

Bestimmung der relativen Kameraorientierung aus Bildpaaren

Bachelorarbeit

an der Fakultät für Informations-, Medien und Elektrotechnik der
Fachhochschule Köln im Institut für Medien- und Phototechnik

Autor

Patrick Stein

Mat.-Nr. 11052174

Referent: Prof. Dr. rer. nat. Dietmar Kunz

Korreferent: Prof. Dr.-Ing. Gregor Fischer

Köln, Januar 2009

Determination of the relative camera orientation from image pairs

Bachelorthesis

at the Faculty of Information, Media and Electrical Engineering at the
Cologne University of Applied Sciences in the Institute of Media and
Imaging Technology

Author

Patrick Stein

Mat.-No. 11052174

First Reviewer: Prof. Dr. rer. nat. Dietmar Kunz

Second Reviewer: Prof. Dr.-Ing. Gregor Fischer

Cologne, January 2009

Kurzbeschreibung

Titel: Bestimmung der relativen Kameraorientierung aus Bildpaaren

Autor: Patrick Stein

Referenten: Prof. Dr. rer. nat. Dietmar Kunz
Prof. Dr.-Ing. Gregor Fischer

Zusammenfassung:

In dieser Bachelorarbeit wird ein Verfahren zur Bestimmung von relativen Kameraparametern eines Stereosystems vorgestellt. Die Methode basiert auf Informationen von Punktkorrespondenzen aus Bildpaaren. Der Algorithmus wurde in der Programmiersprache Java implementiert.

Stichwörter: Epipolargeometrie, Fundamental-Matrix, 8-Punkt-Algorithmus, Bildverarbeitung, ImageJ

Sperrvermerk:

Kein Sperrvermerk

Datum: 20.01.2009

Abstract

Title: Determination of the relative camera orientation from image pairs

Author: Patrick Stein

Advisores: Prof. Dr. rer. nat. Dietmar Kunz
Prof. Dr.-Ing. Gregor Fischer

Abstract: This bachelor thesis introduces a technique to determine the relative camera parameters of a stereo system. The method is based on point correspondences of image pairs. The algorithm is implemented in Java.

Keywords: Epipolar Geometry, Fundamental Matrix, 8-point algorithm, Image Processing, ImageJ

Remark of closure:

None

Date: 20.01.2009

Inhaltsverzeichnis

1 Einleitung	1
2 Theoretische Grundlagen	3
2.1 Projektive Geometrie	3
2.1.1 Projektiver Raum	4
2.1.2 Projektive Ebene	5
2.2 Kameramodell	6
2.2.1 Lochkameramodell	6
2.2.2 Externe Transformation	8
2.2.3 Perspektivische Transformation	9
2.2.4 Interne Transformation	10
2.3 Epipolargeometrie	12
2.3.1 Aufbau der Epipolargeometrie	13
2.3.2 Essential-Matrix	14
2.3.3 Fundamental-Matrix	16
3 Schätzung der Fundamental-Matrix	18
3.1 Lineare Verfahren	18
3.2 8-Punkt-Algorithmus	20
3.2.1 Lineare Methode des kleinsten quadratischen Fehlers	21
3.2.2 Auswertung der Eigenwerte	22
4 Implementierung	24
4.1 Software	24
4.2 Aufbau des Programms	25
4.2.1 Die Klasse KorrespondierendePunkte	26
4.2.2 Die Klasse EpipolargeometriePixel	28

4.2.3 Die Klasse Fehlerrechnung	30
4.2.4 Die Klasse EpipolargeometrieKamera.....	32
4.2.5 Die Klasse Ausgabe	33
4.3 Visualisierung der Epipolarlinien.....	33
5 Ergebnisse	35
5.1 Visualisierung der Epipolarlinien.....	35
5.2 Abweichung der Epipolarlinien.....	40
5.3 Beurteilung der Essential-Matrix	42
6 Diskussion	44
Literaturverzeichnis	46
Abbildungsverzeichnis	48
Eidesstattliche Erklärung	50
Sperrvermerk	51
Weitergabeerklärung	52

1 Einleitung

Die ersten Verfahren, eine dreidimensionale Szene auf eine zweidimensionale Ebene zu projizieren, bestehen schon seit dem frühen Mittelalter. Aus dieser Zeit stammt auch das erste bekannte Kameramodell, die sogenannte Lochkamera. Dieses Modell wurde bis zur heutigen Zeit immer weiter entwickelt und der Umgang mit digitalen Bildern gehört für viele Menschen zum Alltag. Dabei werden die Bilder mit Hilfe von Computersystemen wiedergegeben und verarbeitet. Die Bildanalyse, oder auch Computer Vision genannt, befasst sich mit der Problematik, natürliche Bilder mittels eines Computers auszuwerten und zu interpretieren. Dazu gehört zum Beispiel die Segmentierung von Objekten und deren quantitativen Auswertung (Medizin), sowie die qualitative Auswertung (Qualitätsprüfung), aber auch die Erfassung von Bewegungen und räumlichen Szenen (Robotik). Diesem Fachgebiet kommt heute in fast allen industriellen Branchen eine große Bedeutung zu.

Die notwendigen Algorithmen basieren auf komplexen mathematischen Modellen, haben aber gleichzeitig die Anforderung einer schnellen und robusten Verarbeitung. Die Anwendung im Bereich der Videotechnik verkürzt den benötigten Zeitrahmen auf den Bruchteil einer Sekunde. Außerdem liegen die Fähigkeiten des maschinellen Sehens weit unter denen des menschlichen Sehsystems.

Somit besteht in der Bildanalyse ein großer Bedarf zur Optimierung und Anpassung. Sie ist damit Bestandteil in der aktuellen Forschung und Entwicklung [1, 3, 7, 8, 9].

Ziel dieser Arbeit ist es, mit Hilfe von Algorithmen der Stereobildanalyse, Informationen der Kameraparameter zu berechnen. Grundlage hierfür sind Bildpaare, die eine Szene aus zwei unterschiedlichen Perspektiven zeigen. Die Algorithmen werden in der Programmiersprache Java implementiert und mit dem Open-Source-Bildverarbeitungsprogramm ImageJ getestet.

Allerdings muss man bei dieser Projektion in Kauf nehmen, dass die Objekte nicht maßgetreu abgebildet werden [2]. So werden Längen und Winkel des Objekts in der Abbildung verzerrt dargestellt. Je weiter sich ein Objekt vom Betrachter befindet, desto kleiner wird es auf der Bildebene abgebildet. Dadurch werden gleichgroße Objekte in der Szene nicht gleichgroß auf der Bildebene abgebildet, wenn sie einen unterschiedlich großen Abstand zum Betrachter haben [1].

2.1.1 Projektiver Raum

Um die projektive Geometrie mathematisch zu beschreiben, wird der projektive Raum eingeführt. Dieser wird mit Hilfe von homogenen Koordinaten beschrieben [1]. Dabei kann jeder beliebige Vektor in homogene Koordinaten¹ transformiert werden, indem er um eine Dimension erweitert wird [3].

$$m = [m_1, m_2, \dots, m_n]^T, \quad \Rightarrow \quad \tilde{m} = [m_1, m_2, \dots, m_n, m_{n+1}]^T \quad (1)$$

Daraus ergibt sich, dass homogene Koordinaten bis auf ein Vielfaches definiert sind. Normiert man die letzte Komponente auf den Wert Eins, repräsentieren die weiteren Werte die Koordinaten für den Unterraum.

¹ Im Weiteren werden Vektoren in homogenen Koordinaten mit einem „~“ gekennzeichnet

$$\tilde{m} = [m_1, m_2, \dots, m_n, m_{n+1}]^T \Rightarrow m = \left[\frac{m_1}{m_{n+1}}, \frac{m_2}{m_{n+1}}, \dots, \frac{m_n}{m_{n+1}} \right] \quad (2)$$

Durch die Verwendung von homogenen Koordinaten, kann ein Punkt der Dimension n verschoben werden, indem er mit einer Transformationsmatrix² der Dimension $n+1$ multipliziert wird.

$$\tilde{m}_2^n = P^{n+1} \tilde{m}_1^n \quad (3)$$

Dieser Vorgang ist unter dem Namen Homographie bekannt [1]. Ohne homogene Koordinaten, ist dies nur mit einer Vektoraddition möglich [3].

2.1.2 Projektive Ebene

Für die zweidimensionale Ebene werden die Koordinaten eines Punktes folgendermaßen in Koordinaten für die projektive Ebene überführt:

$$m = [x, y]^T \rightarrow \tilde{m} = [x, y, 1]^T \quad (4)$$

Über die Homographie können Punkte von einer Ebene in eine andere Ebene transformiert werden. Die Transformationsmatrix, oder auch

² Im Weiteren werden Matrizen mit Großbuchstaben gekennzeichnet

Homographie-Matrix, ist für die projektive zweidimensionale Ebene eine 3x3 Matrix und die Gleichung lautet [1]:

$$\tilde{m}_2 = H\tilde{m}_1 \quad \Rightarrow \quad \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \quad (5)$$

2.2 Kameramodell

Für die vollständige Beschreibung einer Projektion müssen jetzt noch die Parameter der Kamera mathematisch erläutert werden. Das einfachste Modell einer zweidimensionalen Abbildung von einer dreidimensionalen Szene bietet die Lochkamera.

