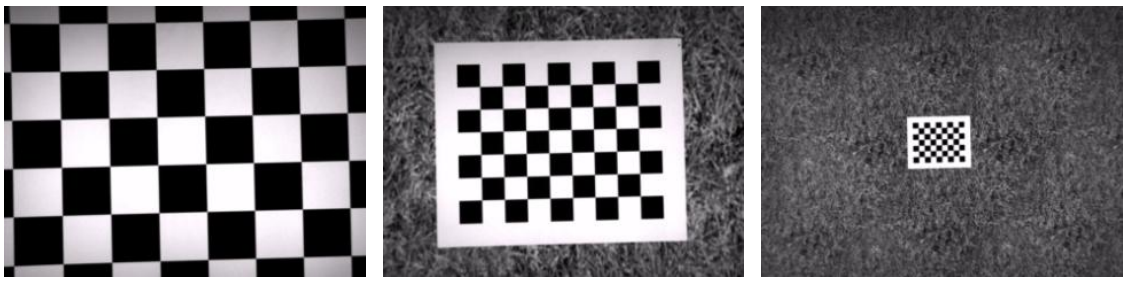
Wert	generiert	gemessen
t_x [mm]	300	299,45
t_y [mm]	0	8,56
t_z [mm]	0	3,29
ω [°]	0	−0,26
φ [°]	0	2,47
κ [°]	0	0,015

Tabelle 6.1: Vergleich der relativen Kameraorientierungen bei der Bildgenerierung und der Messung der Orientierung aus den Bildern der abgefilmten Monitore. Die Messungen wurden mit dem Programm *smallvcal* der svs-Software [SVS] vorgenommen.

Aus Tabelle 6.1 wird ersichtlich, dass die gemessene Kameraorientierung nicht exakt mit der relativen Orientierung übereinstimmt, die bei der Generierung der Sicht auf die beiden Bildschirme verwendet wurde. Ursache für die Abweichung ist neben geringfügigen Unterschieden der Kameras und der Monitore vor allem die ungenaue Positionierung der Kameras über den beiden Bildschirmen.

6.3.2 Kameraposition

Für die Bestimmung der Kameraposition am Helikopter wird ebenfalls das Kalibriermuster verwendet, welches in verschiedenen Entfernungen in x_f-Richtung vor dem Helikopter angezeigt wird. Die Translation des Musters in y_f- und z_f-Richtung sowie die Rotation ist null. Aus bekannter Translation und Rotation zwischen Helikopter und Muster und der aus den Bilddaten errechneten Verschiebung und Drehung der linken Kamera lässt sich die Abbildung von Punkten von Kamera- in Helikopterkoordinaten und umgekehrt rekonstruieren. Die Grundlagen dafür liefert Abschnitt 2.1.5 ab Seite 14.



(a) (b) (c)

Abbildung 6.3: Bild des Musters in der Simulationsumgebung. Der Helikopter ist in 0,5 Meter (a), 1,0 Meter (b) und 3,8 Meter Entfernung (c) platziert, die hier dargestellte Kamerasicht ist zum Helikopter weder verschoben, noch rotiert. Die dargestellten Bilder entsprechen den entzerrten und rektifizierten Aufnahmen der linken Kamera. Da die Positionierung der Kameras über den Monitoren nicht exakt gerade ist, entsteht bei der Rektifikation der Bilder eine leichte Drehung – auf den Bildschirmen sind die Kanten der Schachbrettquadrate exakt horizontal bzw. vertikal.

Bei der Beziehung zwischen Kamera und Helikopter wird eine Translation von null und eine Rotation von 90° um z_f- und x_f-Achse erwartet, da bei der Bildgenerierung die Koordinatensysteme bis auf die Verdrehung der Achsen übereinstimmen. Gemessen wurde die Translation und Rotation zwischen Kamera und Muster. In den Abbildungen 6.4 und 6.5 sind diese Messungen, umgerechnet in die Lage des Helikopters, in Abhängigkeit von der verwendeten Musterdistanz x_{mf} in x_f-Richtung zu sehen.

Gültige Werte sind in diesem Fall für Musterentfernungen zwischen 0,5 und 3,8 Metern messbar. Bei geringeren Distanzen sind nicht mehr alle Ecken des Schachbrettmusters im Blickfeld der Kamera, bei zu großer Entfernung ist das Muster zu klein, um alle Punkte fehlerfrei zu erkennen. Ursache der großen Streuung der Messwerte insbesondere bei den Rotationen können

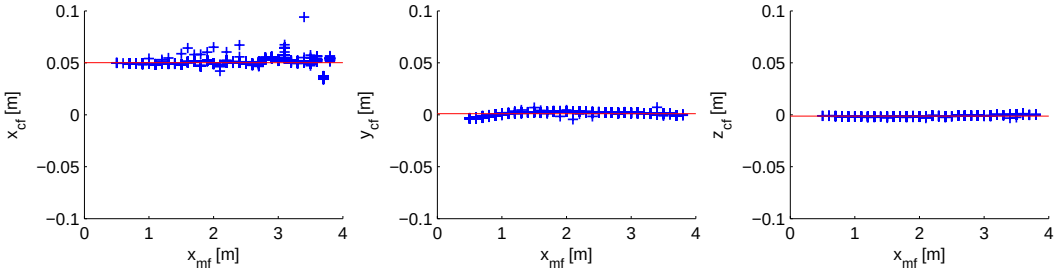


Abbildung 6.4: Positionsbestimmung, Gemessene Translation von Kamera- in Helikopterkoordinaten in Abhängigheit von der Musterdistanz (blau) und der Mittelwert aller Messungen (rot).

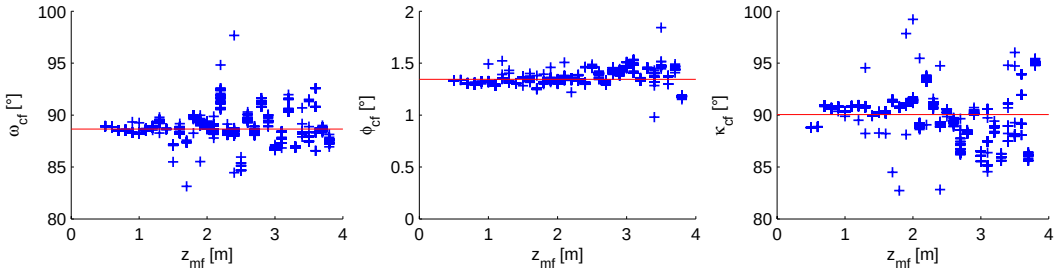


Abbildung 6.5: Positionsbestimmung, Gemessene Rotation von Kamera- in Helikopterkoordinaten in Abhängigkeit von der Musterdistanz (blau) und der Mittelwert aller Messungen (rot).

Bildrauschen und vor allem Aliasing-Effekte bei der Generierung gerasterter Musterbilder auf den beiden Bildschirmen sein.

Für die Transformation von Kamera- in Helikopterkoordinaten in späteren Versuchen werden die Mittelwerte dieser Messungen verwendet. Es ist $\mathbf{R}_{cf} = R(\omega_{cf}, \varphi_{cf}, \kappa_{cf})$ und $\mathbf{t}_{cf} = (x_{cf}, y_{cf}, z_{cf})$. Die konkreten Werte, in den Abbildungen 6.4 und 6.5 rot gekennzeichnet, sind in Tabelle 6.2 dargestellt. Wie zu erkennen ist, stimmt die in der Sichtsimulation erzeugte Rotation zwischen Kamera und Helikopter nicht exakt mit der gemessenen Rotation überein, bei der Translation fällt auf, dass sich das ermittelte Kamerazentrum etwa 50 mm vor dem Helikopterzentrum befindet. Die Unterschiede ergeben sich aus der ungenauen Positionierung der Kameras über den Bildschirmen und die Drehung bzw. Vergrößerung der Bilder durch Entzerrung und Rektifikation.

Wert	generiert	gemessen
x_{cf} [mm]	0	50,19
y_{cf} [mm]	0	0,99
z_{cf} [mm]	0	−1,23
ω_{cf} [°]	90	88,65
φ_{cf} [°]	0	1,34
κ_{cf} [°]	90	90,04

Tabelle 6.2: Von der Sichtsimulation erzeugte und mittlere gemessene Kameraposition am Helikopter. Angegeben sind die Transformationsparameter von Kamera- in Helikopterkoordinaten.

6.3.3 Entfernungsmessung

Die Entfernungsmessung zu potentiellen Hindernissen mit Hilfe der Disparitätenbilder setzt eine Kamerakalibrierung und die Bestimmung der Kameraposition voraus.

Für die Messung von Entfernungen wird in der Simulation eine Umgebung erzeugt, die lediglich einen Baum enthält. In diese Umgebung wird der Helikopter auf verschiedene Positionen so platziert, dass sich der Baum im sichtbaren Bereich der Kamera befindet. Die globalen Koordinaten von Helikopter und Baum sind bekannt, woraus sich die Entfernung zwischen Baum und Helikopter ergibt. Darüber hinaus wird mit Hilfe der Tiefenbilder die Entfernung zum Baum gemessen und mit der tatsächlichen Entfernung verglichen.

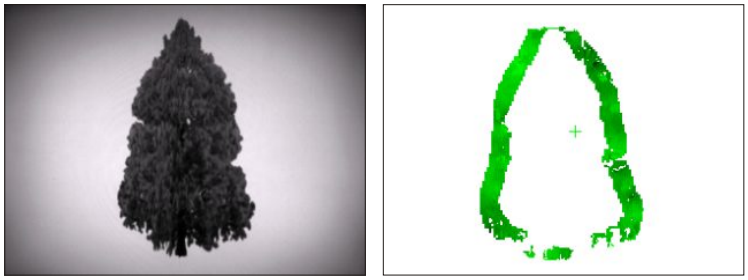


Abbildung 6.6: Objekt, welches für die Entfernungsmessung verwendet wird, hier im Abstand von 20 Meter. Linkes Originalbild (a), Disparitätenbild (b).

