

Praktikumsversuch

Kamerakalibrierung

S. Rahmann und H. Burkhardt

Inhaltsverzeichnis

1	Kamerakalibrierung	5
1.1	Einleitung	5
1.2	Projektive Geometrie	6
1.2.1	Einleitung	6
1.2.2	Die projektive Ebene	8
1.2.3	Transformationen und ihre Hierarchie	9
1.2.4	Berechnung einer Homographie	12
1.3	Kameramodell	13
1.3.1	Die Lochkamera	13
1.3.2	Linseverzerrungen	15
1.3.3	Projektion einer Ebene	17
1.4	Kalibrierung	18
1.4.1	Einleitung und Überblick	18
1.4.2	Kalibriermuster	18
1.4.3	3D Kalibrierung	19
1.4.4	2D Kalibrierung	22
1.4.5	Selbstkalibrierung	25
1.5	Numerisches Handwerkszeug	27
1.5.1	Fehlerquadrat Methode	27
1.5.2	Normalisierung	28
1.6	Literatur	29
1.7	Aufgaben	29
1.7.1	Theoretischer Teil	29
1.7.2	Praktischer Teil	31

1 Kamerakalibrierung

1.1 Einleitung

Eine Kamera bildet die dreidimensionale Welt auf ein zweidimensionales Bild ab; 3D Welt- oder Szenenpunkte werden auf 2D Bildpunkte projiziert. Dieser Praktikumsversuch beschäftigt sich mit der Modellierung von realen Kameras und geeigneten Methoden zur Bestimmung der Modellgrößen.

Ein zentraler Bereich des Computer Sehens ist die dreidimensionale Interpretation von Bildern. Basierend auf den Bilddaten können Objekte vermessen oder ihre Lage im Raum bestimmt werden. Auch vollständig unbekannte Szene können rekonstruiert werden. Für all diese Aufgaben ist es notwendig, daß in irgendeiner Form die Parameter des Kameramodells bestimmt werden. Die Bestimmung der Kameraparameter nennt man Kamerakalibrierung. Kamerakalibrierung ist ein wichtiges Thema, da die Genauigkeit der Rekonstruktion natürlich von der Genauigkeit der Kalibrierung abhängt. Grundsätzlich gibt es dazu zwei Möglichkeiten: Eine Kalibrierung auf Grund von Bildern eines speziellen Kalibrierungsmusters und die sogenannte Selbstkalibrierung, welche allein mehrere Aufnahmen einer unbekanntem Szene benutzt. Ansätze zur traditionellen Kamerakalibrierung sind schon seit langem bekannt und liefern eine präzise Bestimmung der Kameragrößen. Nachteilig ist hierbei, daß ein spezielles und genaues dreidimensionales Kalibrierungsmuster aufgebaut werden muß. In den 90-er Jahren wurden dann Ansätze zur Selbstkalibrierung entwickelt. Diese benötigen kein spezielles Kalibrierungsmuster, doch sind sie auch nicht so präzise. In diesem Praktikumsversuch wird eine Methode vorgestellt, welche die Vorzüge beider Ansätze vereint. Nur ein einfaches ebenes Kalibrierungsmuster, welches durch einen Standarddrucker erzeugt werden kann, ist dazu nötig. Dies wird von mehreren beliebigen Blickwinkeln aus aufgenommen. Da das Kalibrierungsmuster bekannt ist, die Blickwinkel aber nicht, ist das Verfahren ein Hybrid aus Standardkalibrierung und Selbstkalibrierung. Es ist einfach anzuwenden und liefert eine präzise Kalibrierung.

Es wird gezeigt, daß die projektive Geometrie eine geeignete Basis für die Beschreibung von geometrischen 2D/2D und 3D/2D Transformationen darstellt. 2D/2D Transformationen, wie z.B. die Abbildung eines ebenen Kalibrierungsmusters auf die Bildebene, werden ausführlich besprochen. 3D/2D Transformationen, also Abbildungen von Weltpunkten auf Bildpunkte, werden durch Kameras geleistet.

Die dazu notwendigen Grundlagen der projektiven Geometrie werden im zweiten Kapitel präsentiert. Im folgenden Kapitel wird die Modellierung von Kameras besprochen. Im vierten Kapitel werden dann die eigentlichen Kalibriermethoden vorgestellt. Die 3D Kalibrierung wird der Vollständigkeit halber vorgestellt, ist sonst aber nicht Gegenstand des praktischen Teils. Die 2D Kalibrierung ist das Verfahren, welches im praktischen Teil umgesetzt wird. Der kurze Abschnitt der Selbstkalibrierung soll nur die grundsätzliche Idee vermitteln; ansonsten ist das Thema weitaus komplexer. Das fünfte Kapitel behandelt Aspekte der numerischen Berechnung. Die Beschreibung schließt mit einer Darstellung der relevanten Literatur. Die theoretischen Fragen im Aufgabenteil sollten vor dem Praktikums-termin bearbeitet werden, während die praktischen Aufgaben zum Termin selbst umgesetzt werden.

1.2 Projektive Geometrie

1.2.1 Einleitung

Euklidische und projektive Geometrie

Für uns Menschen ist in erster Linie die euklidische Geometrie von Bedeutung, da sie unserer natürlichen Erfahrung des uns umgebenden Raums entspricht. Im euklidischen Raum sind zum Beispiel Größen wie Längen und Winkel definiert. Punkte in der Ebene oder im Raum werden durch zwei- beziehungsweise dreidimensionale Vektoren dargestellt; dies sind ihre euklidischen (oder auch inhomogenen) Koordinaten. Punkte sind identisch, wenn ihre Koordinaten identisch sind. Auch können Abständen zwischen zwei Punkten berechnet werden. Geometrische Gebilde können im euklidischen Raum verschoben und gedreht werden.