2.2.1 Lochkameramodell

Bei einer Lochkamera werden die Punkte der dreidimensionalen Szene über das Projektionszentrum (Brennpunkt) auf die Bildebene projiziert. Für eine einfachere Darstellung wird das Projektionszentrum, im Gegensatz zu einer realen Kamera, hinter die Bildebene gelegt. Abbildung 2 zeigt die Projektion m eines dreidimensionalen Punktes M_w auf die Bildebene I mit dem Prinzip des Lochkameramodells. Der

Punkt C stellt das Projektionszentrum dar und liegt im Ursprung des Kamerakoordinatensystems [1].

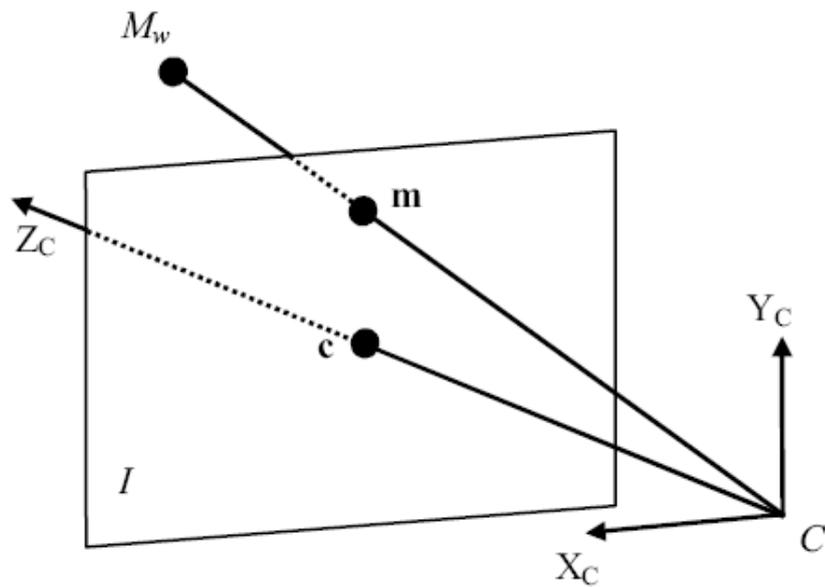


Abbildung 2: Schematischer Aufbau des Lochkameramodells [1]

Für die mathematische Beschreibung der Projektion sind drei Transformationen notwendig. Eine externe, eine perspektivische und eine interne.

2.2.2 Externe Transformation

Der Punkt M_w in Abbildung 2 liegt im Weltkoordinatensystem und wird in Kamerakoordinaten beschrieben, indem man das Koordinatensystem des Punktes so transformiert, dass es in dem der Kamera liegt.

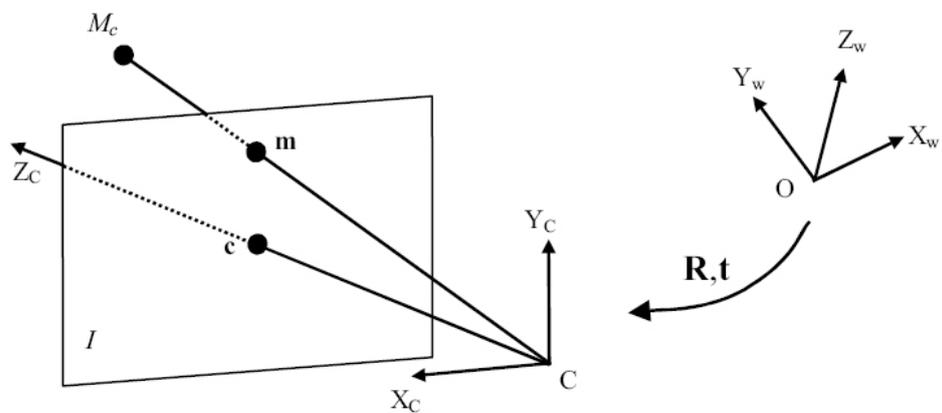


Abbildung 3: Externe Transformation [1]

Diese Transformation wird mit Hilfe einer Rotation R und einer Translation t erreicht und der im Kamerakoordinatensystem liegende Punkt M_c wird durch folgende Gleichung in homogenen Koordinaten berechnet:

$$\tilde{M}_c = D\tilde{M}_w \quad \text{mit} \quad D = \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \quad \text{und} \quad 0_3 = [0,0,0]^T \quad (6)$$

Dabei ist D die extrinsische Matrix und sie beschreibt die externen Parameter einer Kamera [1].

2.2.3 Perspektivische Transformation

Die Darstellung des dreidimensionalen Punktes $M_c = (X, Y, Z)$ in zweidimensionalen Koordinaten des Sensors³ $m' = (x, y)$, wird über die Zentralprojektion beschrieben. Dabei gelten nach dem Strahlensatz folgende Teilverhältnisse, wobei f die Brennweite angibt [1, 4].

$$\frac{x}{X} = \frac{y}{Y} = \frac{f}{Z} \quad (7)$$

So ergibt sich die Transformation von Kamerakoordinaten in Sensorkoordinaten durch folgende Gleichung:

$$s\tilde{m}' = P'\tilde{M}_c \Rightarrow \begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad \text{mit } s = S \quad (8)$$

³ Im Weiteren werden Vektoren in Sensorkoordinaten mit einem „ $\tilde{\prime}$ “ gekennzeichnet

Durch Normierung von S auf den Wert Eins, erhält man aus den homogenen Koordinaten von \tilde{m}' die Sensorkoordinaten x und y .

$$x = \frac{U}{S}, \quad y = \frac{V}{S} \quad (9)$$

Ersetzt man \tilde{M}_c aus Gleichung (8) mit dem Ergebnis aus Gleichung (6), erhält man die Transformation, um einen Punkt aus dem Weltkoordinatensystem auf das Sensorkoordinatensystem abzubilden.

$$s\tilde{m}' = P'D\tilde{M}_w \quad (10)$$

Die Matrix P' wird perspektivische Projektionsmatrix genannt [1].

2.2.4 Interne Transformation

Um den Punkt in Pixelkoordinaten auszudrücken, muss er noch ein weiteres Mal transformiert werden. Diese Transformation besteht aus zwei Schritten. Zuerst wird der Punkt in beide Koordinatenrichtungen mit den Faktoren k_u und k_v skaliert, so dass er nicht mehr mit einem metrischen Wert beschrieben wird, sondern in Pixelkoordinaten [1].

Da der Ursprung des Bildkoordinatensystems links oben liegt, muss der Punkt noch um die Werte u_0 und v_0 verschoben werden⁴ [3]. Damit erhält man die Transformationsmatrix

$$H = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

und die Gleichung für die Transformation von Sensorkoordinaten in Bildkoordinaten lautet:

$$\tilde{m} = H\tilde{m}' \quad (12)$$

Die gesamte Transformation sieht dann wie folgt aus:

$$s\tilde{m}' = HP'D\tilde{M}_w \quad \text{mit} \quad HP' = \begin{bmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (13)$$

Setzt man die Brennweite auf den Wert Eins, erhält man die normierten Koordinaten des Bildes. Für die Projektionsmatrix gilt:

$$P_N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (14)$$

⁴ Die Bildkoordinaten werden im Weiteren mit u für die horizontale Achse und v für die vertikale Achse angegeben.

Die Matrix HP' aus Gleichung (13) wird dann umgeschrieben als:

$$HP' = AP_N \quad \text{mit } A = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (15)$$

Die Matrix A enthält die internen Parameter einer Kamera und wird intrinsische Matrix genannt. Mit normierten Koordinaten lautet die Transformation eines dreidimensionalen Punktes auf die zweidimensionale Bildebene in Pixelkoordinaten dann [1]:

$$s\tilde{m}' = AP_N D\tilde{M}_w \quad (16)$$

2.3 Epipolargeometrie

Die geometrischen Zusammenhänge von zwei Bildern, die aus unterschiedlichen Perspektiven erstellt wurden, werden mit der Epipolargeometrie zusammengefasst. Diese Zusammenhänge können mit einer 3x3 Matrix beschrieben werden. Kennt man die intrinsischen Parameter, können die Korrespondenzen in Sensorkoordinaten beschrieben werden und die Matrix heißt Essential-Matrix. Sind diese Parameter nicht bekannt, werden die Beziehungen in Bildkoordinaten angegeben und die Matrix heißt Fundamental-Matrix [1].