Die aufgezeichneten Kamerabilder haben eine Auflösung von 640×480 Bildpunkten, zusätzlich werden verkleinerte Bilder für die Erstellung einer Tiefenkarte und damit einer Entfernungsmessung verwendet.

In dem Versuch ist der Helikopter in verschiedenen Abständen zum Baum positioniert. Bei Werten zwischen 8,5 m und 77 m konnte dessen Entfernung mit Hilfe der Stereobildauswertung geschätzt werden. Bei Distanzen außerhalb dieses Bereiches wurde das Hindernis bei allen Auflösungen nicht korrekt erkannt. Ergebnisse mit verschiedenen Bildauflösungen zeigt Abbildung 6.7.

Grundsätzlich kann hier die Entfernung des Hindernisses bei Bildauflösungen von 640 × 480, 320×240 und 160×120 Pixeln bis zu einer Entfernung von ca. 55 m in akzeptabler Genauigkeit gemessen werden. Die unterschiedlichen Werte bei einer Entfernung entstehen durch Bildrauschen, da die Sicht bei der Entfernungsmessung eines konkreten Abstandes nicht verändert wurde. Abweichungen sind vor allem nach unten zu beobachten und auf fehlerhafte Bildpunkte mit hoher Disparität zurückzuführen. Die Mittelwerte der Messungen sind im linken Diagramm von Abbildung 6.8 visualisiert.

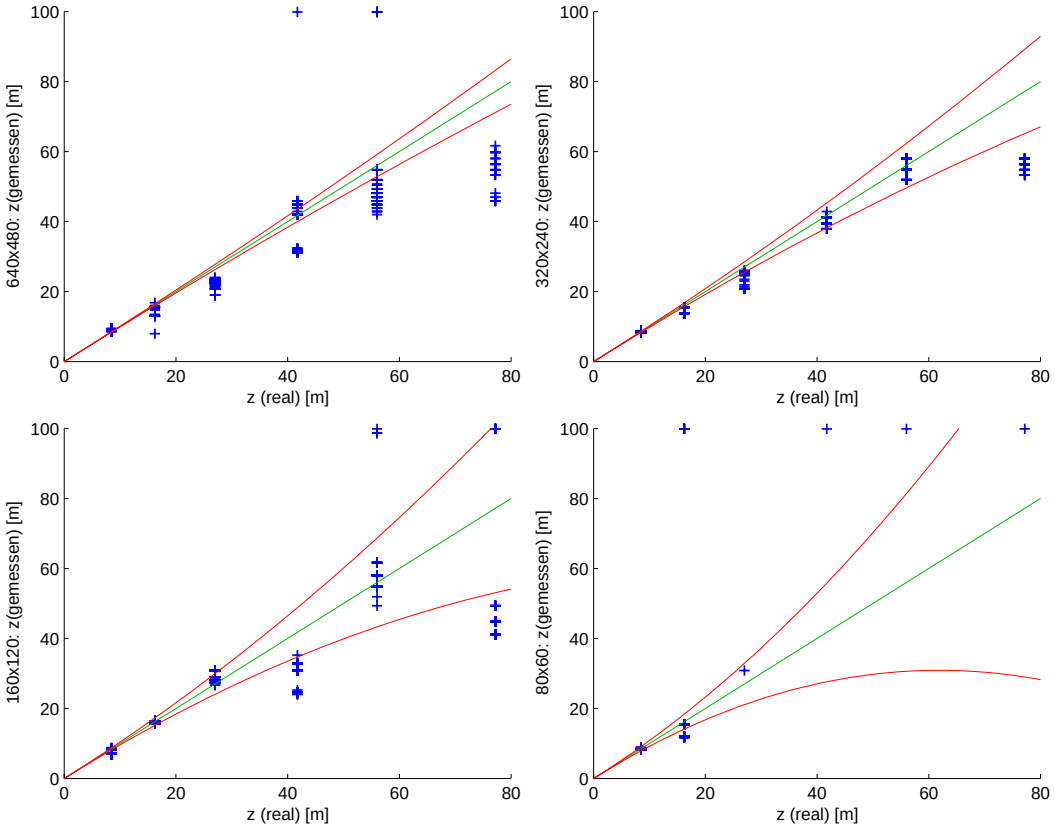


Abbildung 6.7: Entfernungsmessungen mit dem Disparitätenbild unter verschiedenen Auflösungen. Die grüne Linie zeigt jeweils die tatsächliche Entfernung z, die roten Linien begrenzen den Genauigkeitsbereich $z \pm \Delta z/2$ bei einer Genauigkeit von 1/2 Subpixel. Der Basisabstand beträgt 299,45 mm und die Brennweite 824 Pixel bei einer Bildauflösung von 640×480.

Falsche Ergebnisse von 100 Metern sind keine Fehlmessungen in dem Sinne, sondern entsprechen der Aussage „keine Information“, da die Maximaldistanz des Filters auf diesen Wert gesetzt ist und dieser Wert ausgegeben wird, sofern keine gültigen Werte im entsprechenden Bereich des Tiefenbildes verfügbar sind. Wie in Abbildung 6.6 zu sehen ist, ist nur der Rand des Baumes kontrastreich genug, um eine Tiefenschätzung zu ermöglichen. Dies kann bei großen Hindernisobjekten zu Problemen führen, wenn der Rand nicht mehr sichtbar und damit keine Entfernungsmessung möglich ist. Bei der Bildauflösung von 80×60 Bildpunkten ist dies im Gegensatz zu Bildern anderer Größe ab der Messung von 40 Metern stets der Fall.

Werden die Ausgangsbilder auf 40×30 Pixel herunter gerechnet, nimmt der Rand, in dem keine gültigen Disparitäten verfügbar sind, das gesamte Bild ein, was eine Entfernungsmessung nicht ermöglicht. Die Ergebnisse der Entfernungsmessung sind in diesem Fall und auch bei noch geringeren Bildauflösungen komplett unbrauchbar und wurden daher nicht mit aufgezeichnet.

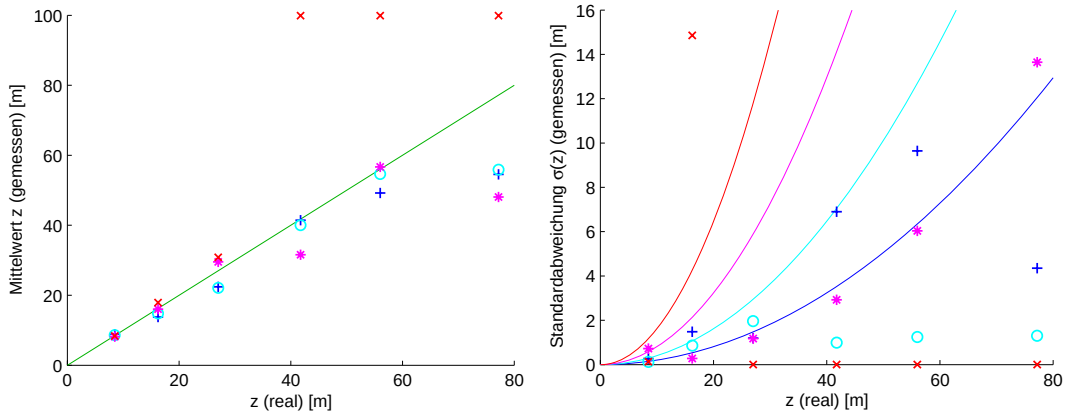


Abbildung 6.8: Mittelwerte (links) und Standardabweichung (rechts) der Entfernungsmessungen, als Kreuze markiert. Die verschiedenen Bildauflösungen sind farblich gekennzeichnet: 640×480 blau (+), 320×240 cyan ($\circ$), 160×120 magenta ($\star$) und 80×60 rot ($\times$). Die farbigen Linien entsprechen der echten Distanz (links) bzw. der jeweiligen Entfernungsauflösung (rechts) bei einer Genauigkeit von 1/2 Subpixel.

Darüber hinaus ist ersichtlich, dass die Streuung der Werte bei höheren Auflösungen entgegen der theoretischen Vermutung nicht geringer ist, eine Veranschaulichung liefert Abbildung 6.8 (rechts). Die Verkleinerung der Bilder und die damit verbundene Glättung reduziert dort das Rauschen der Stereobilder und damit den Anteil fehlerhafter Tiefenschätzungen. Die Genauigkeit ist oft sogar besser als bei höheren Auflösungen. Das Fazit der Messungen ist, dass bei dieser Versuchsanordnung die Bilder mit 640×480, 320×240 und 160×120 Punkten zur Kollisionsvermeidung herangezogen werden können. Bei einer Auflösung von 320×240 Pixel ist die Streuung derjenigen Messwerte, deren Mittel in etwa der wirklichen Entfernung entspricht, am geringsten.

6.3.4 Simulation eines Fluges

In diesem Versuch wird die gesamte Simulationsumgebung verwendet – es entsteht ein Regelkreis mit Kameras, Bildverarbeitung, Flugsimulation und der Erzeugung von Bildern. Dabei wird in der Simulation eine Umgebungswelt erzeugt, die als Hindernis den oben dargestellten Baum zeigt und ansonsten aus einer hindernisfreien Landschaft besteht.