Gewisse Sachverhalte sind aber in einer euklidischen Betrachtungsweise nur un- schön zu beschreiben. So wird in der Geometrie der Ebene zwischen Punkten, Li- nien und Richtungen unterschieden. Zum Beispiel haben Linien mit unterschiedli- chen Richtungen einen eindeutigen Schnittpunkt, während Linien mit identischer Richtung keinen oder alle Punkte gemeinsam haben. Weiterhin möchte man eine allgemeinere Klasse von Transformationen zur Verfügung haben, welche nicht nur Verschiebung und Drehung enthält. Betrachten wir zu Veranschaulichung folgen- des Beispiel: Der Mensch, oder eine Kamera, betrachten ein Gebäude, welches typischerweise aus rechtwinkligen Strukturen aufgebaut ist. Das Bild, welches dabei entsteht, ist durch eine perspektivische Projektion entstanden, so daß das wohlbekannte Phänomen von perspektivischen Verzerrungen zu beobachten ist: In der Welt parallele Linien treffen sich im Bild in den sogenannten Fluchtpunk- ten; ursprünglich senkrecht aufeinandertreffende Geraden bilden im Bild keine

rechten Winkel mehr.

Die beschriebenen Phänomene sind mit Hilfe der projektiven Geometrie sehr elegant und einfach zu beschreiben. Der Schlüsselaspekt dabei ist, daß die projektive Geometrie die Erweiterung der euklidischen Geometrie um Punkte im Unendlichen ist. In der Ebene können Linien und Punkte nun einheitlich beschrieben werden. Auch parallele Linien besitzen einen Schnittpunkt, welcher dann nämlich ein Punkt im Unendlichen ist. Unter einer projektiven Transformation können unendlich weit entfernte Punkte auf endlich weit entfernte Punkte abgebildet werden und umgekehrt: Die parallelen Linien des Schachbretts schneiden sich im Unendlichen, während auf dem Bild (oder in den Augen des Betrachters) der Fluchtpunkt dieser Linien in dem Bild selbst, also im Endlichen, liegt. Mit Hilfe der projektiven Geometrie ergeben sich auch für die Modellierung von Kameras wesentlich einfachere Zusammenhänge, da die Abbildung von 3D Weltpunkten auf 2D Bildpunkte eine lineare Form erhält.

Euklidische und projektive Koordinaten

Ein Punkt im n -dimensionalen euklidischen Raum \mathcal{R}^n wird durch einen n -dimensionalen Vektor $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathcal{R}^n$ beschrieben. Dies ist die Repräsentation durch euklidische oder inhomogene Koordinaten.

Ein Punkt im n -dimensionalen projektiven Raum \mathcal{P}^n wird durch einen $(n+1)$ -dimensionalen Vektor $\tilde{\mathbf{x}} = (x_1, \dots, x_{n+1})^T$ beschrieben. Dies ist die Repräsentation durch projektive oder homogene Koordinaten. Homogene Koordinaten werden auch Verhältniskoordinaten genannt. Zwei projektive Punkte $\tilde{\mathbf{x}}$ und $\tilde{\mathbf{y}}$ sind identisch, genau dann wenn ihre homogenen Koordinatenvektoren äquivalent sind. Die Äquivalenz, beschrieben durch das Formelzeichen \sim , bedeutet Gleichheit bis auf einen Skalierungsfaktor:

$$\tilde{\mathbf{x}} \sim \tilde{\mathbf{y}} \quad \Leftrightarrow \quad \tilde{\mathbf{x}} = \lambda \tilde{\mathbf{y}}, \quad \text{mit } \lambda \in \mathcal{R} \setminus \{0\}.$$

Der Nullvektor $\mathbf{0}_{n+1}$ ist kein Element von \mathcal{P}^n , da dieser keiner geometrischen Größe im projektiven Raum entspricht. Der projektive Raum \mathcal{P}^n ist also eine Menge von äquivalenten Vektoren in $\mathcal{R}^{n+1} \setminus \{\mathbf{0}_{n+1}\}$. Euklidische Koordinaten können einfach in projektive konvertiert werden:

$$(x_1, \dots, x_n) = \mathbf{x} \rightarrow \tilde{\mathbf{x}} = (\lambda x_1, \dots, \lambda x_n, \lambda) \quad .$$

Da der euklidische Raum einen Teilraum des projektiven Raums darstellt, können umgekehrt nicht alle Punkte des projektiven Raums in euklidische Koordinaten umgewandelt werden. Nur solche Punkte, für welche die letzte Komponente x_{n+1} nicht Null ist, können auch in inhomogenen Koordinaten umgeschrieben werden:

$$(x_1, \dots, x_n, x_{n+1}) = \tilde{\mathbf{x}} \rightarrow \mathbf{x} = \left(\frac{x_1}{x_{n+1}}, \dots, \frac{x_n}{x_{n+1}} \right) \quad .$$

Punkte der Form $\tilde{x}_\infty = (x_1, \dots, x_n, 0)^T$ sind Punkte im Unendlichen. Der projektive Raum ist der euklidische Raum erweitert durch Punkte im Unendlichen.

1.2.2 Die projektive Ebene

Punkte und Geraden

In der projektiven Ebene werden Punkte und Geraden gleichermaßen als dreidimensionale Vektoren dargestellt. Die Bedingung dafür, daß ein Punkt $\mathbf{x} = (x, y)^T$ auf einer Linie liegt, ist durch die allgemeine Form der Geradengleichung gegeben: $ax + by + c = 0$. Sei die Linie \mathbf{l} definiert durch den Vektor $\mathbf{l} = (a, b, c)^T$, so ist die linke Seite der Geradengleichung nichts anderes als das Skalarprodukt zwischen dem Linienvektor \mathbf{l} und dem Punktvektor: $\tilde{x} = (\mathbf{x}^T, 1)^T$:

$$\mathbf{l}^T \tilde{x} = 0 \quad . \quad (1.1)$$

Eine Gerade ist durch zwei Punkte eindeutig definiert. Würde man den herkömmlichen Weg gehen, so müßte man dazu ein Gleichungssystem bestehend aus zwei Geradengleichungen lösen. Mit projektiven Koordinaten geht dies jedoch einfacher. Sei \mathbf{l} die Gerade, welche durch die Punkte \tilde{x} und \tilde{y} geht. Es gilt sowohl $\mathbf{l}^T \tilde{x} = 0$ als auch $\mathbf{l}^T \tilde{y} = 0$. Der Vektor \mathbf{l} steht also sowohl auf \tilde{x} als auch auf \tilde{y} senkrecht. Der Vektor, welches dieses leistet ist das Vektorprodukt aus \tilde{x} und \tilde{y} :

$$\mathbf{l} \sim \tilde{x} \times \tilde{y} \quad . \quad (1.2)$$

Gleichermaßen ist ein Punkt \tilde{x} eindeutig durch den Schnitt zweier Linien \mathbf{l}_1 und \mathbf{l}_2 definiert, und es ergibt sich:

$$\tilde{x} \sim \mathbf{l}_1 \times \mathbf{l}_2 \quad . \quad (1.3)$$

Man sieht, daß die Rolle von Punkten und Linien vertauschbar ist. Dies nennt man die Dualität von Punkten und Linien in der projektiven Ebene.

Werfen wir noch einen Blick auf die geometrische Bedeutung der einzelnen Elemente eines Linienvektors. Die Geradengleichung wird allgemeiner auch als Hesse Normalenform bezeichnet, wobei der Vektor (a, b) die Normale der Geraden ist. Der Abstand der Geraden zum Ursprung ist dann: $\frac{|c|}{\sqrt{a^2+b^2}}$.

Die Linie im Unendlichen

Punkte im Unendlichen, auch ideale Punkte genannt, haben die Form: $\tilde{x}_\infty = (x, y, 0)^T$. Alle idealen Punkte liegen auf einer einzigen Linie, der Linie im Unendlichen: $\mathbf{l}_\infty = (0, 0, 1)^T$. Ideale Punkte spielen die Rolle von Richtungen. Schneidet

man eine Linie mit der Linie im Unendlichen, so ergibt sich ein idealer Schnittpunkt: $\mathbf{l} \times \mathbf{l}_\infty = (b, -a, 0)^T$. Die ersten beiden Komponenten $(b, -a)$ sind der Richtungsvektor der Geraden.

1.2.3 Transformationen und ihre Hierarchie

Die im folgenden beschriebenen Transformationen werden für die projektive Ebene erläutert. Alle Eigenschaften gelten aber natürlich ebenso für den höherdimensionalen Fall, d.h. den projektiven Raum. Der Reihe nach werden die projektive, affine, ähnliche und euklidische Transformation behandelt und bilden so eine Hierarchie von Transformationen. Die projektive ist die allgemeinste und die euklidische die speziellste Transformation. Die in dieser Reihe folgende Transformation ist eine Untergruppe der vorangehenden, das heißt eine euklidische ist auch eine ähnliche Transformation, eine ähnliche ist auch eine affine Transformation und eine affine ist auch eine projektive Transformation. Umgekehrt ist aber zum Beispiel nicht jede projektive auch eine affine Transformation. Für jede Gruppe von Transformationen ist es wichtig diejenigen geometrischen Größen zu kennen, welche durch die jeweilige Transformation nicht verändert werden. Man spricht von sogenannten Invarianten. Bezogen auf die Hierarchie der Transformationen nimmt die Anzahl der Invarianten von der projektiven zur euklidischen Transformation zu. Jede Invariante der allgemeineren Gruppe ist auch gleichzeitig Invariante der Untergruppe.

Projektive Transformation

Eine projektive Transformation bildet Punkte auf Punkte und Linien auf Linien ab; sie ist invertierbar. Daher ist eine projektive Transformation als lineare Abbildung in den homogenen Koordinaten zu schreiben:

$$\tilde{x}' \sim \mathbf{H}\tilde{x} \quad . \quad (1.4)$$

\mathbf{H} ist eine reguläre 3×3 Matrix, so daß die Abbildung invertierbar ist. Eine projektive Transformation wird auch Homographie genannt (daher auch die Bezeichnung \mathbf{H}). Die obige Gleichung ist homogen, das heißt alle Matrizen, welche sich nur durch einen Skalierungsfaktor unterscheiden, sind äquivalent. Die Matrix enthält 9 Elemente. Abzüglich des Skalierungsfaktors ergeben sich 8 Freiheitsgrade.

Die einzige Invariante einer projektiven Transformation ist das Doppelverhältnis. Seien A, B, C, D Punkte auf einer Geraden, so ist ein Verhältnis von Längen zum Beispiel $\frac{|AB|}{|AC|}$, wobei $|AB|$ die Länge der Strecke von Punkt A zu Punkt B ist. Das

Doppelverhältnis der vier Punkte ist dann: $\frac{|AB||CD|}{|AC||BD|}$. Durch eine Homographie können unendlich weit entfernte Punkte auf endliche Punkte abgebildet werden. In der folgenden Abbildung sieht man, daß der Schnittpunkt von vormals parallelen Geraden ein endlicher Punkt ist (Fluchtpunkt).

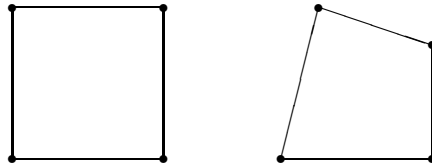


Abbildung 1.1: Variation eines Quadrats unter einer projektiven Transformation \mathbf{H}_P .