2.3.1 Aufbau der Epipolargeometrie

Bei der allgemeinen Epipolargeometrie geht man davon aus, dass eine Szene aus zwei verschiedenen Perspektiven aufgenommen wird. Dabei werden die Kameras verschoben und zueinander verdreht. Abbildung 4 zeigt den geometrischen Aufbau der Epipolargeometrie und deren wichtigsten Parameter, welche im Folgenden erläutert werden.

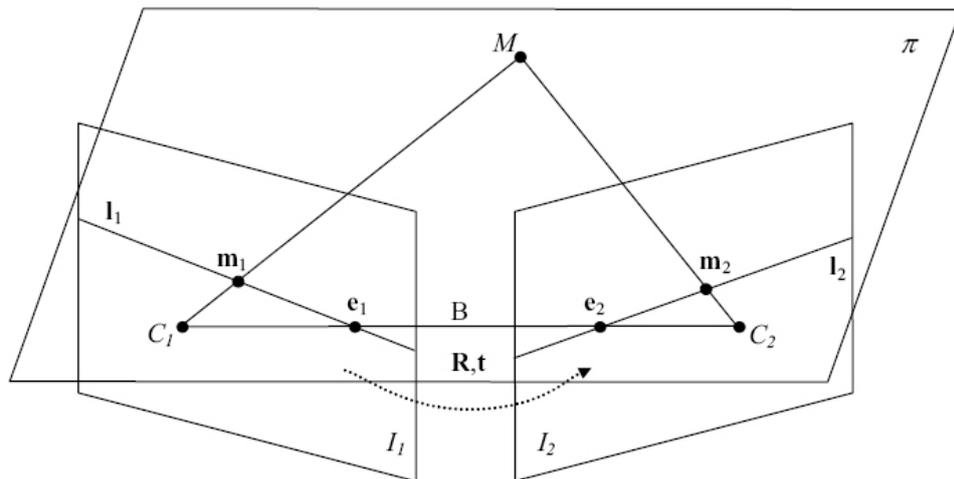


Abbildung 4: Aufbau der Epipolargeometrie [1]

Die beiden optischen Zentren von Kamera 1 und Kamera 2 werden mit C_1 und C_2 bezeichnet und die Gerade, die sie verbindet, nennt man Basislinie, B . Die Schnittpunkte dieser Linie mit den Bildebenen I_1 und I_2 sind die sogenannten Epipole. Diese werden mit e_1 und e_2 gekennzeichnet. Der Punkt M ist ein dreidimensionaler Punkt, mit seinen Abbildungen auf der zweidimensionalen Ebene, m_1 und m_2 . Der Punkt

M und die beiden Punkte der optischen Zentren C_1 und C_2 spannen eine Ebene π auf, die Epipolarebene genannt wird. Diese Ebene schneidet die beiden Bildebenen mit den Geraden l_1 und l_2 . Diese Schnittgeraden werden Epipolarlinien genannt. Die Verschiebung und Drehung der zwei Kameras, wird mit einer Rotationsmatrix R und einem Verschiebungsvektor t beschrieben [1].

2.3.2 Essential-Matrix

Setzt man den Ursprung des Kamerakoordinatensystems in das Koordinatensystem von Kamera 1, wird ein dreidimensionaler Punkt M folgendermaßen von Kamera 1 zu Kamera 2 transformiert [6]:

$$M_2 = R M_1 + t \quad (17)$$

Aus Gleichung (10) ergibt sich für die Abbildung in Sensorkoordinaten:

$$s_2 \tilde{m}'_2 = R s_1 \tilde{m}'_1 + t \quad (18)$$

Erweitert man beide Seiten dieser Gleichung mit dem Kreuzprodukt von t , dann ergibt sich:

$$s_2 t \times \tilde{m}'_2 = s_1 t \times R \tilde{m}'_1 \quad (19)$$

Um die beiden Parameter s_1 und s_2 zu eliminieren, wird die Gleichung noch mit $\tilde{m}'_2{}^T$ multipliziert und man erhält:

$$s_2 \tilde{m}'_2{}^T \cdot t \times \tilde{m}'_2 = s_1 \tilde{m}'_2{}^T \cdot t \times R \tilde{m}'_1 \quad (20)$$

Da der Vektor $t \times \tilde{m}'_2$ senkrecht auf \tilde{m}'_2 steht, ist das Skalarprodukt $\tilde{m}'_2{}^T \cdot t \times \tilde{m}'_2$ gleich Null.

Somit kann man die Gleichung umformen zu [6]:

$$\tilde{m}'_2{}^T \cdot t \times \tilde{m}'_2 = \tilde{m}'_2{}^T \cdot t \times R \tilde{m}'_1 = 0 \quad (21)$$

Schreibt man jetzt noch das Kreuzprodukt der Vektoren t und $R\tilde{m}'_1$ in ein Matrixprodukt um, so erhält man aus dem zweiten Teil der Gleichung die sogenannte Epipolargleichung [1]:

$$\tilde{m}'_2{}^T [t]_{\times} R \tilde{m}'_1 = 0 \quad \text{mit } [t]_{\times} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}_{\times} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \quad (22)$$

Dabei ist das Ergebnis der Multiplikation von $[t]_{\times}$ mit R die Essential-Matrix und die Gleichung lautet:

$$\tilde{m}'_2{}^T E \tilde{m}'_1 = 0 \quad \text{mit } E = [t]_{\times} R \quad (23)$$

Dabei beschreibt die Essential-Matrix die Transformation der Kameras [1]. Mit dieser Matrix lassen sich die beiden Epipole berechnen.

$$E \tilde{e}'_1 = 0 \quad \text{und} \quad E^T \tilde{e}'_2 = 0 \quad (24)$$

Außerdem können die Epipolarlinien zu einem Punkt in beiden Ansichten bestimmt werden. Der jeweils korrespondierende Punkt muss dann auf dieser Geraden liegen [1].

$$l'_1 = E^T \tilde{m}'_2 \quad \text{und} \quad l'_2 = E \tilde{m}'_1 \quad (25)$$

2.3.3 Fundamental-Matrix

Wie in Kapitel 2.2.4 beschrieben, wird ein Punkt in Sensorkoordinaten mit Hilfe einer Transformationsmatrix in Pixelkoordinaten ausgedrückt. So werden die Punkte \tilde{m}'_1 und \tilde{m}'_2 transformiert durch:

$$\tilde{m}_1 = A_1 \tilde{m}'_1 \quad \text{und} \quad \tilde{m}_2 = A_2 \tilde{m}'_2 \quad (26)$$

Stellt man diese Gleichung nach den Punkten in Sensorkoordinaten (\tilde{m}'_1 und \tilde{m}'_2) um, und setzt diese in die Gleichung (23) ein, erhält man [1]:

$$\tilde{m}_2^T A_2^{-T} E A_1^{-1} \tilde{m}_1 = \tilde{m}_2^T F \tilde{m}_1 = 0 \quad (27)$$

Wobei die Matrix $F = A_2^{-T} E A_1^{-1}$ die Fundamental-Matrix ist, und die intrinsischen und extrinsischen Parameter der Kamera enthält. Damit kann die Epipolargeometrie in Pixelkoordinaten beschrieben werden und die Epipole werden berechnet durch:

$$F \tilde{e}_1 = 0 \quad \text{und} \quad F^T \tilde{e}_2 = 0 \quad (28)$$

Die Epipolarlinien in Pixelkoordinaten zu einem Punkt lassen sich errechnen mit:

$$l_2 = F \tilde{m}_1 \quad \text{und} \quad l_1 = F^T \tilde{m}_2 \quad (29)$$

Da die Determinante der Schiefsymmetrischen Matrix $[t]_{\times}$ gleich Null ist, wird die Determinante von E , und somit auch von F , ebenfalls zu Null. Demnach besitzt die Fundamental-Matrix den Rang 2 und verliert dadurch einen Freiheitsgrad. Außerdem ist diese Matrix eine projektive Größe und deshalb nur bis auf einen Skalierungsfaktor bestimmt. Infolgedessen werden die unabhängigen Parameter auf sieben reduziert, womit die Fundamental-Matrix sieben Freiheitsgrade besitzt [1].

3 Schätzung der Fundamental-Matrix

Aus den vorherigen Kapiteln geht hervor, dass man mit den Parametern der Kameras, die Epipolargeometrie des Stereosystems, über die Essential-Matrix bzw. die Fundamental-Matrix, bestimmen kann. Es besteht jedoch die Möglichkeit, die Fundamental-Matrix auch ohne diese Parameter, nur aus den Informationen der vorhandenen Bildpaare, zu schätzen. Dazu stehen verschiedene lineare und nicht-lineare Verfahren zur Verfügung [1]. In diesem Kapitel wird ein lineares Verfahren vorgestellt und erläutert.