Die Weltkoordinaten (x_g, y_g, z_g) des Baumes sind $(-60, -200, 0)$, die Startposition des Helikopters ist $(-60, 100, 0)$. Der Baum besitzt einen Durchmesser von 10 m. Bezugspunkt des Koordinatensystems ist ein fest definierter Punkt der Simulationsumgebung, die x_g-Achse zeigt nach Norden. Die x_f-Achse des Helikopters ist nach Westen ausgerichtet, ein Flug in diese Richtung würde eine Kollision mit dem Hindernis verursachen.

Die Verwendung der in dieser Arbeit implementierten Kollisionsvermeidung ermöglicht die Erkennung des Hindernisses und das Senden von Flugkommandos an den Helikopter. Funk-

tioniert die Kollisionsvermeidung korrekt, wird der Kurs entsprechend geändert und der Helikopter fliegt an dem Hindernis vorbei. In diesem Fall wird kein Kommando zur Änderung des Steuerkurses (Azimut) erzeugt. Es ist demnach zu erwarten, dass bei Erkennung des Hindernisses der Helikopter an diesem vorbei und in die gleiche Richtung weiter fliegt.

Bei der Simulation wurde der Radius des Sicherheitskorridors (*Sicherheitsradius*) und der Schwellwert (*Ausweichdistanz*) variiert, bis zu dem ein Ausweichen stattfinden soll. Ebenso wurden die Kamerabilder auf verschiedene Auflösungen skaliert. Die Maximalgeschwindigkeit v_0 des Helikopters ist in jedem Fall 5 m/s, die Bremsbeschleunigung a ist auf 2 m/s^2 begrenzt. Nach der beschriebenen Bewegungsgleichung 5.17 (Seite 83) mit den dort verwendeten Verzögerungswerten kann der Helikopter innerhalb von 12,5 m anhalten – dies ist der minimale Wert für die Ausweichdistanz. Ebenso müssen Hindernisse bis zu dieser Entfernung erkannt werden können. In der verwendeten Sichtsimulation sind Entfernungsmessungen von über 50 m möglich, sie ist damit geeignet für Flüge unter gegebenen Bedingungen.

Nr.	Bildauflösung	Sicherheitsradius	Ausweichdistanz	Ergebnis
1	640×480	1,0 m	30 m	Hindernis umflogen
2	640×480	1,5 m	30 m	Hindernis umflogen
3	640×480	2,5 m	30 m	vor Hindernis angehalten
4	320×240	1,5 m	30 m	vor Hindernis angehalten
5	320×240	1,0 m	30 m	Hindernis umflogen
6	320×240	2,0 m	50 m	Hindernis umflogen
7	160×120	2,0 m	30 m	vor Hindernis angehalten
8	160×120	1,0 m	30 m	Hindernis umflogen

Tabelle 6.3: Ergebnisse des Simulationsversuchs mit verschiedenen Einstellungen der Kollisionsvermeidung.

Tabelle 6.3 zeigt die Ergebnisse von simulierten Flügen mit verschiedenen Einstellungen. In allen Flügen konnte das Hindernis erkannt und eine Kollision vermieden werden. Allerdings ist in einigen Fällen ein Stopp vor dem Baum erfolgt, es konnte kein alternativer Ausweichkurs gefunden werden. Abbildung 6.9 zeigt die Bahnen der einzelnen Simulationsflüge.

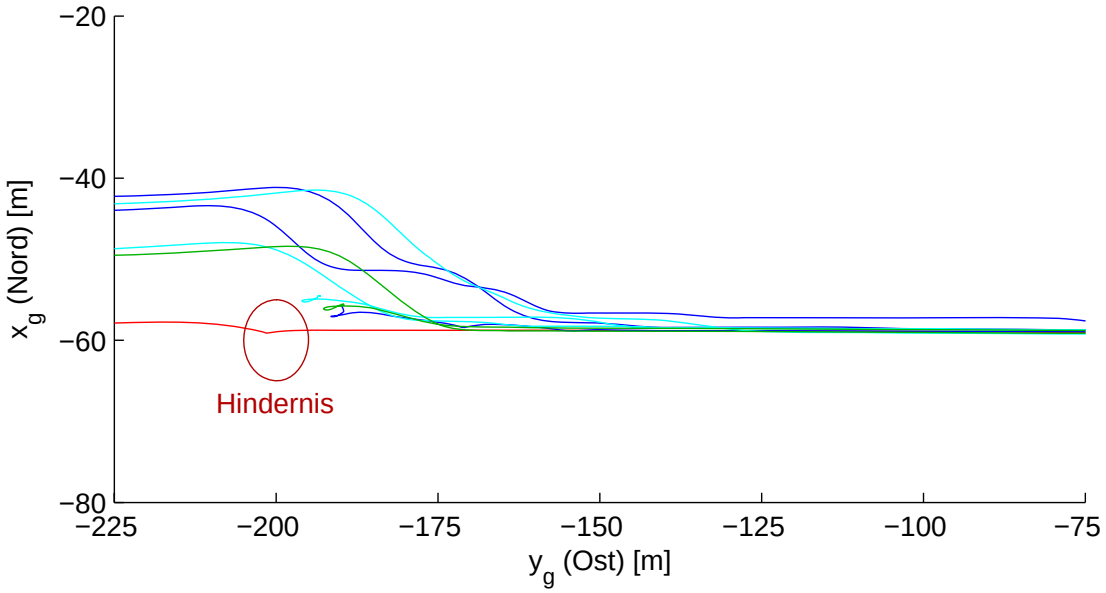


Abbildung 6.9: Flugbahnen des Helikopters beim Simulationsversuch. Hinderniserkennung bei Bildauflösungen von 640 × 480 (blau), 320 × 240 (cyan) und 160 × 120 Pixel (grün). Zum Vergleich: Flugbahn ohne Hinderniserkennung (rot). In der Darstellung ist der Helikopter von rechts nach links geflogen.

Größere Sicherheitsradien und geringere Ausweichdistanzen führen dazu, dass der Helikopter das Hindernis nicht umfliegt, sondern vorher anhält. Bei großen Sicherheitsradien müssen freie Bereiche entsprechend größer sein und es ist damit wahrscheinlicher, dass kein freier Bereich gefunden wird. Bei kleinen Ausweichdistanzen reagiert der Helikopter beim Heranfliegen zu spät auf das Hindernis, wodurch es in den Kamerabildern größer ist und ebenfalls keine Ausweichmöglichkeiten mehr gefunden werden. Ebenso zeigt sich, dass auch mit geringerer Bildauflösung eine Kollisionsvermeidung problemlos möglich ist.

Kapitel 7

Zusammenfassung und Ausblick

Gegenstand dieser Arbeit ist die Entwicklung einer Strategie zur Kollisionsvermeidung unter Verwendung von Kamerabildern. Sie ist insbesondere für den unbemannten Kleinhubschrauber ARTIS gedacht.

Für die Kollisionsvermeidung sind einige Vorbetrachtungen erforderlich. Zunächst müssen aus Kamerabildern dreidimensionale Informationen der Umgebung entnommen werden. Da dies aus einem Einzelbild nicht möglich ist, wird eine Stereokamera verwendet. Zwei Kameras erfassen die Umgebung aus verschiedenen Perspektiven und lassen so eine Rekonstruktion der Umgebung zu, sofern die innere und äußere Orientierung jeder Kamera bekannt ist.

Die Kalibrierung der Stereokamera ermöglicht eine genaue Messung der inneren Parameter und der relativen Orientierung beider Kameras. Durch Entzerrung und Rektifikation werden Bilder erzeugt, die der standard-stereoskopischen Anordnung entsprechen. Damit ist es im Vergleich zu einer beliebigen Kameraanordnung relativ einfach möglich, aus den beiden Bildern die dreidimensionalen Koordinaten von Objektpunkten in einem kamerafesten Koordinatensystem zu berechnen.

Für die Darstellung der Tiefeninformation wird ein Bild erzeugt, das für jeden Bildpunkt ein Maß für die Entfernung des dargestellten Objektpunktes enthält. Aus den Punkten dieses Tiefenbildes lassen sich unter Verwendung der Kameraparameter die Koordinaten der dargestellten Objektpunkte berechnen.

Die Kollisionsvermeidung wertet die erzeugten Tiefenbilder aus. Hierbei wird die äußere Orientierung der Kamera, d.h. dessen Position am Helikopter, benötigt. Damit lassen sich Objektkoordinaten mit Bezug zum Helikopter herstellen. Ziel der Tiefenbildauswertung ist die Erkennung von Hindernissen, die sich direkt vor dem Helikopter befinden und bei angenommenem Geradeausflug zu einer Kollision führen würden. Werden nahe Hindernisse erkannt, muss der aktuelle Kurs geändert werden. Naheliegend sind Ausweichmanöver mit möglichst wenig Richtungsänderungen zum Umfliegen der Hindernisse. Dazu wird im Tiefenbild nach

hindernisfreien Bereichen gesucht. Falls solche Bereiche gefunden werden, kann der Helikopter dorthin fliegen und es wird eine Anweisung zur Kursänderung an die Flugsteuerung gesendet. Werden im Tiefenbild keine freien Regionen gefunden, muss der Helikopter an der aktuellen Position anhalten – es wird ein entsprechendes Kommando gegeben.

In dieser Arbeit ist das entwickelte Verfahren zur Kollisionsvermeidung in einer Flugsimulation erprobt worden. Es wurde die innere, relative und äußere Orientierung der Stereokamera in der Simulationsumgebung ermittelt und die Zuverlässigkeit der Entfernungsmessung getestet. Abschließend erfolgte ein simulierter Flug mit dem Ziel, die Kollision zu einem Hindernis zu verhindern. Die Versuche zur Kollisionsvermeidung verliefen erfolgreich.