Affine Transformation

In inhomogenen Koordinaten schreibt sich eine affine Transformation wie folgt: $\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{t}$. Die beiden Teile, Multiplikation mit \mathbf{A} und die Addition von \mathbf{t} , vereinfachen sich in der homogenen Schreibweise zu einer einzigen Multiplikation:

$$\tilde{\mathbf{x}}' \sim \mathbf{H}_A \tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \tilde{\mathbf{x}} \quad . \quad (1.5)$$

Die Matrix \mathbf{A} ist hierbei regulär. Die affine Transformation bildet endliche auf endliche Punkte und unendliche auf unendliche Punkte ab. Schreiben wir obige Gleichung noch einmal explizit in ihren Komponenten auf:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} \sim \begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad . \quad (1.6)$$

Es ist ersichtlich, daß die affine Transformation 6 Freiheitsgrade (FG) hat: zwei FG für die Translation, ein FG für die Rotation, zwei FG für die Skalierung in x - und y -Richtung und ein FG für die Scherung. Die Werte für Rotation, Skalierung und Scherung stecken implizit in den Elementen von \mathbf{A} . Die affine Transformation weist zusätzlich zum Doppelverhältnis der projektiven Transformation noch folgende Invarianten auf: Parallelität von Geraden bleibt erhalten, Längenverhältnisse auf Geraden und Flächenverhältnisse bleiben konstant.

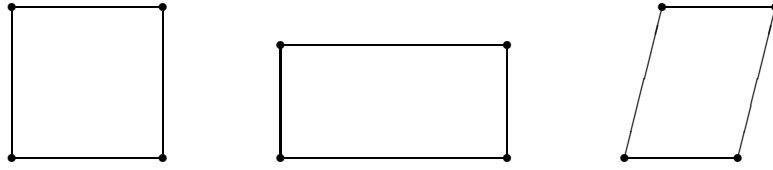


Abbildung 1.2: Variation eines Quadrats unter einer affinen Transformation \mathbf{H}_A .

Ähnlichkeitstransformation

Die Ähnlichkeitstransformation setzt sich aus Translation, Drehung und Skalierung zusammen und enthält somit 4 Freiheitsgrade:

$$\tilde{x}' \sim \mathbf{H}_S \tilde{x} = \begin{pmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \tilde{x} \quad . \quad (1.7)$$

\mathbf{R} ist eine Rotationsmatrix und in den Komponenten geschrieben ergibt sich:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} \sim \begin{pmatrix} s \cos(\alpha) & s \sin(\alpha) & t_x \\ -s \sin(\alpha) & s \cos(\alpha) & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad , \quad (1.8)$$

wobei α der Drehwinkel ist. Unter der Ähnlichkeitstransformation sind Winkel und beliebige Längenverhältnisse invariant.

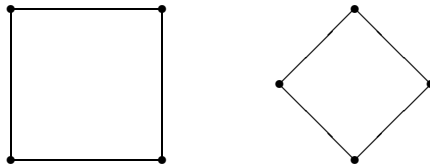


Abbildung 1.3: Variation eines Quadrats unter einer Ähnlichkeitstransformation \mathbf{H}_S .

Euklidische Transformation

Die euklidische Transformation besteht nur aus einer Translation und einer Rotation und besitzt daher 3 Freiheitsgrade:

$$\tilde{x}' \sim \mathbf{H}_S \tilde{x} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \tilde{x} \quad . \quad (1.9)$$

Längen und Flächen sind unter der euklidischen Transformation invariant.

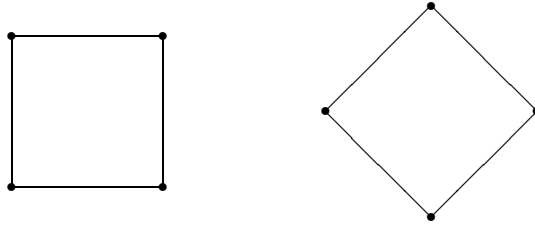


Abbildung 1.4: Variation eines Quadrats unter einer euklidischen Transformation \mathbf{H}_E .

1.2.4 Berechnung einer Homographie

Eine Homographie besitzt 8 Freiheitsgrade. Das heißt, zur Bestimmung einer Homographie müssen 8 Gleichungen bereitgestellt werden. Eine Punktkorrespondenz wie in Gleichung 1.4 liefert 2 Gleichungen (3 Gleichungen minus des Skalierungsfaktors). Folglich sind 4 Punktkorrespondenzen ausreichend, um die Homographie zu berechnen. Schreiben wir die Homographie \mathbf{H} in ihren drei Zeilenvektoren, so ergibt sich:

$$\tilde{x}' \sim \begin{pmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{pmatrix} \tilde{x} \quad . \quad (1.10)$$

Inhomogene Koordinaten bedeuten auch Verhältniskoordinaten. Bildet man das Verhältnis $\frac{x'}{z'}$ und $\frac{y'}{z'}$:

$$\begin{aligned} \frac{x'}{z'} &= \frac{\mathbf{h}_1^T \tilde{x}}{\mathbf{h}_3^T \tilde{x}} \quad \wedge \quad \frac{y'}{z'} = \frac{\mathbf{h}_2^T \tilde{x}}{\mathbf{h}_3^T \tilde{x}} \quad \Leftrightarrow \\ z' \mathbf{h}_1^T \tilde{x} - x' \mathbf{h}_3^T \tilde{x} &= 0 \quad \wedge \quad z' \mathbf{h}_2^T \tilde{x} - y' \mathbf{h}_3^T \tilde{x} = 0 \quad \Leftrightarrow \\ \begin{pmatrix} z' \tilde{x}^T & \mathbf{0} & -x' \tilde{x}^T \\ \mathbf{0} & z' \tilde{x}^T & -y' \tilde{x}^T \end{pmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} &= \mathbf{0} \quad . \end{aligned} \quad (1.11)$$