3.1 Lineare Verfahren

Die linearen Verfahren basieren auf der Epipolargleichung in Pixelkoordinaten (27). Wenn zu einem Punkt \tilde{m}_1 im ersten Bild ein korrespondierender Punkt \tilde{m}_2 im zweiten Bild existiert, muss diese Gleichung erfüllt sein. Also gilt:

$$\tilde{m}_2^T F \tilde{m}_1 = \begin{bmatrix} u_2 & v_2 & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = 0 \quad (30)$$

Daraus folgt:

$$u_1 u_2 f_{11} + u_2 v_1 f_{12} + u_2 f_{13} + u_1 v_2 f_{21} + v_1 v_2 f_{22} + v_2 f_{23} + u_1 f_{31} + v_1 f_{32} + f_{33} = 0 \quad (31)$$

und die Gleichung (30) kann in eine lineare homogene Gleichung für das korrespondierende Punktepaar i umgeschrieben werden:

$$u_i^T f = 0 \quad \text{mit} \quad u_i = [u_{i1} u_{i2}, u_{i2} v_{i1}, u_{i2}, u_{i1} v_{i2}, v_{i1} v_{i2}, v_{i2}, u_{i1}, v_{i1}, 1]^T \quad (32)$$

$$\text{und} \quad f = [f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}]^T$$

Sind n Korrespondenzen gegeben, kann man folgende lineare Gleichung aufstellen:

$$U_n f = 0 \quad \text{mit} \quad U_n = \begin{bmatrix} u_1^T \\ \vdots \\ u_n^T \end{bmatrix} \quad (33)$$

Mit Hilfe dieser linearen Gleichungen ist es möglich die Epipolar-geometrie zu schätzen [1, 5].

Die linearen Verfahren sind relativ einfach zu realisieren und es werden brauchbare Ergebnisse erzielt, wenn die verwendeten Punktkorrespondenzen eine hohe Genauigkeit aufweisen. Bei automatisierten Verfahren stellt diese Bedingung ein Problem dar, da bei der automatischen Erfassung von korrespondierenden Bildpunkten, fehlerhafte Ergebnisse nicht ausgeschlossen werden können. Bei den

nichtlinearen Verfahren nimmt man sich diesem Problem an und es wird versucht diese Fehler (Ausreißer) nicht mit in die Berechnung einfließen zu lassen. Allerdings basieren die Rechnungen der nichtlinearen Verfahren, auf Ergebnissen der linearen Methoden. Daher sind die linearen Algorithmen ein guter Einstieg zur Schätzung der Epipolargeometrie [1].

3.2 8-Punkt-Algorithmus

Wenn die Rangbedingung der Fundamental-Matrix ignoriert wird, besitzt die Matrix acht Freiheitsgrade. Dadurch können die Koeffizienten f_{ij} der Fundamental-Matrix berechnet werden, wenn mindestens acht Punktkorrespondenzen gegeben sind [5]. Je ungenauer die korrespondierenden Punkte sind, desto mehr Korrespondenzen sind nötig, um ein brauchbares Ergebnis zu erzielen. Mit den gegebenen Punktkorrespondenzen kann die Gleichung (30) durch die Methode des kleinsten quadratischen Fehlers gelöst werden.

$$\min_f \sum_i \left\| \tilde{m}_2^{iT} F \tilde{m}_1^i \right\|^2 \quad (34)$$

Oder mit Gleichung (33) umgeschrieben als:

$$\min_f \sum_i \|U_n f\|^2 \quad (35)$$

Da F nur bis auf einen Skalierungsfaktor definiert ist, gilt dies auch für den Vektor f . Die einfachste Lösung wäre $f = 0$, jedoch würde dieser Fall zu keinem verwendbaren Ergebnis führen. Um diesen Fall auszuschließen, müssen weitere Bedingungen aufgestellt werden [5]. Zwei dieser Verfahren werden in den nächsten Abschnitten vorgestellt.

3.2.1 Lineare Methode des kleinsten quadratischen Fehlers

Bei dieser Methode wird ein Koeffizient der Fundamental-Matrix auf Eins gesetzt und der Ansatz aus Gleichung (35) wird umgeschrieben:

$$\|U_n f\|^2 = \|U'_n f' - c_9\|^2 = f'^T U_n'^T U'_n f' - 2c_9^T U'_n f' + c_9^T c_9 \quad (36)$$

Wobei die Matrix U'_n aus den ersten acht Spalten von U_n gebildet wird und c_9 die neunte Spalte von U_n ist. Setzt man die Ableitung gleich Null, erhält man die folgende Lösung [5]:

$$\frac{\partial \|U_n f\|^2}{\partial f'} = 0 \quad (37)$$

Somit ist

$$2U_n'^T U'_n f' - 2U_n'^T c_9 = 0 \quad \text{und daher} \quad f' = (U_n'^T U'_n)^{-1} U_n'^T c_9 \quad (38)$$

Wird ein Koeffizient auf Eins gesetzt, dessen Wert eigentlich bei Null liegt, oder der deutlich kleiner ist als die anderen Koeffizienten, erhält man ein fehlerhaftes Ergebnis. Eine Lösung für dieses Problem, ist die Berechnung aller neun Möglichkeiten. Aus diesen Ergebnissen wird dann die beste Schätzung ermittelt [1,5].

3.2.2 Auswertung der Eigenwerte

Bei diesem Verfahren wird die Bedingung aufgestellt, dass die Norm des Vektors f den Wert Eins annimmt. Dieses Minimierungsproblem kann mit der Lagrangeschen⁵ Multiplikationsregel umgeschrieben werden als:

$$\min_f L(f, \lambda) \quad \text{mit } L(f, \lambda) = \|U_n f\|^2 + \lambda(1 - \|f\|^2) \quad (39)$$

Setzt man die erste Ableitung dieser Funktion gleich Null, erhält man die Gleichung:

$$U_n^T U_n f = \lambda f \quad (40)$$

⁵ Um eine Funktion mit Nebenbedingungen zu minimieren, wird oftmals die Lagrangesche Multiplikationsregel benutzt [10].

Diese Gleichung repräsentiert das klassische Eigenwertproblem, und die Lösung für f ist ein Eigenvektor der Matrix $U_n^T U_n$, mit dem dazugehörigen Eigenwert λ . Somit gibt es bis zu neun mögliche Lösungen. Da wir die Funktion (40) minimieren wollen, ist die Lösung für diese Gleichung, der Eigenvektor zum kleinsten Eigenwert [1, 5]. Im Gegensatz zum ersten Verfahren, werden hier alle Koeffizienten der Fundamental-Matrix in gleicher Weise berücksichtigt. Die Rangbedingung wird allerdings in beiden Ansätzen ignoriert [1].

Für die Berechnungen der Fundamental-Matrix im Verlauf dieser Arbeit, wird auf die zweite Methode zurückgegriffen.

4 Implementierung

Für die praktische Berechnung der Fundamental-Matrix zu einem gegebenen Stereosystem, wird der 8-Punkt-Algorithmus in der Programmiersprache Java implementiert. Zur Überprüfung der Richtigkeit, werden mittels der berechneten Fundamental-Matrix, die Epipolarlinien für ein Bildpaar bestimmt und mit einer Bildverarbeitungssoftware visualisiert. Außerdem wird die Essential-Matrix bestimmt und deren Parameter überprüft. In den nächsten Abschnitten wird die, für diese Arbeit verwendete Software, beschrieben. Darüber hinaus wird der Aufbau des Java-Programms und dessen Verbindung mit dem ImageJ-Plugin erläutert.

4.1 Software

Der Java-Programmcode wurde mit Hilfe der Open-Source-Entwicklungsplattform Eclipse erstellt. Dieses Programm bietet eine Benutzeroberfläche zur Entwicklung von Softwareprojekten in verschiedenen Programmiersprachen und steht kostenlos auf der Eclipse-Webseite⁶ zur Verfügung. Für die Nutzung von Eclipse ist die Installation eines *Java Runtime Environment* notwendig. Dieses kann separat oder als Teil des *Java Development Kit* von *Sun Microsystems*⁷ bezogen

⁶ <http://www.eclipse.org> (Stand: 19.01.2009)

⁷ <http://java.sun.com> (Stand: 19.01.2009)

werden. Für die mathematische Verarbeitung von Matrizen und Vektoren, wurde die Java-Bibliothek JAMA zu dem Eclipse-Projekt hinzugefügt. Mit diesem Paket ist es möglich, alle für diese Arbeit notwendigen Matrixoperationen in Java durchzuführen. Über die Seite des *National Institute of Standards and Technology*⁸ gelangt man, neben der aktuellen Version von JAMA, auch an eine hilfreiche Dokumentation. Zur Visualisierung der Ergebnisse in den Bildpaaren, wird die auf Java basierende und gemeinfreie Bildverarbeitungssoftware ImageJ verwendet. Auf der Internetseite von ImageJ⁹ stehen die neuste Version der Software, sowie eine Dokumentation und Plugins zum Download bereit.