Die Arbeit hat gezeigt, dass die stereoskopische Tiefenschätzung zur Findung von Hindernissen und Ausweichmöglichkeiten geeignet ist. Die verwendete Software zur Berechnung von Tiefenbildern ist dazu ausreichend schnell und genau, ebenso sind Bildauflösung und -qualität der Kameras akzeptabel. Die nachträgliche Filterung der Tiefenbilder kann einige Schwachstellen der verwendeten Software ausgleichen, das Resultat sind weniger fehlerhafte Bilder.

Allerdings entstehen optimale Tiefenbilder erst nach der manuellen Justierung der Parameter des Tiefenfilters durch den Anwender. Das Finden von Regeln zur adaptiven Einstellung dieser Parameter und damit zur automatischen Optimierung der Tiefenbilder bietet ein hohes Potenzial für zukünftige Forschungen. Ziel kann beispielsweise die Entwicklung eines Filters auf Basis der hier verwendeten Software sein, der in jeder Situation bestmögliche Tiefenkarten erzeugt und, im Gegensatz zu bereits vorhandenen globalen Optimierungsmethoden, echtzeitfähig ist.

Darüber hinaus kann die in der Einführung bereits erwähnte Tiefenbilderzeugung auf einem FPGA den Bildverarbeitungsrechner entlasten. Dadurch steht anderen Anwendungen, die auf dem gleichen System laufen, wesentlich mehr Rechenleistung zur Verfügung. Beispielsweise hätte dann die Bildanalyse zur Kollisionsvermeidung eine höhere Bildwiederholrate und könnte dem Helikopter in kürzeren Intervallen Richtungsanweisungen geben.

Offen ist, inwieweit die Ergebnisse aus der Simulationsumgebung auf reale Flüge übertragbar sind. Die Reaktion des Helikopters auf Steuerkommandos kann weitestgehend nachgebildet werden, allerdings ist es nicht bekannt, ob dies auch auf die Bilderzeugung und -verarbeitung zutrifft. Das Abfilmen von Bildschirmen, die eine gerenderte 3D-Landschaft zeigen, hat andere Schwachstellen als die Aufzeichnung einer echten Umgebung. Erst die Flugversuche mit dem ARTIS-Helikopter können die Funktionstauglichkeit der Kollisionsvermeidung unter realen Bedingungen beweisen. Der Vergleich zwischen Flug und Simulation hinsichtlich Genauigkeit, Fehlerquellen und Flugverhalten bei der Kollisionsvermeidung ist daher eines der zukünftigen Forschungsthemen.

Anhang A

Herleitungen

A.1 Positionsschätzung

Ausgangspunkt für die Positionsschätzung der Kamera ist ein Bild eines planaren Kalibriermusters, welches n signifikante Punkte, z.B. Ecken, enthält. Die Lage der Punkte bezogen auf den Ursprung des Kalibriermusters ist bekannt, ebenso die Lage der entsprechenden Bildpunkte. Die Bildpunkte seien mit $\mathbf{x}_i$, die Objektpunkte des Musters mit $\mathbf{q}_i$ bezeichnet, mit $i \in \{1, ..., n\}$.

Die Beziehung zwischen Bild- und Objektpunkten ist unter Verwendung homogener Koordinaten

$$z_i \mathbf{x}_i = \mathbf{K}[\mathbf{R}\,\mathbf{t}]\mathbf{q}_i \tag{A.1}$$

mit einer unbekannten Skalierung z_i, der Kameramatrix $\mathbf{K}$ und der Projektionsmatrix $[\mathbf{R}\,\mathbf{t}]$, die die Rotation zwischen Objekt- und Bildebene beschreibt. Unter Verwendung einer kalibrierten Kamera sei die Kameramatrix $\mathbf{K}$ die Einheitsmatrix – die gegebenen Bildkoordinaten liegen entsprechend in normiertem Format vor. Damit ist $z_i \mathbf{x}_i = [\mathbf{R}\,\mathbf{t}]\mathbf{q}_i$ und die Skalierung z_i gegeben durch

$$z_i = \mathbf{e}_z^T\,[\mathbf{R}\,\mathbf{t}]\mathbf{q}_i \tag{A.2}$$

mit $\mathbf{e}_z = (0, 0, 1)^T$. Es folgt

$$(\mathbf{x}_i\,\mathbf{e}_z^T - \mathbf{I})\,[\mathbf{R}\,\mathbf{t}]\,\mathbf{q}_i = 0 \tag{A.3}$$

mit der gesuchten Kameraposition $[\mathbf{R}\,\mathbf{t}]$.

Bei der linearen Positionsbestimmung sei ohne Beschränkung der Allgemeinheit das verwendete Koordinatensystem so definiert, dass die Ebene der Objektpunkte $\mathbf{q}_i$ durch die x- und y-Achse aufgespannt wird. Die z-Komponente aller Objektpunkte ist unter dieser Voraussetzung gleich null. Es ist also $\mathbf{e}_z^T \mathbf{q}_i = 0\,(\forall i)$.

Mit $\mathbf{q}_i = (q_{ix}, q_{iy}, q_{iz}, 1)^T$ und $[\mathbf{R}\,\mathbf{t}] = [\mathbf{r}_1\,\mathbf{r}_2\,\mathbf{r}_3\,\mathbf{t}]$ ergibt sich aus Gleichung A.3 unter der festgelegten Bedingung $q_{iz} = 0$ die Gleichung

$$(\mathbf{x}_i\,\mathbf{e}_z^T - \mathbf{I})[\mathbf{r}_1\,\mathbf{r}_2\,\mathbf{t}] \begin{pmatrix} q_{ix} \\ q_{iy} \\ 1 \end{pmatrix} = 0. \tag{A.4}$$

Dieses lineare Gleichungssystem lässt sich durch Anordnung der skalaren Elemente von $\mathbf{r}_1$, $\mathbf{r}_2$ und $\mathbf{t}$ in einer Spalte nach

$$\mathbf{F} \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{t} \end{pmatrix} = 0 \tag{A.5}$$

umstellen, mit der $2n \times 9$-Matrix

$$\mathbf{F} = \begin{pmatrix} q_{1x} & 0 & -q_{1x}x_{1x} & q_{1y} & 0 & -q_{1y}x_{1x} & 1 & 0 & -x_{1x} \\ 0 & q_{1x} & -q_{1x}x_{1y} & 0 & q_{1y} & -q_{1y}x_{1y} & 0 & 1 & -x_{1y} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{nx} & 0 & -q_{nx}x_{nx} & q_{ny} & 0 & -q_{ny}x_{nx} & 1 & 0 & -x_{nx} \\ 0 & q_{nx} & -q_{nx}x_{ny} & 0 & q_{ny} & -q_{ny}x_{ny} & 0 & 1 & -x_{ny} \end{pmatrix} \tag{A.6}$$

und den Bildpunkten $\mathbf{x}_i = (x_{ix}, x_{iy}, 1)^T$.

Es kann gezeigt werden, dass $\text{rang}(\mathbf{F}) = 8$ gilt, sofern mindestens vier Punkte existieren, von denen keine drei Punkte kollinear sind. Rauschen führt allerdings zu Fehlern bei der Erkennung der Bildpunkte, so dass $\mathbf{F}$ normalerweise vom Rang 9 ist.

Eine Schätzung des Nullraums von $\mathbf{F}$, d.h. jener Vektor $(\hat{\mathbf{r}}_1^T\,\hat{\mathbf{r}}_2^T\,\hat{\mathbf{t}}^T)^T$, welcher Gleichung A.5 minimiert, lässt sich durch Singulärwertzerlegung[1] finden.

Sei die Singulärwertzerlegung $\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ mit der Diagonalmatrix $\mathbf{D} = \text{diag}(d_1, ..., d_9)$ und $d_1 \geq \ldots \geq d_9$. Dann löst der Singulärvektor $(\hat{\mathbf{r}}_1^T\,\hat{\mathbf{r}}_2^T\,\hat{\mathbf{t}}^T)^T$, der mit dem kleinsten Singulärwert d_9 korrespondiert, das Minimierungsproblem. Es ist also die letzte Spalte der Matrix $\mathbf{V}$.

Falls nötig, wird der Ergebnisvektor negiert, um die Gleichung $\hat{t}_z \geq 0$ zu erfüllen. Da sich das Objekt vor der Kamera befinden muss, existiert eine positive Translation in z-Richtung, die dadurch sichergestellt ist.

Die letztendlich ausgegebene skalierte Translation $\mathbf{t}$ ist

$$\mathbf{t} = \frac{2\hat{\mathbf{t}}}{(\|\hat{\mathbf{r}}_1\| + \|\hat{\mathbf{r}}_2\|)}. \tag{A.7}$$

[1] Jede komplexe Matrix $\mathbf{M}$ vom Rang r lässt sich in drei Matrizen $\mathbf{U}$,$\mathbf{D}$,$\mathbf{V}$ in der Form $\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^*$ zerlegen. $\mathbf{V}^*$ bezeichnet die adjungierte Matrix von $\mathbf{V}$, die im reellen Fall der transponierten Matrix $\mathbf{V}^T$ entspricht. Die Spaltenvektoren von $\mathbf{U}$ und $\mathbf{V}$ sind orthonormal und $\mathbf{D}$ eine Diagonalmatrix mit r positiven, der Größe nach geordneten Singulärwerten [Wil03].