Kompakt schreibt sich dieses Gleichungssystem als $\mathbf{A}_i \mathbf{h} = \mathbf{0}$, wobei \mathbf{A}_i die bekannten Koordinaten der i -ten Korrespondenz enthält. Der Lösungsvektor $\mathbf{h} = (\mathbf{h}_1^T, \mathbf{h}_2^T, \mathbf{h}_3^T)^T$ enthält die gesuchten Elemente der Homographie. Schreibt man für 4 Korrespondenzen die Gleichung $\mathbf{A}_i \mathbf{h} = \mathbf{0}$ untereinander, so erhält man $\mathbf{A} \mathbf{h} = \mathbf{0}$, wobei \mathbf{A} eine 8×9 Matrix ist. \mathbf{A} hat den Rang 8 und somit gibt es eine nicht triviale Lösung für \mathbf{h} (die triviale Lösung ist $\mathbf{h} = \mathbf{0}$). Für ideale, das heißt rauschfreie Messungen in den Punktkoordinaten gibt es auch für mehr als 4 Korrespondenzpaare eine Lösung für $\mathbf{A} \mathbf{h} = \mathbf{0}$, da der Rang von \mathbf{A} immer noch 8 beträgt. Da jede Messung aber rauschbehaftet ist, gilt dies für reale Punkte nicht mehr. Vielmehr wird nach einer bestmöglichen Lösung gesucht, welcher der ursprünglichen Gleichung am nächsten kommt. Die bestmögliche Lösung für \mathbf{h} ist dann diejenige,

welche den quadratischen Fehler $\|\mathbf{A}\mathbf{h}\|$, also den Abstand zwischen $\mathbf{A}\mathbf{h}$ und dem Nullvektor, minimiert. Die Lösung für \mathbf{h} kann formal folgendermaßen aufgeschrieben werden:

$$\mathbf{h} = \arg \min_{\mathbf{h}} \|\mathbf{A}\mathbf{h}\| \quad \text{unter der Bedingung} \quad \|\mathbf{h}\| = 1 \quad . \quad (1.12)$$

Die Bedingung $\|\mathbf{h}\| = 1$ ist notwendig um die triviale Lösung auszuschließen. Diese Methode zur Berechnung von \mathbf{H} wird *Direct-Linear-Transformation* genannt. Das rein numerische Verfahren zur Lösung ist als kleinste Fehlerquadrat Methode bekannt (siehe auch Abschnitt 1.5.1).

1.3 Kameramodell

Dieser Praktikumsversuch beschränkt sich auf die Modellierung von normalen Video- oder Industriekameras. Das einfachste Modell ist die sogenannte Lochkamera oder perspektivische Kamera. Auch der Begriff Zentralprojektion ist hierfür gebräuchlich. Das Modell der Lochkamera wird durch die Modellierung von Linzenverzerrungen erweitert, welche bei einer großen Anzahl von Kameras auftreten.

Weitere Kameratypen, welche hier nicht behandelt werden, sind die Panorama- oder omnidirektionalen Kameras. Auch die Bildsensoren von Satelliten sind anders geartet und unter dem Begriff Pushbroom-Kameras bekannt.

1.3.1 Die Lochkamera

In einer Lochkamera werden Weltpunkte mittels Zentralprojektion auf die Bildpunkte abgebildet. Wie in der Abbildung 1.5 zu sehen ist, liegen Weltpunkt und Bildpunkt auf einem Strahl, welcher vom Kamerazentrum ausgeht. Das Kamerazentrum liegt im Ursprung des Koordinatensystems der Kamera. Punkte in diesem Koordinatensystem werden mit $\mathbf{X}_C = (X_C, Y_C, Z_C)^T$ bezeichnet. Die Bildebene ist die Ebene $Z_C = f$, wobei f die Brennweite ist. Die optische Achse ist durch den Richtungsvektor $(0, 0, 1)$ gegeben, und der Punkt, wo die optische Achse die Bildebene durchsticht, ist der Hauptpunkt mit den Koordinaten $(0, 0, f)$. Der Bildpunkt $\mathbf{x} = (x, y)^T$ hat die 3D Koordinaten $(x, y, f)^T$. Gemäß des Strahlensatzes gelten dann folgende Verhältnisse:

$$\frac{x}{X_C} = \frac{y}{Y_C} = \frac{f}{Z_C}$$

Für die Abbildung von Weltpunkten auf Bildpunkte gilt dann:

$$(X_C, Y_C, Z_C) \rightarrow \left(\frac{f}{Z_C} X_C, \frac{f}{Z_C} Y_C \right) \quad .$$

Durch Überführung in homogene Koordinaten erhalten wir: $(\frac{f}{Z_C}X_C, \frac{f}{Z_C}Y_C, 1) \sim (fX_C, fY_C, Z_C)$. Die obige nichtlineare Gleichung läßt sich dann als einfache lineare Gleichung schreiben:

$$\begin{pmatrix} fX_C \\ fY_C \\ Z_C \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{pmatrix} \Leftrightarrow \tilde{\mathbf{x}} \sim \mathbf{P}\tilde{\mathbf{X}}_C \quad . \quad (1.13)$$

Die Matrix \mathbf{P} kann in das Produkt einer Diagonalmatrix \mathbf{K} und einer Einheitsprojektionsmatrix \mathbf{P}_p aufgespalten werden:

$$\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \Leftrightarrow \mathbf{P} = \mathbf{K}\mathbf{P}_p \quad .$$

Die Einheitsprojektionsmatrix \mathbf{P}_p kann auch als Aneinanderreihung der 3×3 Einheitsmatrix \mathbf{I} und des Nullvektors $\mathbf{0}$ geschrieben werden: $\mathbf{P}_p = (\mathbf{I}|\mathbf{0})$.