4.2 Aufbau des Programms

Aufgabe des Programms ist es, mit Hilfe des vorgestellten 8-Punkt-Algorithmus, Parameter der Epipolargeometrie eines Stereosystems zu berechnen. Grundlage für diese Berechnung sind Bildpaare, die aus unterschiedlichen Perspektiven aufgenommen wurden. Aus einem Bildpaar müssen manuell Punktkorrespondenzen in Pixelkoordinaten bestimmt werden. Mit den Informationen dieser Korrespondenzen werden die Fundamental-Matrix und die sich daraus ergebenden Merkmale des Stereosystems ermittelt.

⁸ <http://math.nist.gov/javanumerics/jama> (Stand: 19.01.2009)

⁹ <http://rsb.info.nih.gov/ij> (Stand 19.01.2009)

Hierfür ist das Java-Projekt *Epipolargeometrie* in fünf Klassen unterteilt:

- *KorrespondierendePunkte.java*
- *EpipolargeometriePixel.java*
- *Fehlerrechnung.java*
- *EpipolargeometrieKamera.java*
- *Ausgabe.java*

Die Aufgaben der Klassen werden in den nächsten Kapiteln erläutert. Der Programmcode steht auf der beiliegenden CD-Rom zur Verfügung und wird daher nur in Ausschnitten gezeigt.

4.2.1 Die Klasse *KorrespondierendePunkte*

In dieser Klasse werden den Pixelkoordinaten der Punktkorrespondenzen fortlaufende Variablen zugewiesen. Wie jetzt bekannt ist, sind mindestens acht Punktkorrespondenzen für den 8-Punkt-Algorithmus, mit der Bedingung aus Kapitel 3.2.2, notwendig. Die ersten Berechnungen haben jedoch gezeigt, dass es für ein akzeptables Ergebnis sinnvoll ist, mindestens zehn korrespondierende Punkte auszuwählen (s. Kapitel 5.1). Zu diesen zehn Punkten werden noch weitere sechs Korrespondenzen bestimmt. (Diese Punkte dienen zur Bestimmung von Epipolarlinien, zu Punkten, die nicht zur Berechnung der Fundamental-Matrix verwendet wurden). Somit werden insgesamt

16 Punktkorrespondenzen bestimmt. Jeder Punkt besteht aus zwei Variablen u und v , für die horizontale und vertikale Koordinatenbeschreibung. Diese werden jeweils mit einem Index für die Nummer der Korrespondenz und des Bildes gekennzeichnet. Beispielsweise wird die dritte Korrespondenz folgendermaßen deklariert (Links die Koordinaten im ersten Bild und rechts im zweiten).

```
int u31 = 148;    int u32 = 178;  
int v31 = 329;    int v32 = 321;
```

Für die weiteren Berechnungen werden die integer Variablen eines Punktes, in ein Matrixobjekt der JAMA-Bibliothek geschrieben. Für die Nutzung der Funktionen dieser Bibliothek, muss sie am Anfang der Klasse importiert werden.

```
import Jama.*;
```

Dann werden die Punkte zuerst in einen Array gefasst und danach in ein Matrixobjekt umgeschrieben. Für den Punkt im ersten Bild, der dritten Korrespondenz, würde dies folgendermaßen aussehen:

```
double[][] m31 = { {u31},  
                  {v31},  
                  {1} };  
  
Matrix M31 = new Matrix(m31);
```

Zusätzlich werden in dieser Klasse die Punktkorrespondenzen, mit Hilfe von bekannten Kameraparametern, in normierte Sensorkoordinaten

umgerechnet. Da der Ursprung des Bildkoordinatensystems links oben liegt, müssen die Punkte zuerst in das Koordinatensystem des Sensors verschoben werden. Dazu zieht man die Hälfte der gesamten Pixelbreite von der u -Koordinate, und die Hälfte der gesamten Pixelhöhe von der v -Koordinate ab. Mit Hilfe der Sensorgröße kann die Breite und Höhe eines Pixels in Millimetern angegeben werden. Somit wird die Pixelanzahl mit einer metrischen Einheit, in Sensorkoordinaten wiedergegeben. Teilt man diesen Wert durch die Brennweite, mit welcher die Bilder aufgenommen wurden, erhält man die normierten Sensorkoordinaten der Punkte.

Die Klasse *KorrespondierendePunkte* ist eine Elternklasse der Klassen *EpipolargeometriePixel*, *Fehlerrechnung* und *EpipolargeometrieKamera*. Somit können diese Klassen auf die eingegebenen Punktkorrespondenzen zugreifen.

4.2.2 Die Klasse EpipolargeometriePixel

In dieser Klasse wird, mittels der korrespondierenden Punkte, die Fundamental-Matrix berechnet. Dazu wird ein Matrixobjekt U , nach dem im Kapitel 3.1 vorgestellten Verfahren, erstellt. Der nächste Schritt ist die Berechnung der Matrix aus dem Produkt $U^T U$. Dazu wird mit einer JAMA-Funktion die Transponierte der Matrix U gebildet.

```
Matrix Utrans = U.transpose();
```

Ein Produkt aus zwei Matrizen lässt sich ebenfalls mit einer JAMA-Funktion berechnen.

```
Matrix prod = Utrans.times(U);
```

Nach Gleichung (40) müssen zu dieser 9x9 Matrix die Eigenwerte bestimmt werden. Hierfür wird die Matrix in ein Objekt der JAMA-Klasse *EigenvalueDecomposition* übergeben.

```
EigenvalueDecomposition EVD =  
    new EigenvalueDecomposition(prod);
```

Mit diesem Objekt wird die Funktion `getRealEigenvalues()` aufgerufen, mit welcher die Eigenwerte in ein eindimensionales Array geschrieben werden können. Dabei werden die Eigenwerte der Größe nach sortiert, angefangen beim Kleinsten. Des Weiteren ist es mit der Funktion `getV()` möglich, die Eigenvektoren spaltenweise in ein Matrixobjekt zu schreiben. Dabei gibt die erste Spalte den Eigenvektor zum kleinsten Eigenwert an. Somit erhält man die neun Koeffizienten der Fundamental-Matrix. Für weitere Berechnungen werden diese in eine 3x3 Matrix übergeben.

In dieser Klasse wird die Fundamental-Matrix zur Bestimmung der Epipolarlinien verwendet. Diese berechnen sich nach Gleichung (29) und die Werte werden auf der Konsole ausgegeben. Diese Werte

werden dann in das ImageJ-Plugin eingefügt, um die Epipolarlinien in beide Bilder des Stereosystems zu zeichnen (Näheres dazu in Kapitel 4.3). Da die Koordinaten der Punkte und deren Korrespondenzen bekannt sind, kann die Abweichung der Linien berechnet werden. Dazu werden Funktionen der Klasse *Fehlerrechnung* aufgerufen, die im nächsten Kapitel beschrieben werden.

4.2.3 Die Klasse Fehlerrechnung

Durch die Schätzung der Fundamental-Matrix muss mit Ungenauigkeiten der Ergebnisse gerechnet werden. Daher ist es interessant, die Größe eines eventuellen Fehlers der Epipolarlinie zu kennen. Dazu werden in der Klasse *Fehlerrechnung* Funktionen bereitgestellt, die den Abstand einer Epipolarlinie zu dem Punkt berechnen, den sie schneiden soll. Die berechneten Werte der Linien werden bei Ausführung der Funktionen übergeben. Die zugehörigen Punkte sind bekannt und können von der Elternklasse *KorrespondierendePunkte* bezogen werden. Somit ist eine Gerade l und ein Punkt m gegeben und der Fehler d wird mit folgender Formel berechnet [5]:

$$d = \frac{au + bv + c}{\sqrt{a^2 + b^2}} \quad \text{mit } l = [a, b, c]^T \text{ und } m = [u, v]^T \quad (41)$$

Die Berechnung des Abstandes wird für beide Bilder separat durchgeführt und in drei unterschiedliche Kategorien unterteilt:

- Berechnung des Abstandes zwischen Epipolarlinien und Punkten, die zur Schätzung der Fundamental-Matrix verwendet wurden.
- Berechnung des Abstandes zwischen Epipolarlinien und Punkten, die nicht zur Schätzung der Fundamental-Matrix verwendet wurden.
- Aus den insgesamt 16 Epipolarlinien pro Bild, werden acht zufällig ausgewählt und der Abstand zu den jeweiligen Punkten berechnet.

Bei den ersten beiden Varianten, werden die zehn bzw. sechs Ergebnisse der Epipolarlinien übergeben und der Fehler wird nach Gleichung (41) berechnet. Für die Berechnung der Abstände, zu acht zufällig ausgewählten Epipolarlinien, werden alle 16 Werte übergeben. Mit der Methode `Math.random()` wird eine Zufallszahl zwischen eins und 16 erzeugt. Außerdem wird jeder Epipolarlinie eine Variable vom Typ `boolean` zugewiesen. Ob der Fehler zu einer bestimmten Linie berechnet werden soll, hängt von zwei Bedingungen ab, die in einer `if`-Abfrage überprüft werden. Entspricht die Zufallszahl der Nummer der Epipolarlinie und steht der Wert der booleschen Variable auf `true`, wird die Fehlerrechnung durchgeführt. Mit Ausführung der Rechnung wird der boolesche Wert von `true` auf `false` gesetzt. Somit wird verhindert, dass der Fehler zu einer Epipolarlinie mehrfach berechnet

wird. Diese Prozedur wird mit einer Schleife so lange ausgeführt, bis acht Fehler berechnet wurden.