Die Rotationsmatrix $\mathbf{R}$ lässt sich aus den Vektoren $\hat{\mathbf{r}}_1$ und $\hat{\mathbf{r}}_2$ errechnen. Sei die Singulärwertzerlegung $(\hat{\mathbf{r}}_1\,\hat{\mathbf{r}}_2\,0) = \mathbf{UDV}^T$, so ist $\mathbf{R} = \mathbf{UV}^T$ [För99]. Um sicherzustellen, dass es sich um eine Rotationsmatrix handelt, muss $\det(\mathbf{R}) = 1$ gelten. Falls sich eine Determinante von -1 ergibt, wird die letzte Spalte von $\mathbf{R}$ negiert.

Die so ermittelte Rotation und Translation ist aufgrund von Rauschen und der Tatsache, dass 9 Parameter in einem System von 6 Freiheitsgraden geschätzt werden, fehlerbehaftet. Die ermittelte Genauigkeit ist für eine Positionsschätzung möglicherweise nicht ausreichend. Allerdings existiert ein nichtlineares Verfahren, das die Genauigkeit iterativ erhöht. Als Startwerte für dieses Verfahren sind die linear ermittelten Werte geeignet.

Die Ungenauigkeit sei dargestellt durch den Fehlervektor $G = (G_1^T \ \ldots \ G_n^T)$ mit den Elementen

$$G_i = (\mathbf{x}_i\mathbf{e}_z^T - \mathbf{I})\,[\mathbf{R}\,\mathbf{t}]\,\mathbf{q}_i, \tag{A.8}$$

wobei die letzte Zeile der Ergebnismatrix ignoriert wird. Die Rotationsmatrix $\mathbf{R}$ wird parametrisiert durch die Eulerwinkel ω, φ und κ Die sechs Schätzparameter β für die nichtlineare Optimierung sind damit $\beta = (\omega\,\varphi\,\kappa\,t_x\,t_y\,t_z)^T$.

Mit der mehrdimensionalen Newton-Raphson-Methode[2] erfolgt die Approximation einer Nullstelle von $\|G\|$ und damit eine iterative Schätzung von β. Es ist

$$\beta_{n+1} = \beta_n - k_n(\mathbf{D}_\beta G|_{\beta_n})^\dagger\, G(\beta_n) \tag{A.9}$$

mit einer adaptiven Schrittweite k_n, der Jacobimatrix $\mathbf{D}_\beta G$ mit den Ableitungen von G nach β und dem Moore-Penrose Pseudoinversen[3] $(\mathbf{D}_\beta G|_\beta)^\dagger$ von $\mathbf{D}_\beta G|_\beta$. Als Startwert β_0 wird das Ergebnis der linearen Schätzung verwendet.

Mit diesem Verfahren erfolgt eine genauere Schätzung der Rotation und Translation. Aufgrund der Nichtlinearität können weitere, falsche lokale Minima existieren. Daher muss der Startwert ausreichend genau sein, damit die korrekte Nullstelle von $\|G\|$ angenähert wird.

[2] Das Newton-Raphson-Verfahren dient zur iterativen Lösung von nichtlinearen Gleichungen und Gleichungssystemen mit Hilfe der Differentialrechnung. Im eindimensionalen Fall ist die Nullstelle einer Tangente an einem Punkt die Näherungslösung bzw. Ausgangspunkt für den nächsten Iterationsschritt [Gou01].

[3] Ist mit $\mathbf{M} = \mathbf{UDV}^*$ die Singulärwertzerlegung der Matrix $\mathbf{M}$ gegeben, so ist das Pseudoinverse $\mathbf{M}^\dagger = \mathbf{VD}^\dagger\mathbf{U}^*$. Das Pseudoinverse der Diagonalmatrix $\mathbf{D}^\dagger$ ergibt sich aus den Reziproken aller von null verschiedenen Diagonalelemente, sonstige Elemente sind null [Wil03]. Die Eigenschaften sind dem Inversen ähnlich, es ist allerdings auch für nichtquadratische Matrizen definiert.

A.2 Die Fundamentalmatrix

Die Fundamentalmatrix beschreibt die Beziehung zwischen zwei Bildpunkten auf verschiedenen Bildebenen, die den gleichen Objektpunkt darstellen. Die beiden Bildpunkte sind mit zwei Kameras aus verschiedenen Positionen und Blickwinkeln aufgezeichnet.

Prinzipiell ist die Lagebeziehung der Kameras so gegeben, dass die zweite Kamera bezogen auf die erste verschoben und rotiert ist, beschrieben durch einen Translationsvektor $\mathbf{t} = (t_x, t_y, t_z)^T$ und eine Rotationsmatrix $\mathbf{R}$.

Zur Vereinfachung stellt die erste Kamera das Weltkoordinatensystem, d.h. die Translation und Rotation bezüglich des ersten Kamerazentrums ist null. Nach der Projektion in homogenen Koordinaten (Gleichung 2.9, Seite 10) ist damit

$$\mathbf{P}_1 = \mathbf{K}_1[\mathbf{I}\,\mathbf{0}] \quad \text{, bzw.} \quad \mathbf{P}_2 = \mathbf{K}_2[\mathbf{R}\,\mathbf{t}] \tag{A.10}$$

mit der Einheitsmatrix $\mathbf{I}$ und dem Nullvektor $\mathbf{0}$ zur Beschreibung einer nicht vorhanden Translation und Rotation. Eingesetzt in Gleichung 2.8 (Seite 10) ergibt sich

$$z_1\widetilde{\mathbf{m}}_1 = \mathbf{K}_1[\mathbf{I}\,\mathbf{0}]\widetilde{M} \quad \text{, bzw.} \quad z_2\widetilde{\mathbf{m}}_2 = \mathbf{K}_2[\mathbf{R}\,\mathbf{t}]\widetilde{M}. \tag{A.11}$$

Nach Zhang [Zha98b] lassen sich z_1, z_2 und M aus der Gleichung eliminieren, es ergibt sich

$$\widetilde{\mathbf{m}}_2^T\mathbf{K}_2^{-T}[\mathbf{t}]_\times\mathbf{R}\mathbf{K}_1^{-1}\widetilde{\mathbf{m}}_1 = 0. \tag{A.12}$$

Dabei ist $[\mathbf{t}]_\times$ die antisymmetrische Kreuzproduktmatrix von $\mathbf{t}$.

Die Fundamentalmatrix $\mathbf{F}$ ergibt sich aus der Substitution

$$\mathbf{F} := \mathbf{K}_2^{-T}\,[\mathbf{t}]_\times\,\mathbf{R}\,\mathbf{K}_1^{-1} \tag{A.13}$$

und es folgt aus Gleichung A.12 letztendlich

$$\widetilde{\mathbf{m}}_2^T\,\mathbf{F}\,\widetilde{\mathbf{m}}_1 = 0. \tag{A.14}$$

Sind zusätzlich die internen Parameter der Kamera bekannt oder die normierten Bildpunkte $\widetilde{\mathbf{m}}_{c1} := \mathbf{K}_1^{-1}\widetilde{\mathbf{m}}_1$ bzw. $\widetilde{\mathbf{m}}_{c2} := \mathbf{K}_2^{-1}\widetilde{\mathbf{m}}_1$ gegeben, lässt sich die Beziehung dieser beiden Punkte durch die *essentielle Matrix* $\mathbf{E}$ von Longuet-Higgins [Lon81] beschreiben – es ist

$$\mathbf{E} = [\mathbf{t}]_\times\,\mathbf{R} \tag{A.15}$$

und

$$\widetilde{\mathbf{m}}_{c2}^T\,\mathbf{E}\,\widetilde{\mathbf{m}}_{c1} = 0. \tag{A.16}$$

Die Fundamentalmatrix $\mathbf{F}$ hat folgende Eigenschaften:

- Es ist $\det(\mathbf{F}) = 0$.
- Es ist $\text{rang}(\mathbf{F}) = 2$.
- F ist skalierungsinvariant. $\mathbf{F}$ kann nur bis auf einen skalaren Faktor bestimmt werden, jedes Element dieser Menge von Matrizen bildet einen Punkt $\widetilde{\mathbf{m}}$ auf die gleiche Linie $\mathbf{l}$ ab.
- $\mathbf{F}$ besitzt 7 Freiheitsgrade.
- Für die beiden Epipole $\mathbf{e}_1$ und $\mathbf{e}_2$ gilt: $\mathbf{F}\,\mathbf{e}_1 = \mathbf{F}^T\,\mathbf{e}_2 = 0$.
- Die umgekehrte Beziehung zwischen zwei korrespondierenden Punkten ist durch die transponierte Fundamentalmatrix beschrieben. Es gilt die Gleichung $\widetilde{\mathbf{m}}_1^T \mathbf{F}^T \widetilde{\mathbf{m}}_2 = 0$.