Schlußendlich werden Punkte nicht in dem Kamerakoordinatensystem, sondern in dem Bildkoordinatensystem angegeben. Dort ist die physikalisch Einheit ein Pixel. Ist der Bildvektor \mathbf{x} eine Größe in Pixel, so hat auch die Brennweite f die Einheit Pixel. Normalerweise ist es so, daß die optische Achse nicht auf den Ursprung des Bildkoordinatensystems abgebildet wird. Die ursprünglichen Bildkoordinaten (x, y) erhalten einen Offset (x_0, y_0) durch den Bildhauptpunkt: $(x, y) \rightarrow (x, y) + (x_0, y_0)$. Die Matrix \mathbf{K} wird dann zu:

$$\mathbf{K} = \begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad . \quad (1.14)$$

Im Falle einer realen Kamera kann es vorkommen, daß der Bildsensor aus nicht quadratischen Pixeln aufgebaut ist. Das bedeutet, daß das Bild in vertikaler und horizontaler Richtung unterschiedlich skaliert ist. Ein Quadrat wird dann auf ein Rechteck abgebildet. Folglich ergibt sich:

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad . \quad (1.15)$$

Manchmal kommt es vor, daß die optische Achse des Objektivs nicht senkrecht auf dem Bildsensor steht. In diesem Fall tritt eine Scherung auf und rechte Winkel erscheinen leicht schiefwinkelig im Bild. Dieser Einfluß wird durch den Scherungsparameter s ausgedrückt:

$$\mathbf{K} = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad . \quad (1.16)$$

Normalerweise werden Weltpunkte nicht relative zu dem Kamerakoordinatensystem beschrieben, sondern relative zu einem eigenen Weltkoordinatensystem. Kamera- und Weltkoordinatensystem sind durch eine euklidische Transformation miteinander verbunden: $\mathbf{X}_C = \mathbf{R}\mathbf{X} + \mathbf{t}$. Die Projektionsmatrix \mathbf{P} läßt sich dann schreiben als:

$$\mathbf{P} = \mathbf{K}(\mathbf{I}|\mathbf{0}) \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} = \mathbf{K}(\mathbf{R}|\mathbf{t}) \quad . \quad (1.17)$$

Die *internen Kameraparameter* werden durch die Kalibriermatrix \mathbf{K} beschrieben. Hier sind alle Größen über die Kamera als solche enthalten. Die *externen Kameraparameter* beschreiben die Transformation von Welt- in Kamerakoordinaten in Form der Rotationmatrix \mathbf{R} und des Translationsvektors \mathbf{t} .

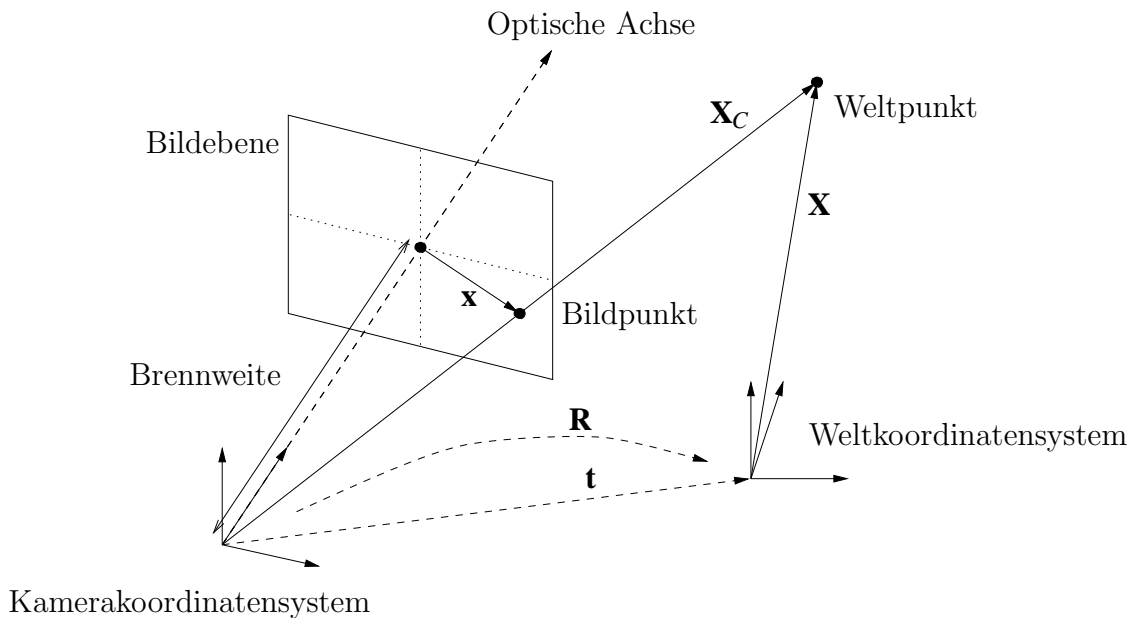


Abbildung 1.5: Modell der Lochkamera

1.3.2 Linsenverzerrungen

Bisher wurde angenommen, daß ein lineares Modell geeignet ist um reale Kameras nachzubilden. In einem linearen Modell, also einer projektive 3D zu 2D Transformation, liegen Weltpunkt, Bildpunkt und optisches Zentrum auf demselben optischen Strahl. Weiterhin werden Geraden als Geraden abgebildet. Bei vielen realen Kameras trifft dies allerdings nicht zu. Besonders dann, wenn die Brennweite klein ist, das heißt das Blickfeld ist groß (Weitwinkelobjektiv), werden Geraden auf krumme Linien abgebildet. Oft treten Linsenverzerrungen bei

eher billigen Objektiven auf. In der Regel sind die auftretenden Linsenverzerrung radiale Linsenverzerrungen, das heißt, das Maß der Verzerrung ist abhängig von dem Abstand des Bildpunktes zum Bildmittelpunkt. Der Effekt der Linsenverzerrung ist in der Abbildung 1.6 dargestellt. In diesem Fall erfährt das Quadrat eine konvexe Deformation; eine konkave Deformation kann aber ebenfalls auftreten.