Bei allen drei Rechnungen wird außerdem der maximale und minimale, sowie der mittlere Fehler bestimmt.

4.2.4 Die Klasse *EpipolargeometrieKamera*

Wird der 8-Punkt-Algorithmus mit Sensorkoordinaten, anstatt mit Pixelkoordinaten ausgeführt, erhält man als Ergebnis die Essential-Matrix. Diese Berechnung wird in der Klasse *EpipolargeometrieKamera* realisiert. Für Informationen der externen Kameraparameter, wird die Matrix mit der Singulärwertzerlegung¹⁰ in zwei orthogonale Matrizen und eine diagonale Matrix aufgeteilt.

$$E = USV^T \quad (42)$$

Dafür stehen die Methoden `getU()`, `getS()` und `getV()` in der JAMA-Bibliothek zur Verfügung. Diese können über ein Objekt der Klasse `SingularValueDecomposition` aufgerufen werden. Diesem Objekt wird die zu berechnende Matrix übergeben. Die Matrizen U und

¹⁰ Mit der Singulärwertzerlegung (engl. singular value decomposition, SVD) kann eine beliebige Matrix A , als ein Produkt von drei Matrizen faktorisiert werden. Wobei $A = USV^T$, mit U und V = orthogonale Matrizen und S = diagonale Matrix [11].

V sind Rotationsmatrizen, die jeweils die optische Achse der Kamerakoordinatensysteme auf den Verschiebungsvektor setzen. Außerdem werden die Koordinatensysteme 90 Grad um diese Achse zueinander gedreht. Die diagonale Matrix S enthält dann die relativen Informationen des Verschiebungsvektors t mit $S = \text{diag}\{\|t\|, \|t\|, 0\}$ [6].

4.2.5 Die Klasse Ausgabe

Die Rechnungen der Klassen *EpipolargeometriePixel* und *EpipolargeometrieKamera* sind in Methoden verpackt, die in der Klasse *Ausgabe* ausgeführt werden. Diese Klasse enthält die main-Methode¹¹ und dient zur Ausführung des Programms. Mit der Ausführung der Methoden werden die Berechnungen durchgeführt und die Ergebnisse werden gegebenenfalls ausgegeben.

4.3 Visualisierung der Epipolarlinien

Da die Ergebnisse der Epipolarlinien in Pixelwerten vorliegen, können diese auch in einem digitalen Bild veranschaulicht werden. Dies geschieht mit Hilfe eines Plugins für ImageJ. Plugins für diese Software

¹¹ Durch die main-Methode kennt der Java-Interpreter den Anfang des Programms. Ohne diese Methode kann das Programm nicht ausgeführt werden [12].

sind Java-Klassen, die vorgegebene Methoden enthalten müssen. Für die Übergabe und Verarbeitung eines Bildes sind die Methoden `setup()` und `run()` notwendig. Die erste Methode überprüft, ob die Werte des übergebenen Bildes vom Plugin verarbeitet werden können. Die zweite Methode implementiert ein Objekt vom Typ *ImageProcessor*, welches die Informationen des Bildes enthält. Somit können alle Pixel über zwei verschachtelte `for`-Schleifen ausgelesen und verändert werden [3]. Die Java-Klassen müssen sich im `plugins`-Ordner von `ImageJ` befinden. Die Ausführung erfolgt über die grafische Benutzeroberfläche. Nachdem ein Bild zur Verarbeitung ausgewählt wurde, kann ein Plugin über das Menü *Plugins > Compile and Run* ausgewählt und angewendet werden.

Für die Verarbeitung der Epipolarlinien wurde das Plugin *Epipolarlinien_.java* erstellt, welches für 8-Bit-Grauwertbilder ausgelegt ist. Dafür müssen die berechneten Werte der Epipolarlinien $l_i = [a_i, b_i, c_i]^T$ eines Bildes, in dieser Klasse vorhanden sein. Die Epipolarlinie wird mit der folgenden Gleichung berechnet:

$$y = -\frac{a}{b}x - \frac{c}{b} \quad (43)$$

Dabei sind x und y die Koordinaten des Pixels. Ist diese Gleichung erfüllt, wird der entsprechende Pixel auf den Wert 255 gesetzt. Somit werden die Epipolarlinien, als weiße Geraden, in den Bildern visualisiert (siehe Kapitel 5.1).

5 Ergebnisse

Die vorgestellten Berechnungen wurden an verschiedenen Stereobildpaaren durchgeführt. Einige der resultierenden Ergebnisse und deren Genauigkeit, werden in diesem Kapitel zusammengefasst.

5.1 Visualisierung der Epipolarlinien

In der Theorie benötigt der 8-Punkt-Algorithmus mindestens acht korrespondierende Punkte für die Berechnung der Fundamental-Matrix. Allerdings wird dabei eine hohe Genauigkeit der Werte vorausgesetzt. Bei einer manuellen Bestimmung der Punkte, hat es sich als sinnvoll herausgestellt, mehr als acht Korrespondenzen zu wählen. Die Werte der Fundamental-Matrix geben keine Auskunft über die Genauigkeit. Allerdings kann man die Genauigkeit der Epipolarlinien, welche mit der Fundamental-Matrix berechnet wurden, bestimmen.

Die folgenden Stereobildpaare zeigen die unterschiedlichen Ergebnisse, bei einer unterschiedlichen Anzahl von verwendeten Punkt-korrespondenzen. Dabei wurde die Fundamental-Matrix mit acht, zehn und zwölf Punktepaaren berechnet. Mit diesen Matrizen wurden die Epipolarlinien berechnet und in die Bilder eingezeichnet. Für eine bessere Übersicht, werden nur acht Linien in jedem Bild gezeigt.



Abbildung 5: Stereobildpaar einer Puppe [1].

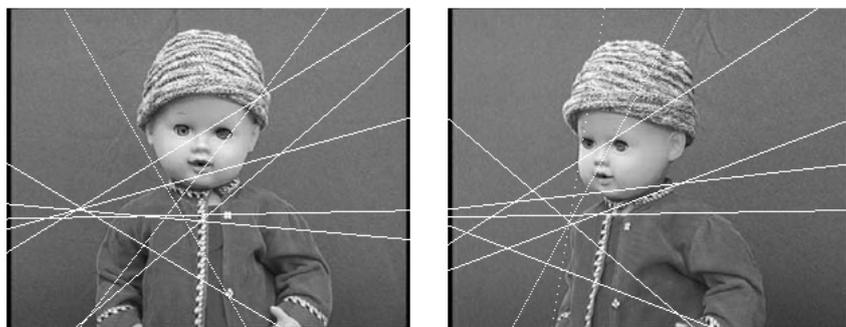


Abbildung 6: Epipolarlinien zum Stereobildpaar der Puppe. Die Fundamental-Matrix wurde aus acht Korrespondenzen berechnet.

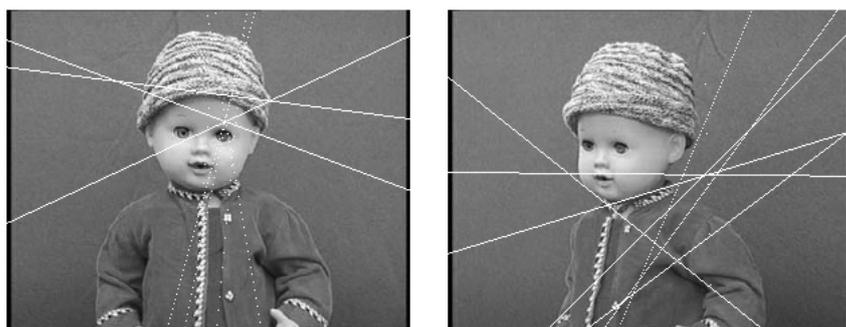


Abbildung 7: Epipolarlinien zum Stereobildpaar der Puppe. Die Fundamental-Matrix wurde aus zehn Korrespondenzen berechnet.



Abbildung 8: Epipolarlinien zum Stereobildpaar der Puppe. Die Fundamental-Matrix wurde aus zwölf Korrespondenzen berechnet.



Abbildung 9: Stereobildpaar eines Schreibtisches [13].



Abbildung 10: Epipolarlinien zum Stereobildpaar des Schreibtisches. Die Fundamental-Matrix wurde aus acht Korrespondenzen berechnet.