A.3 Bestimmung der Fundamentalmatrix

Sei die Beziehung korrespondierender Punkte durch die Gleichung A.14 mit den Punkten $\widetilde{\mathbf{m}} := \widetilde{\mathbf{m}}_1$ und $\widetilde{\mathbf{m}}' := \widetilde{\mathbf{m}}_2$ gegeben, welche die Elemente $(x, y, 1)^T$ bzw. $(x', y', 1)^T$ besitzen. Die Fundamentalmatrix $\mathbf{F}$ besitzt die neun Elemente $f_{11}, ..., f_{33}$. Aus der Gleichung $\widetilde{\mathbf{m}}'^T\,\mathbf{F}\,\widetilde{\mathbf{m}}$ folgt

$$\begin{pmatrix} x' & y' & 1 \end{pmatrix} \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0. \tag{A.17}$$

Sei $\mathbf{f}$ die Schreibweise der Matrix $\mathbf{F}$ in einer Spalte, also $\mathbf{f} = (f_{11}, f_{12}, ..., f_{33})^T$. Es folgt daraus

$$\begin{pmatrix} x'x & x'y & x' & y'x & y'y & y' & x & y & 1 \end{pmatrix} \mathbf{f} = 0. \tag{A.18}$$

Für n korrespondierende Punktpaare gilt entsprechend

$$\begin{pmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{pmatrix} \mathbf{f} = \mathbf{A}\,\mathbf{f} = 0. \tag{A.19}$$

Gesucht ist eine Lösung $\mathbf{f} \neq \mathbf{0}$, welche die Gleichung bei gegebenen Punkten, also der Matrix $\mathbf{A}$, erfüllt. Eine Lösung ist als eine unendliche Menge von Ergebnissen zu verstehen, die sich nur durch einen skalaren Faktor unterscheiden – die Fundamentalmatrix ist skalierungsinvariant. Alle Elemente einer solchen Lösungsmenge stellen die gleiche Abbildung zwischen zwei korrespondierenden Bildpunkten dar.

Da $\mathbf{f}$ 7 Freiheitsgrade besitzt, kann die Gleichung ab einer Anzahl von 7 Punktpaaren gelöst werden. In diesem Fall ist $\text{rang}(\mathbf{A}) = 7$. Basierend auf der Annahme, dass sich mit Hilfe

der Singulärwertzerlegung zwei Vektoren $\mathbf{f}_1$ und $\mathbf{f}_2$ finden lassen, die den Nullraum von $\mathbf{A}$ aufspannen., d.h. die Gleichung $\mathbf{A}\,\mathbf{f} = 0$ lösen. Dieser hat die Form

$$\alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2, \tag{A.20}$$

mit den aus den Vektoren erzeugten Matrizen $\mathbf{F}_1$ und $\mathbf{F}_2$, sowie dem Skalar α. Wegen $\det(\mathbf{F}) = 0$ ist

$$\det(\alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2) = 0, \tag{A.21}$$

was bei Kenntnis von $\mathbf{F}_1$ und $\mathbf{F}_2$ in einem Polynom 3. Grades in α resultiert, mit einer oder drei reellen Lösungen. Sie ergeben somit maximal drei Lösungen der Gleichung

$$\mathbf{F} = \alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2, \tag{A.22}$$

also bis zu drei mögliche Fundamentalmatrizen. Falls mehr als eine Lösung existiert, kann diejenige Fundamentalmatrix als die geeignetste angesehen werden, dessen Epipolarlinien die geringsten Abstände zu den entsprechenden Passpunkten haben.

Da die verwendeten Passpunktpaare u.a. durch Rasterung und Bildrauschen eine Ungenauigkeit besitzen, hat die so errechnete Fundamentalmatrix $\mathbf{F}$ einen gewissen Fehler, der bei nur 7 verwendeten Punktpaaren recht groß werden kann. Sind mehr als 7 Punkte, d.h. $n \geq 8$ Punkte bekannt, lassen sich nach Zhang [Zha98b] die Gleichungen A.14 bzw. A.19 durch anfängliches Ignorieren der Bedingung $\mathrm{rang}(\mathbf{F}) = 2$ in ein Minimierungsproblem zur Suche nach der „besten“ Fundamentalmatrix umwandeln. Dies hat die Form

$$\min_{\mathbf{F}} \sum_{i=1}^{n} \left(\widetilde{\mathbf{m}}_i'^T\,\mathbf{F}\,\widetilde{\mathbf{m}}_i\right)^2 \quad \text{, bzw.} \quad \min_{\mathbf{f}} \|\mathbf{A}\,\mathbf{f}\|^2. \tag{A.23}$$

Zusätzlich wird $\|\mathbf{f}\| = 1$ gefordert, um solche Ergebnisse, die sich nur um einen Faktor unterscheiden, auf eines zu reduzieren und bei der folgenden Eigenwertanalyse keines der Elemente von $\mathbf{F}$ zu bevorzugen. Stellt man den Term des oben gesuchte Minimums um, ist

$$\min_{\mathbf{f}} \mathbf{f}^T\,\mathbf{A}^T\,\mathbf{A}\,\mathbf{f} \tag{A.24}$$

gesucht. Für einen Eigenvektor f der 9×9-Matrix $\mathbf{A}^T\,\mathbf{A}$ ist

$$\mathbf{A}^T\,\mathbf{A}\,\mathbf{f} = \lambda\mathbf{f}, \tag{A.25}$$

der Eigenvektor $\mathbf{f}$ mit dem kleinsten Eigenwert λ minimiert somit den Term.

Praktisch kann dieser Vektor mit Hilfe der Singulärwertzerlegung gefunden werden. In der Zerlegung $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ ist $\mathbf{f}$ ist die letzte Spalte der Rechtssingulärmatrix $\mathbf{V}$.

Nach [Har95a] gibt es die Alternative, anstatt $\|\mathbf{f}\| = 1$ das Matrixelement $f_{33} = 1$ zu setzen und die Methode der kleinsten Quadrate anzuwenden, um das Minimierungsproblem zu lösen.

Die hierbei entstandene Lösung ist ein Vektor $\mathbf{f}$, dessen zugehörige Fundamentalmatrix $\mathbf{F}$ jedoch die Singularitätseigenschaft nicht besitzt, d.h. nicht vom Rang 2 ist. Die mit Hilfe einer solchen Matrix ermittelten Epipolarlinien stellen zwar die Beziehungen der untersuchten Punktpaare dar, müssen sich aber nicht in einem gemeinsamen Epipol schneiden. Dazu muss eine Näherung gefunden werden, welche diese Eigenschaft besitzt.

Es wird eine Schätzung $\mathbf{F}'$ von $\mathbf{F}$ durchgeführt, die Rang 2 besitzt und die Bedingung $\det(\mathbf{F}') = 0$ erfüllt. Dabei soll die Frobeniusnorm $\|\mathbf{F} - \mathbf{F}'\|_F$ minimal sein. Sei $\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ die Singulärwertzerlegung von $\mathbf{F}$ mit der Diagonalmatrix $\mathbf{D} = \mathrm{diag}(r, s, t)$ und $r \geq s \geq t$, so minimiert $\mathbf{F}' = \mathbf{U}\mathbf{D}'\mathbf{V}^T$ mit $\mathbf{D}' = \mathrm{diag}(r, s, 0)$ die Abweichung $\|\mathbf{F} - \mathbf{F}'\|_F$. Die Fundamentalmatrix $\mathbf{F}'$ erfüllt nun die geforderten Eigenschaften.

Hartley [HZ00] nennt diesen *8-Punkte-Algorithmus* eine einfache Methode, die Fundamentalmatrix zu berechnen. Dieses Verfahren wird auch Zhang und Luong et al. [Luo+93] als geeignet präsentiert. Allerdings existiert stets die Tendenz, dass die Epipole in die Nähe der Mitte des jeweiligen Bildes gesetzt werden. Die Normierung der Koordinaten nach Hartley verhindert diesen Seiteneffekt. Darüber hinaus existiert von Zhang [Zha98b] ein iteratives Verfahren, das den Fehler der Fundamentalmatrix noch weiter minimiert, Vergleiche mit der Normierung werden von Hartley [Har95a] durchgeführt. Die Schätzung der Epipole nach Chen et al. [Che+00] bringt ebenfalls optimierte Ergebnisse. Ebenso kann die Robustheit durch das Aussortieren fehlerhafter bzw. ungenauer Passpunktpaare nach Torr und Murray [TM95] verbessert werden. Torr [Tor02] stellt darüber hinaus eine frei verfügbare Software zur Errechnung der Fundamentalmatrix vor.

Verwendete Formelzeichen

In der Arbeit werden, sofern nicht anders erwähnt, folgende Formelzeichen verwendet:

$\mathbf{x} \times \mathbf{y}$	Kreuzprodukt der 3D-Vektoren $\mathbf{x}$ und $\mathbf{y}$.
$\widetilde{\mathbf{x}}$	Punkt in homogenen Koordinaten.
$[\ldots]^T$	Transponierte(r) Vektor bzw. Matrix.
$[\ldots]^{-T}$	Transponierte inverse Matrix. Es ist $[\ldots]^{-T} = ([\ldots]^{-1})^T = ([\ldots]^T)^{-1}$.
$\mathbf{0}$	Nullvektor.
b	Stereoskopische Basis. Horizontaler Abstand der beiden Kameras im idealen Modell.
b_1, b_2	Verzeichnungsparameter für Affinität und Scherung.
d	Disparität. Der Abstand korrespondierender Punkte.
f	Kamerakonstante bzw. Brennweite einer Kamera.
$\mathbf{c}, \mathbf{c}_1, \mathbf{c}_2$	Kamerazentrum bzw. Zentren der linken und rechten Kamera.
$\mathbf{F}$	Die Fundamentalmatrix zur Projektion eines Bildpunktes auf die entsprechende Epipolarlinie des anderen Bildes.
g_8, g_{16}	Disparitätenbild mit 8 bzw. 16 Bit pro Bildpunkt. $g(u, v)$ bezeichnet den Wert eines Bildpunktes des jeweiligen Bildes mit den Koordinaten (u, v).
$\mathbf{I}$	Einheitsmatrix.
I_L	Bild der L-ten Pyramidenebene des Tiefenbildes.
$\mathbf{K}$	Kameramatrix. Enthält Brennweiten und Bildhauptpunkt einer Kamera.
k_1, k_2, k_3	Radial-symmetrische Verzeichnungsparameter.
$M, \mathbf{m}$	Raum- bzw. Bildpunkt bei perspektivischen Projektionen.
$\mathbf{P}$	Abbildungsmatrix.

p_1, p_2 Radial-asymmetrische und tangentiale Verzeichnungsparameter.