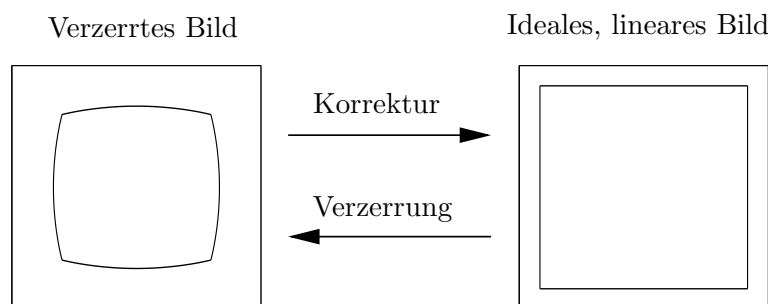


Abbildung 1.6: Effekt einer radialen Linsenverzerrung bei der Abbildung eines Quadrats.

Seien $\mathbf{x} = (x, y)^T$ die idealen, linearen Bildkoordinaten und $\mathbf{x}_d = (x_d, y_d)^T$ die realen, verzerrten Bildkoordinaten. Das Zentrum der Verzerrung sei der Bildhauptpunkt \mathbf{x}_0 . Die Linsenverzerrung kann dann folgendermaßen modelliert werden:

$$(\mathbf{x}_d - \mathbf{x}_0) = L(r)(\mathbf{x} - \mathbf{x}_0) \quad , \quad (1.18)$$

wobei r der Abstand von \mathbf{x} zum Bildhauptpunkt ist: $r = \|\mathbf{x} - \mathbf{x}_0\|$. Die Linsenfunktion L nimmt für den Bildhauptpunkt den Wert eins an: $L(0) = 1$. Ansonsten ist L unbekannt und wird daher durch einer Taylorreihenentwicklung um $r = 0$ approximiert: $L(r) = 1 + \kappa_1 r + \kappa_2 r^2 + \kappa_3 r^3 + \dots$. Angenommen eine quadratische Approximation ist ausreichend genau, obige Formel schreibt sich dann wie folgt:

$$\mathbf{x}_d = \mathbf{x} + (\mathbf{x} - \mathbf{x}_0)(\kappa_1 r + \kappa_2 r^2) \quad . \quad (1.19)$$

Kamerakalibrierung bedeutet also nicht nur die Bestimmung der Elemente der Kalibriermatrix \mathbf{K} , sondern auch der Verzerrungsparameter $\{\kappa_1, \kappa_2, \kappa_3, \dots\}$.

Anmerkung

Sei \mathbf{x}' der virtuelle Bildpunkt, welcher bei der Abbildung mit einer Einheitskalibriermatrix $\mathbf{K} = \mathbf{I}$ entsteht: $\tilde{x}' \sim (\mathbf{R} | \mathbf{t}) \tilde{X}$. Der tatsächliche Bildpunkt \tilde{x} entsteht dann durch: $\tilde{x} \sim \mathbf{K} \tilde{x}'$. Die oben beschriebene Modellierung der Linsenverzerrung kann auf die Punkte \mathbf{x} oder \mathbf{x}' angewendet werden. Rein theoretisch betrachtet ist dies äquivalent. Unter numerischen Gesichtspunkten aber ist die Modellierung der Verzerrung bezüglich der Punkte \mathbf{x}' günstiger als die der Punkte \mathbf{x} . Die Begründung dafür ist analog zu den Überlegungen bei der Normalisierung von

Daten, welche im Abschnitt 1.5.2 behandelt wird. Für eine Modellierung mit n Verzerrungsparametern und einem repräsentativen Bildpunkt von $\mathbf{x} = (100, 100)$, ist $r^n = \|\mathbf{x} - \mathbf{x}_0\|^n$ in der Größenordnung von 1 bis 10^{2n} . Benutzt man stattdessen den virtuellen Bildpunkt mit $r' = \|\mathbf{x}'\|$, so ist r^n typischerweise kleiner Eins. Numerische Instabilitäten bei der nicht linearen Optimierung von mehr als zwei Verzerrungsparametern werden dadurch vermieden.

1.3.3 Projektion einer Ebene

Von besonderem Interesse ist die Projektion einer Ebene in die Bildebene, da dies Teil des 2D Kalibriervorgangs ist, welcher im nächsten Kapitel behandelt wird. Sei die Ebene die X/Y -Ebene in Weltkoordinaten. Die Gleichung der Ebene ist dann: $Z = 0$. Die Rotationsmatrix \mathbf{R} läßt sich in ihren drei Spaltenvektoren auch als $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ schreiben. Für die Abbildung von Punkten einer Ebene ergibt sich dann:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \mathbf{K}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{t}) \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = \mathbf{K}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}) \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} . \quad (1.20)$$

Bezeichnen wir mit $\tilde{X}_E = (X, Y, 1)^T$ den 2D projektiven Punkt in der besagten Weltebene. Man erkennt, daß die projektive Abbildung der Kamera von 3D Weltpunkten nach 2D Bildpunkten nun die Form einer 2D zu 2D Abbildung erhält:

$$\tilde{x} \sim \mathbf{H}\tilde{X}_E \quad \text{mit} \quad \mathbf{H} = \mathbf{K}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}) . \quad (1.21)$$

Auf der Basis von mindestens vier Punktkorrespondenzen kann die Homographie \mathbf{H} berechnet werden. Ausgehend davon können Bedingungen für die Kalibriermatrix \mathbf{K} abgeleitet werden und zur Bestimmung dieser genutzt werden.

Kennt man bereits die Kamera \mathbf{K} so kann mittels der Homographie die Position und Orientierung der Ebene berechnet werden. Mit $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3)$ gilt:

$$(\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3) = \lambda \mathbf{K}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}) . \quad (1.22)$$

Die Rotationsmatrix $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ ist eine orthogonale Matrix mit $\|\mathbf{r}_i\| = 1$ und $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$. \mathbf{K} ist definitionsgemäß invertierbar und mit $\lambda = \|\mathbf{K}^{-1}\mathbf{h}_1\|$ folgt:

$$(\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}) = \mathbf{K}^{-1}\mathbf{H}/\|\mathbf{K}^{-1}\mathbf{h}_1\| . \quad (1.23)$$

Das heißt, daß auf der Basis von vier bekannten Punkten auf einer Ebene und einer kalibrierten Kamera die Lage dieser Ebene im Raum berechnet werden kann.