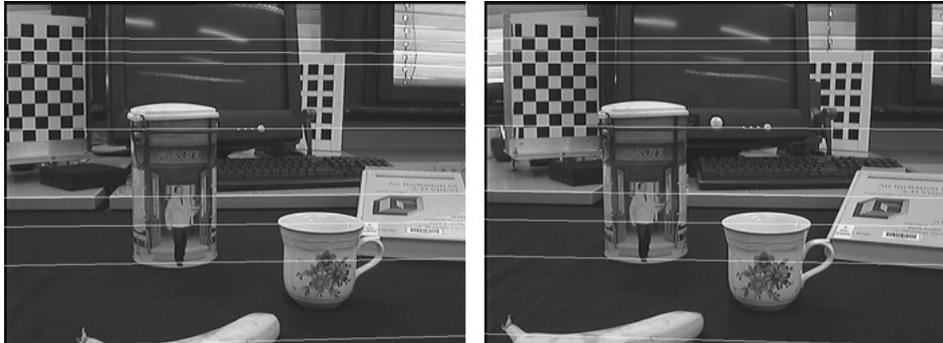


Abbildung 11: Epipolarlinien zum Stereobildpaar des Schreibtisches. Die Fundamental-Matrix wurde aus zehn Korrespondenzen berechnet.

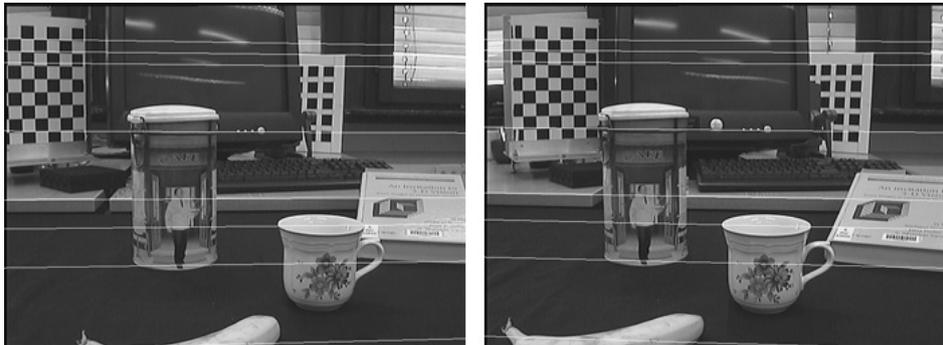


Abbildung 12: Epipolarlinien zum Stereobildpaar des Schreibtisches. Die Fundamental-Matrix wurde aus zwölf Korrespondenzen berechnet.

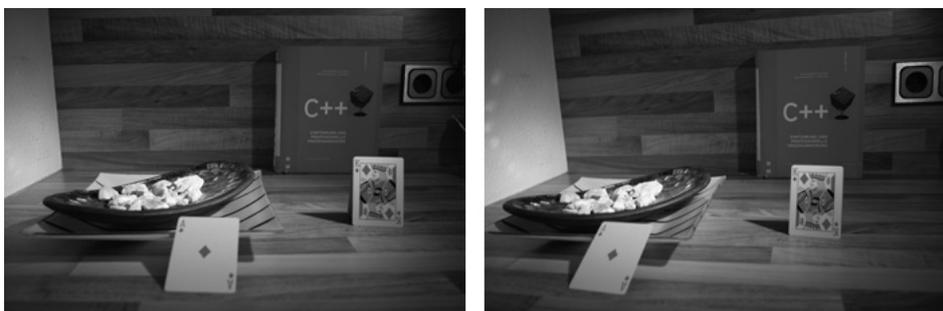


Abbildung 13: Selbst erstelltes Stereobildpaar.

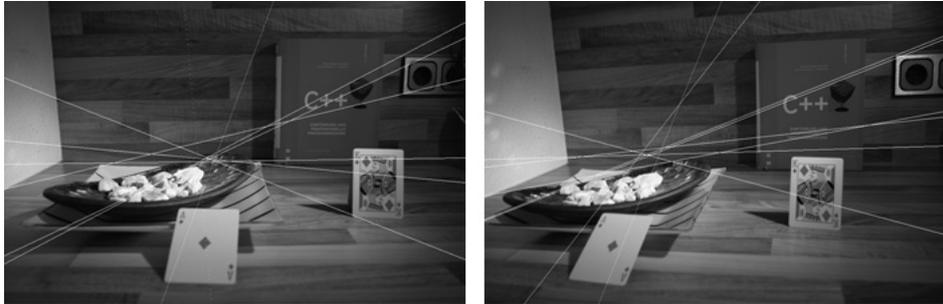


Abbildung 14: Epipolarlinien zum selbst erstellten Stereobildpaar. Die Fundamental-Matrix wurde aus acht Korrespondenzen berechnet.



Abbildung 15: Epipolarlinien zum selbst erstellten Stereobildpaar. Die Fundamental-Matrix wurde aus zehn Korrespondenzen berechnet.



Abbildung 16: Epipolarlinien zum selbst erstellten Stereobildpaar. Die Fundamental-Matrix wurde aus zwölf Korrespondenzen berechnet.

Die Abbildungen verdeutlichen, dass keine allgemeingültigen Aussagen darüber abgeleitet werden können, wie viele korrespondierende Punkte benötigt werden, um die Fundamental-Matrix zu einem Stereosystem ausreichend genau zu schätzen. Für das zweite Stereobildpaar sind acht Korrespondenzen ausreichend, wobei das erste Bildpaar zwölf benötigt. Die Genauigkeit der ausgewählten Punkte ist dabei ausschlaggebend. Für vergleichbare Ergebnisse, wird die Fundamental-Matrix für weitere Berechnungen aus zehn korrespondierenden Punkten berechnet.

5.2 Abweichung der Epipolarlinien

Wie in Kapitel 4.2.3 beschrieben, kann die Abweichung der Epipolarlinien in absoluten Pixelwerten angegeben werden. Die Fehler der Stereobildpaare aus Abbildung 9 und Abbildung 13 werden in diesem Kapitel aufgeführt. Abbildung 17 zeigt den Unterschied zwischen Punkten, die zur Berechnung der Fundamental-Matrix genutzt wurden und beliebigen Punkten. Außerdem wird der Unterschied zwischen beiden Bildpaaren deutlich.

	minimaler Fehler	maximaler Fehler	mittlerer Fehler
Abb.7 Bild 1	0	0	0
Abb.7 Bild 2	0	0	0
Abb.9 Bild 1	0	6	1,9
Abb.9 Bild 2	0	6	1,9

	minimaler Fehler	maximaler Fehler	mittlerer Fehler
Abb.7 Bild 1	0	5	1,5
Abb.7 Bild 2	0	5	1,5
Abb.9 Bild 1	0	5	2,66
Abb.9 Bild 2	0	6	3

Abbildung 17: Fehler der Epipolarlinien aus Abb. 7 und Abb. 9 (oben: Abstand zu Punkten die zur Berechnung der Fundamental-Matrix genutzt wurden; unten: Abstand zu beliebigen Punkten)

Es ist zu erkennen, dass die Fehler zu Punkten, die zur Berechnung der Fundamental-Matrix verwendet wurden, kleiner ausfallen. Abbildung 18 zeigt den mittleren Abstand aller Epipolarlinien. Dafür wurde zehn Mal der mittlere Abstand von acht zufällig gewählten Linien berechnet und davon der Mittelwert gebildet.

	Mittelwert des mittleren Fehlers
Abb.7 Bild 1	0,4375
Abb.7 Bild 2	0,6875
Abb.9 Bild 1	2,1375
Abb.9 Bild 2	2,3375

Abbildung 18: Durchschnittlicher Abstand der Epipolarlinien zu den zugehörigen korrespondierenden Punkten

5.3 Beurteilung der Essential-Matrix

Eine weitere Überprüfung der Ergebnisse erfolgt über die Singulärwertzerlegung der Essential-Matrix, welche in Kapitel 4.2.4 vorgestellt wurde. Somit sollten die ersten beiden Werte der Diagonalmatrix ungefähr gleich groß sein. Der dritte Wert müsste in der Nähe von Null liegen. Abbildung 19 zeigt die Diagonalmatrix zu dem Stereobildpaar aus Abbildung 9.

$$\begin{bmatrix} 0,7113807916732091 & 0 & 0 \\ 0 & 0,7028067017219914 & 0 \\ 0 & 0 & 3,3053449888739657 \cdot 10^{-4} \end{bmatrix}$$

Abbildung 19: Mit der Singulärwertzerlegung berechnete Diagonalmatrix zum Stereobildpaar aus Abbildung 7

Dabei waren folgende Parameter gegeben, die zur Umrechnung von Pixelkoordinaten zu Sensorkoordinaten verwendet wurden [13].

- Sensorbreite: 6,4 mm
- Sensorhöhe: 4,8 mm
- Bildbreite: 768 pixel
- Bildhöhe: 576 pixel
- Brennweite: 8,673491 mm

An den Werten der Diagonalmatrix ist zu erkennen, dass die Essential-Matrix nicht hundertprozentig genau berechnet wurde. Die ersten zwei Werte liegen jedoch sehr nah zusammen und der dritte Wert geht gegen Null. Somit ist auf eine gute Näherung der Essential- und damit auch der Fundamental-Matrix zu schließen.