$\mathbf{R}$ Rotationsmatrix.

$[\mathbf{R} \quad \mathbf{t}]$ Projektionsmatrix, zusammengesetzte Matrix aus $\mathbf{R}$ und $\mathbf{t}$.

$\mathbf{t}$ Translationsvektor.

t_x, t_y, t_z Elemente des Translationsvektors $\mathbf{t}$.

$[\mathbf{t}]_\times$ Antisymmetrische Kreuzproduktmatrix von $\mathbf{t}$. Es ist

$$[\mathbf{t}]_\times = [(t_x, t_y, t_z)^T]_\times = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix},$$

die Matrix erfüllt für alle 3D-Vektoren $\mathbf{x}$ die Gleichung $[\mathbf{t}]_\times \mathbf{x} = \mathbf{t} \times \mathbf{x}$.

u, v Koordinaten eines Bildpunktes. u ist die horizontale, v die vertikale Komponente.

x_0, y_0 Bildhauptpunkt.

x, y, z x-, y- und z-Koordinaten eines Punktes im dreidimensionalen Raum. Bei Bildkoordinaten ist x die horizontale und y die vertikale Komponente. In Kamerakoordinaten ist z die Entfernung, in Helikopterkoordinaten wird die x-Komponente als Entfernung betrachtet. Darüber hinaus werden die Großbuchstaben X, Y, Z zur Darstellung von 3D-Punkten verwendet.

x_c, y_c, z_c Kamerafeste Koordinaten.

x_f, y_f, z_f Helikopterfeste Koordinaten.

x_g, y_g, z_g Erdfeste Koordinaten.

x_{Lenkung} Hindernisdistanz, bis zu der ein Ausweichen bzw. Anhalten kommandiert wird.

$\hat{x}_{\text{Lenkung}}$ Minimal mögliche Hindernisdistanz zum kollisionsfreien Anhalten bzw. Ausweichen.

ω, φ, κ XYZ-Eulerwinkel, die die Rotation der Kamera beschreiben.

In der Regel sind Skalare mit kursiven Klein- oder Großbuchstaben, Punkte im Raum mit Großbuchstaben bezeichnet. Halbfette Kleinbuchstaben bezeichnen ebenfalls Punkte (Vektoren), halbfette Großbuchstaben Matrizen. Weitere, nur in einem konkreten Zusammenhang verwendete Formelzeichen und Abweichungen von der hier aufgelisteten Darstellung sind im Text erläutert, die Bezeichnungen sind an die verwendete Literatur angelehnt.

Literaturverzeichnis

[Ade+84] Edward H. Adelson, Charles H. Anderson, Peter J. Burt, Joan Odgen, James R. Bergen: *Pyramid methods in image processing.* RCA Engineer 29-6, pp. 33-41, 1984.

[AH88] Nicholas Ayache, Charles Hansen: *Rectification of Images for Binocular and Trinocular Stereovision.* Institut National de Recherche en Informatique et en Automatique (INIRA), 1988.

[Ami96] Omead Amidi: *An Autonomous Vision-Guided Helicopter.* PhD Thesis, Carnegie Mellon University, Pittsburgh, 1996.

[BFB94] John L. Barron, David J. Fleet, Steven S. Beauchemin: *Performance of optical flow techniques.* International Journal of Computer Vision, 12(1), pp. 43-77, 1994.

[Bit05] Walter Bittner: *Flugmechanik der Hubschrauber. Technologie, das flugdynamische System Hubschrauber, Flugstabilitäten, Steuerbarkeit.* Springer, Berlin, 2005.

[BK91] Johann Borenstein, Yoram Koren: *The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots.* IEEE Journal of Robotics and Automation Vol 7, No 3, pp. 278-288, 1991.

[Bla04] Roman Blaschek: *Tiefeninformationsgewinnung aus Stereobildern.* Diplomarbeit, Humboldt-Universität zu Berlin, 2004.

[Böh03] Thomas Böhm: *Erkennung und Verfolgung von Fahrzeugen im Videobild.* Diplomarbeit, DaimlerChrysler AG / Universität Stuttgart, 2003.

[Bou00] Jean-Yves Bouguet: *Pyramidal Implementation of the Lucas Kanade Feature Tracker.* Description of the algorithm, Intel Corporation, 2000.

[BT96] Stan Birchfield, Carlo Tomasi: *Depth Discontinuities by Pixel-to-Pixel Stereo.* Computer Science Department, Stanford University, 1996.

[Buc02] Jörg J. Buchholz: *Regelungstechnik und Flugregler.* Vorlesungsskript, Hochschule Bremen, 2002.

[CE04] Terry D. Cornall and Greg K. Egan: *Measuring Horizon Angle from Video on a Small Unmanned Aerial Vehicle.* 2nd International Conference on Autonomous Robots and Agents, 2004.

[CE05] Terry D. Cornall and Greg K. Egan: *Heaven and Earth: How to tell the difference.* 11th Australian International Aerospace Congress, 2005.

[Che+00] Zezhi Chen, Chengke Wu, Peiyi Shen, Yong Liu, Long Quan: *A robust algorithm to estimate the fundamental matrix.* Pattern Recognition Letters 21, pp. 851-861, 2000.

[Cra02] Michael Cramer: *GPS/inertial data in aerial photogrammetry.* Universität Stuttgart, 2002. `http://www.ifp.uni-stuttgart.de/forschung/photo/georef-Dateien/georef.en.html`

[DC1394] Project: *1394-based DC Control Library.* Copyright © 2005, Open Source Technology Group. `http://sourceforge.net/projects/libdc1394/`

[DKL98] Erik B. Dam, Martin Koch, Martin Lillholm: *Quaternions, Interpolation and Animation.* Technical Report, Københavns Universitet, 1998.

[Doh+00] Patrick Doherty et al.: *The WITAS Unmanned Aerial Vehicle Project.* Proc. of the 14th European Conference on Artificial Intelligence, Berlin, pp. 747-755, 2000.

[Dör05] Thomas Döring: *Verfolgung von Verkehrsobjekten aus verschiedenen Kameraperspektiven.* Diplomarbeit, DLR / Universität Stuttgart, 2005.

[Elf87] Alberto Elfes: *Sonar-based Real-World Mapping and Navigation.* IEEE Journal of Robotics and Automation, Vol. RA-3, No 3, pp. 249-265, 1987.

[Fab02] P. Fabiani: *Challenges of the autonomous air vehicle project of ONERA.* Programme national robotique et entités artificielles, 2002.

[FB86] John G. Fryer and Duane C. Brown: *Lens Distortion for Close-Range Photogrammetry.* Photogrammetric Engineering and Remote Sensing, 52(1), pp. 51-58, 1986.

[FG87] Wolfgang Förstner and Eberhard Gülch: *A fast operator for detection and precise location of distinct points, corners and centres of circular features.* In ISPRS Intercommission Workshop, Interlaken, 1987.

[Flo05] Stefan Florczyk: *Robot Vision. Video-based Indoor Exploration with Autonomous and Mobile Robots.* Wiley, Weinheim, 2005.

[FLTK] *Fast Light Toolkit.* Copyright © 1991, Free Software Foundation, Inc. `http://www.fltk.org`

[För99] Wolfgang FÖRSTNER: *On Estimating Rotations.* Institut für Photogrammetrie, Universität Bonn, 1999.

[For+04] Sven FORSTMANN, Jun OHYA, Yutaka KANOU, Alfred SCHMITT, and Sven THÜRING. Real-time stereo by using dynamic programming. *CVPR Workshop on real-time 3D sensors and their use, 2004.*

[FTV97] Andrea FUSIELLO, Emanuele TRUCCO, Alessandro VERRI: *Rectification with unconstrained stereo geometry.* Proceedings of the Eighth British Machine Vision Conference, pp. 400-409, 1997.

[Fua91] Pascal FUA: *A parallel stereo algorithm that produces dense disparity maps and preserves image features.* Rapports de Recherche No. 1369, INRIA, 1991.

[God02] Robert GODDING: *Geometrische Kalibrierung und Orientierung digitaler Bildaufnahmesysteme.* AICON 3D Systems GmbH, 2002.

[Goo04] Lukas GOORMANN: *Objektorientierte Bildverarbeitungsalgorithmen zum relativen Hovern eines autonomen Helikopters.* Diplomarbeit, DLR / FH Braunschweig/Wolfenbüttel, 2004.

[Gut04] Olaf GUTH: *Biologisch inspirierte Bildverarbeitungsalgorithmen zur Realisierung eines Geländefolgfluges für einen autonomen Kleinhubschrauber.* Diplomarbeit, DLR / FH Braunschweig/Wolfenbüttel, 2004.

[Gou01] Xavier GOURDON: *Newton's method and high order iterations,* 2001. `http://numbers.computation.free.fr/Constants/Algorithms/newton.html`

[GW02] Rafael C. GONZALEZ, Richard E. WOODS: *Digital Image Processing, 2nd Edition.* Prentice Hall, New Jersey, 2002.

[Har95a] Richard HARTLEY: *In defence of the 8-point algorithm.* Proc. of the 5th International Conference on Computer Vision, Boston MA. IEEE Computer Society Press, pp. 1064-1070, 1995.

[Har95b] Richard HARTLEY: *Theory and practice of projective rectification.* Technical Report 2538, INRIA, 1995.

[Hei05] Lucas HEITZMANN-GABRIELLI: *Mission Planning and Dynamic Avoidance for Groups of Autonomous Agents.* DLR / São José dos Campos, 2005.