6 Diskussion

Die Ergebnisse aus Kapitel 5 zeigen, dass es möglich ist, die Epipolargeometrie einer Szene, lediglich anhand von Bildpaaren annähernd zu bestimmen. Der dafür verwendete lineare 8-Punkt-Algorithmus setzt allerdings eine genaue Bestimmung von Punktkorrespondenzen voraus. Jedoch kann sowohl bei einer manuellen, sowie automatischen Auswahl der Punkte, keine fehlerfreie Bestimmung angenommen werden. Für eine genauere Bestimmung könnten die korrespondierenden Punkte, beispielsweise über ihren Mittelwert normiert werden. Weiterhin sind durch die Schätzung der Fundamental-Matrix bestimmte Bedingungen nicht erfüllt, die durch Erweiterungen des Verfahrens erzwungen werden können. Dabei könnte der Algorithmus durch ein Verfahren erweitert werden, welches die Rangbedingung der Fundamental-Matrix erzwingt.

Der Vorteil des 8-Punkt-Algorithmus, gegenüber nichtlinearen Verfahren, liegt in der Realisierung. So sind mit einer geringen Anzahl an Punktkorrespondenzen, die ohne weitere Verfahren manuell bestimmt werden können, brauchbare Ergebnisse erzielbar. Auch die Umsetzung des Algorithmus, mit einer Programmiersprache, ist einfacher zu realisieren, als bei nichtlinearen Algorithmen. Weiterhin basieren die meisten nichtlinearen Verfahren auf Ergebnissen, die durch lineare Verfahren erzielt werden. Dem höheren Aufwand der nichtlinearen Verfahren, stehen allerdings genauere Ergebnisse gegenüber. Darüber

hinaus sind lineare Verfahren nicht für eine automatische Erfassung von Punktkorrespondenzen geeignet.

Durch die Bestimmung der Epipolargeometrie einer Szene lassen sich weitere Korrespondenzen des Bildpaares leichter ausfindig machen. Somit können auch korrespondierende Bildmerkmale in Aufnahmen bestimmt werden. Darüber hinaus, kann mit Hilfe der Punktkorrespondenzen, die Tiefeninformation der dreidimensionalen Szene berechnet werden, welche durch die zweidimensionale Abbildung verloren gegangen ist. Diese Informationen können zum Beispiel für eine dreidimensionale Modellierung einer natürlichen Szene, oder zum Navigieren von Roboter gesteuerten Systemen, verwendet werden.

Literaturverzeichnis

- [1] O. Schreer: *Stereoanalyse und Bildsynthese*; Springer-Verlag Berlin Heidelberg; 1. Auflage; 2005.
- [2] J. Vogelmann: *Darstellende Geometrie (Kamprath-Reihe: Technik)*; Vogel-Buchverlag Würzburg; 3. Auflage; 1989.
- [3] W. Burger, M. Burge: *Digitale Bildverarbeitung: Eine Einführung mit Java und ImageJ*; Springer-Verlag Berlin Heidelberg; 1. Auflage; 2005.
- [4] M. Pries: *Darstellende Geometrie*; Technische Fachhochschule Berlin; 2008; Internetquelle: <http://www1.tfh-berlin.de/~pries/Vorlesungen/DG/EinleitungUndGrundlagen.pdf> (Stand: 19.01.2009).
- [5] Z. Zhang: *Determining the Epipolar Geometry and its Uncertainty: A Review*; Int. Journal of Computer Vision; Vol.27(2), pp.161-195; 1998; Internetquelle: <http://research.microsoft.com/en-us/um/people/zhang/Papers/IJCV-Review.pdf> (Stand: 19.01.2009).
- [6] Y. Ma, S. Soatto, J. Kosecka, S. S. Sastry: *An Invitation to 3-D Vision: From Images to Geometric Models*; Springer-Verlag; 1. Auflage; 2004.

- [7] H. Handels: *Medizinische Bildverarbeitung*; B.G. Teubner Stuttgart Leipzig; 1. Auflage; 2000.
- [8] VDI Berichte 1572: *Bildverarbeitung im industriellen Einsatz – Image Processing and Machine Vision*; VDI Verlag GmbH Düsseldorf; 2000.
- [9] J. Steinmüller: *Bildanalyse – Von der Bildverarbeitung zur räumlichen Interpretation von Bildern*; Springer-Verlag Berlin Heidelberg; 1. Auflage; 2008.
- [10] P. Grohs: *Lagrange Multiplikatoren*; Technische Universität Wien; 2006; Internetquelle: <http://www.dmg.tuwien.ac.at/grohs/lva/lagrange.pdf> (Stand: 19.01.2009).
- [11] G. Strang: *Lineare Algebra*; Springer-Verlag Berlin Heidelberg; 1. Auflage; 2003.
- [12] C. Ullenboom: *Java ist auch eine Insel*; Galileo Press; 7. Auflage; 2007.
- [13] S. Jockel: *3–dimensionale Rekonstruktion einer Tischszene aus monokularen Handkamera–Bildsequenzen im Kontext autonomer Serviceroboter*; Diplomarbeit Universität Hamburg; 2006; Internetquelle: <http://tams-www.informatik.uni-hamburg.de/publications/2006/Jockel06Diplomarbeit.pdf> (Stand: 19.01.2009).

Abbildungsverzeichnis

Abbildung 1:	Konstruktion der Perspektive. Der Punkt O stellt den Betrachter dar [1]. Die parallelen Linien müssen nicht notwendigerweise senkrecht auf die Bildebene fallen... 3
Abbildung 2:	Schematischer Aufbau des Lochkammermodells [1]..... 7
Abbildung 3:	Externe Transformation [1]..... 8
Abbildung 4:	Aufbau der Epipolargeometrie [1]..... 13
Abbildung 5:	Stereobildpaar einer Puppe [1]..... 36
Abbildung 6:	Epipolarlinien zum Stereobildpaar der Puppe. Die Fundamental-Matrix wurde aus acht Korrespondenzen berechnet. 36
Abbildung 7:	Epipolarlinien zum Stereobildpaar der Puppe. Die Fundamental-Matrix wurde aus zehn Korrespondenzen berechnet. 36
Abbildung 8:	Epipolarlinien zum Stereobildpaar der Puppe. Die Fundamental-Matrix wurde aus zwölf Korrespondenzen berechnet. 37
Abbildung 9:	Stereobildpaar eines Schreibtisches [13]. 37
Abbildung 10:	Epipolarlinien zum Stereobildpaar des Schreibtisches. Die Fundamental-Matrix wurde aus acht Korrespondenzen berechnet. 37
Abbildung 11:	Epipolarlinien zum Stereobildpaar des Schreibtisches. Die Fundamental-Matrix wurde aus zehn Korrespondenzen berechnet. 38

Abbildung 12: Epipolarlinien zum Stereobildpaar des Schreibtisches. Die Fundamental-Matrix wurde aus zwölf Korrespondenzen berechnet.	38
Abbildung 13: Selbst erstelltes Stereobildpaar.....	38
Abbildung 14: Epipolarlinien zum selbst erstellten Stereobildpaar. Die Fundamental-Matrix wurde aus acht Korrespondenzen berechnet.	39
Abbildung 15: Epipolarlinien zum selbst erstellten Stereobildpaar. Die Fundamental-Matrix wurde aus zehn Korrespondenzen berechnet.	39
Abbildung 16: Epipolarlinien zum selbst erstellten Stereobildpaar. Die Fundamental-Matrix wurde aus zwölf Korrespondenzen berechnet.	39
Abbildung 17: Fehler der Epipolarlinien aus Abb. 7 und Abb. 9 (oben: Abstand zu Punkten die zur Berechnung der Fundamental-Matrix genutzt wurden; unten: Abstand zu beliebigen Punkten).....	41
Abbildung 18: Durchschnittlicher Abstand der Epipolarlinien zu den zugehörigen korrespondierenden Punkten.....	41
Abbildung 19: Mit der Singulärwertzerlegung berechnete Diagonalmatrix zum Stereobildpaar aus Abbildung 7..	42

Eidesstattliche Erklärung

Ich versichere hiermit, die vorgelegte Arbeit in dem gemeldeten Zeitraum ohne fremde Hilfe verfasst und mich keiner anderen als der angegebenen Hilfsmittel und Quellen bedient zu haben.

Köln, den 20.01.2009

Unterschrift
(Patrick Stein)

Sperrvermerk

Die vorgelegte Arbeit unterliegt keinem Sperrvermerk.

Weitergabeerklärung

Ich erkläre hiermit mein Einverständnis, dass das vorliegende Exemplar meiner Bachelorarbeit oder eine Kopie hiervon für wissenschaftliche Zwecke verwendet werden darf.

Köln, den 20.01.2009

Unterschrift
(Patrick Stein)