[Hom+03] Günter HOMMEL et al.: MARVIN – An Autonomously Operating Flying Robot. Technische Universität Berlin, 2003. `http://pdv.cs.tu-berlin.de/MARVIN/`

[Hra06] Stefan Hrabar: *Vision-Based 3D-Navigation for an Autonomous Helicopter.* PhD Thesis, University of Southern California, 2006.

[HS81] Berthold K.P. Horn and Brian G. Schunck. *Determining Optical Flow.* Artificial Intelligence, 17, pp. 185-203, 1981.

[HS88] Chris Harris, Mike Stephens: *A Combined Corner and Edge Detector.* Plessey Research Roke Manor, UK. The Plessey Company, 1988.

[HS97] Janne Heikkilä and Olli Silvén: *A Four-step Camera Calibration Procedure with Implicit Image Correction.* Conference on Computer Vision and Pattern Recognition (CVPR'97), pp. 1106-1112, 1997.

[HSC+05] Stefan Hrabar, Gaurav S. Sukhatme, Peter Corke, Kane Usher, Jonathan Roberts: *Combined Optic-Flow and Stereo-Based Navigation of Urban Canyons for a UAV.* Proceedings of the IEEE International Conference on Intelligent Robots and Systems, 2005.

[HZ00] Richard Hartley, Andrew Zisserman: *Multiple View Geometry in Computer Vision.* University Press, Cambridge, 2000.

[Jäh02] Bernd Jähne: *Digitale Bildverarbeitung.* Springer, Heidelberg, 2002.

[Jen96] Cullen Jennings: *Structure and Motion from Stereo Image Sequences.* University of Calgary, 1996.

[JS02] Eric N. Johnson and Daniel P. Schrage: *The Georgia Tech Unmanned Aerial Research Vehicle: GTMax.* School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, 2002.

[Kan+96] Takeo Kanade et al.: *A video-rade stereo machine and its new applications.* Proceedings of 15th Computer Vision and Pattern Recognition Conference (CVPR), San Francisco, 1996.

[KB04] Kurt Konolige, David Beymer: *Small Vision System – Calibration Addendum to the User's Manual.* Software version 3.2g. SRI International, Menlo Park, 2004.

[KB05] Kurt Konolige, David Beymer: *Small Vision System – User's Manual.* Software version 4.1e. SRI International, Menlo Park, 2005.

[KL05] Viktor Kaufmann, Richard Ladstädter: *Elimination of Color Fringes in Digital Photographs Caused by Lateral Chromatic Aberration.* CIPA XX International Symposium, 2005.

[Kra97] Karl Kraus: *Photogrammetrie, Band 1. Grundlagen und Standardverfahren.* Dümmler, Bonn 1997.

[Kra04] Christian KRAUSE: *Hindernisvermeidung mit Hilfe stereoskopischer Bilder.* Diplomarbeit, DLR / BA Mannheim, 2004.

[LK81] Bruce D. LUCAS, Takeo KANADE: *An Iterative Image Registration Technique with an Application to Stereo Vision.* International Joint Conference on Artificial Intelligence, pp. 674-679, 1981.

[Lon81] H.C. LONGUET-HIGGINS: *A computer algorithm for reconstructing a scene from two projections.* Nature 293, pp. 133-135, September 1981.

[Luh00] Thomas LUHMANN: *Nahbereichsphotogrammetrie: Grundlagen, Methoden und Anwendungen.* Wichmann, Heidelberg, 2000.

[Luo+93] Quang-Tuan LUONG, Rachid DERICHE, Olivier FAUGERAS und Theo PAPADOPOULO: *On determining the fundamental matrix: analysis of different methods and experimental results.* Institut National de Recherche en Informatique et en Automatique (INIRA), 1993.

[MH80] D. MARR and E. HILDRETH: *Theory of edge detection.* Proceeding of the Royal Society London, B207, pp. 187-217, 1980.

[ML98] Don MURRAY, Jim LITTLE: *Using real-time stereo vision for mobile robot navigation.* University of British Columbia, 1998.

[MM04] Javier MINGUEZ and Luis MONTANO: *Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios.* IEEE Transactions on Robotics and Automation, Vol. 20, No. 1, 2004.

[Mon05] Michael MONTEMERLO: *Stanford Racing Team's Entry In The 2005 DARPA Grand Challenge.* Stanford Artificial Intelligence Laboratory, 2005.

[MSK+04] Yi MA, Stefano SOATTO, Jana KOŠECKÁ, S. Shankar SASTRY: *An Invitation to 3-D Vision. From Images to Geometric Models.* Springer, New York, 2004.

[Nec+05] Michael C. NECHYBA et al.: *Sky/Ground Segmentation.* Centre for MAV Research, University of Florida, 2005. `http://www.mil.ufl.edu/mav/research/vision/skygroundseg/`

[OpenCV] *Open Source Computer Vision Library.* Copyright © 2000, Intel Corporation, all rights reserved. `http://www.intel.com/technology/computing/opencv/`

[Ora01] Daniel ORAM: *Rectification for Any Epipolar Geometry.* Proceedings of the 12th British Machine Vision Conference, pp. 653-662, 2001.

[Reu05] Ralf REULKE: *Stereo-Bildverarbeitung.* Vorlesungsskript, HU Berlin, 2005.

[RW97] Lennart Råde, Bertil Westergren: *Springers Mathematische Formeln.* Springer, Berlin, 1997.

[SB95] S.M. Smith, J.M. Brady: *SUSAN – A New Approach to Low Level Image Processing.* Technical Report, Oxford University, 1995.

[Sch97] Jörge Schulz zur Wiesch: *Grundlagen der 3D-Wahrnehmung,* 1997. `http://www.jszw.de/3d_wahrnehmung/tiefe.html`

[SMS03] Srikanth Saripalli, James F. Montgomery and Gaurav S. Sukhatme: *Visually-Guided Landing of an Unmanned Aerial Vehicle.* IEEE Transactions on Robotics and Automation, Vol. 13, No. 3, pp. 371-381, 2003.

[SS99] Daniel Scharstein and Richard Szeliski. *Stereo matching with nonlinear diffusion.* International Journal of Computer Vision, 28(2):155-174, 1998.

[SS00] Linda G. Shapiro, George C. Stockman: *Computer Vision.* Prentice Hall, New Jersey, 2000.

[SS01] Daniel Scharstein, Richard Szeliski: *A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms.* Technical Report, Microsoft Research, 2001.

[SS03] Daniel Scharstein, Richard Szeliski: *Stereo Vision Research Page.* Middlebury College, 2003. `http://www.middlebury.edu/stereo`

[SSS01] Courtney S. Sharp, Omid Shakernia, and S. Shankar Sastry: *A vision system for landing an unmanned aerial vehicle.* Proceedings of IEEE International Conference on Robotics and Automation, pp. 1720-1728, 2001.

[ST94] Jianbo Shi and Carlo Tomasi: *Good Features to Track.* IEEE Conference on Computer Vision and Pattern Recognition, 1994.

[Suk+05] Gaurav S. Sukhatme et al.: *USC Autonomous Flying Vehicle Project – Videos.* University of Southern California, 2005. `http://www-robotics.usc.edu/~avatar/videos.htm`

[SVS] *Small Vision System.* Copyright © 2004, SRI International. `http://www.ai.sri.com/software/SVS`

[Tay03] Camillo J. Taylor: *Surface Reconstruction from Feature Based Stereo.* IEEE International Conference on Computer Vision, 2003.

[TDB04] Frank Thielecke, Jörg S. Dittrich, Andreas Bernatz: *ARTIS – ein VTOL UAV Demonstrator.* DLR, Institut für Flugsystemtechnik, Braunschweig, 2004.

[TM95] Philip H. S. Torr, David W. Murray: *Outlier Detection and Motion Segmentation.* Robotics Research Group, Department of Engineering Science, University of Oxford, 1995.

[Tor02] Philip H. S. Torr: *A Structure and Motion Toolkit in Matlab.* Technical Report, Microsoft Research, 2002. Software unter `http://cms.brookes.ac.uk/staff/PhilipTorr/`

[Tsa87] Roger Y. Tsai: *A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses.* IEEE Journal of Robotics and Automation, Vol. RA-3(4), pp. 323-344, 1987.

[Urm05] Christopher Urmson: *Navigation Regimes for Off-Road Autonomy.* PhD Thesis, Carnegie Mellon University, Pittsburgh, 2005.

[Vid04] Videre Design: *STH-MDCS-VAR/-C Stereo Head – User's Manual.* Rev. 1.02. Videre Design, Menlo Park, 2004.

[WB92] H. Wang, J.M. Brady: *Corner detection with subpixel accuracy.* Technical Report OUEL 1925/92, University of Oxford, 1992.

[Wil98] Todd A. Williamson: *A High-Performance Stereo Vision System for Obstacle Detection.* PhD Thesis, Carnegie Mellon University, Pittsburgh, 1998.

[Wil03] Todd Will: *Introduction to the Singular Value Decomposition.* University of Wisconsin – La Crosse, 2003. `http://www.uwlax.edu/faculty/will/svd/`

[Wre99] Helmut J. Wresnik: *Darstellende Geometrie.* Vorlesungsskript, FH Johanneum / TU Graz, 1999. `http://www.geometrie.tugraz.at/wresnik/`

[Zha98a] Zhengyou Zhang: *A Flexible New Technique for Camera Calibration.* Technical Report, Microsoft Research, 1998.

[Zha98b] Zhengyou Zhang: *Determining the Epipolar Geometry and its Uncertainty: A Review.* International Journal of Computer Vision, 27(2), pp. 161–198, 1998.