1.4 Kalibrierung

1.4.1 Einleitung und Überblick

Dieses Kapitel gliedert sich in drei Abschnitte. Im ersten Abschnitt wird das Standardverfahren für dreidimensionale Kalibriermuster vorgestellt. Eine lineare und eine nicht lineare Methode wird beschrieben. Selbst wenn diese Methoden im Rahmen dieses Versuchs keine praktische Umsetzung finden, sind sie von Bedeutung und wichtig für das allgemeine Verständnis des Themenkomplexes. Im zweiten Abschnitt wird dann eine Methode vorgestellt, welche den Vorteil einer einfachen Umsetzung aufweist: Nur ein einziges ebenes Kalibriermuster muß erstellt werden. Auf Grund dieses enormen praktischen Vorteils wird diese Methode auch im praktischen Teil implementiert werden. Zur theoretischen Untermauerung des 2D Kalibrierverfahrens wird im letzten Abschnitt das Thema der Selbstkalibrierung kurz angeschnitten.

Das Verfahren nach Tsai wird noch häufig in der Literatur zitiert und in der Praxis auch verwendet. Es wurde im Jahr 1987 veröffentlicht und hat seitdem eine weite Verbreitung erfahren, auch weil lauffähiger Programmcode dafür im Netz zu finden ist. Vom Stand der Forschung aus betrachtet ist dieses Verfahren jedoch veraltet. Für die 3D Kalibrierung ist das hier vorgestellte Verfahren wesentlich übersichtlicher und sollte daher verwendet werden. Oft ist zu lesen, daß das Tsai-Verfahren auch mit nur einer Kalibrierebene angewendet werden kann. Dies ist nur teilweise richtig, da als Voraussetzung die Pixelgröße des Sensors bekannt sein muß. Ohne diese Vorkenntnis ist es nicht möglich eine Kalibrierung mit nur einer Ebene durchzuführen, was in diesem Kapitel auch gezeigt wird.

1.4.2 Kalibriermuster

Ein Kalibriermuster kann aus einer Kalibrierebene (2D Kalibrierung), oder mehreren Kalibrierebenen (3D Kalibrierung) bestehen. Die Kalibriermarkierungen selbst bestehen aus schwarz-weiß Mustern, wobei exakt bekannte Punkte auf den Mustern gut zu lokalisierende Punktmerkmale im Bild erzeugen. Im einfachsten Fall wird das Muster auf Papier gedruckt und dann auf einer soliden Platte aufgezogen. Um hochpräzise Kalibriermuster zur erhalten, können die Muster auch mit Hilfe von CNC Maschinen in Metall gefräßt werden.

Die einzelnen Markierungen können unterschiedliche Formen haben. Wie in der Abbildung 1.7(a) zu sehen, besteht hier das Muster aus regelmäßig angeordneten Quadraten. Typischerweise liegen die Muster auf einem rechtwinkligen Gitternetz mit identischen Abständen in den Gitterpunkten. Mit Hilfe eines Kantendetektors werden zuerst die Kanten der Quadrate bestimmt. An diese Kanten

werden Linien angepaßt. Die Schnittpunkte der Linien bilden die Eckpunkte der Quadrate. Dies hat den Vorteil, daß pro Quadrat vier Meßpunkte zur Verfügung stehen. Bei starker Linsenverzerrung ist allerdings zu beachten, daß die Bestimmung der Eckpunkte als Schnitt der Linien ungenau wird.

Eine andere Möglichkeit, welche auch im Rahmen dieses Versuches verwendet wird, ist ein regelmäßiges Muster bestehend aus Kreisen oder Kreisscheiben, siehe Abbildung 1.7(b). Als Nachteil ist die geringere Anzahl an Markierungspunkten zu nennen. Der Vorteil ist die einfache und robuste Extraktion der Markierungsmerkmale: Das Markierungsmerkmal ist der Kreismittelpunkt. Mit Hilfe einer einfachen Segmentierung werden die Kreisscheiben vom Hintergrund getrennt und deren Schwerpunkte berechnet (Siehe auch Abschnitt 1.4.4).

Zum Schluß noch eine Überlegung bezüglich der notwendigen Anzahl an Kalibriermarkierungen. Einmal ist es einsichtig, daß das Kalibriermuster möglichst den Großteil des Bildes abdecken sollte. Wird zum Beispiel das Kalibriermuster so weit entfernt von der Kamera aufgestellt, daß es nur den zentralen Bildteil abdeckt, so wird die Kalibrierung bezüglich der Verzerrungsparameter schlecht ausfallen, weil die Linsenverzerrung sich zum Rand des Bildes hin stärker ausprägt. Weiterhin gibt es eine Daumenregel, welche besagt, daß die Anzahl an notwendigen Korrespondenzen die Anzahl der zu bestimmenden Parameter um den Faktor fünf übersteigen sollte. Das heißt, das Kalibriermuster sollte circa 30 bis 40 Markierungen aufweisen.

1.4.3 3D Kalibrierung

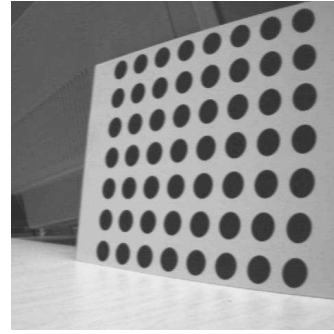
3D Kalibriermuster

Für eine komplette Kalibrierung ist es notwendig, daß die Punkte des Kalibrierusters nicht alle auf einer Ebene liegen. Ein Kalibriermuster besteht daher typischerweise aus zwei oder drei ebenen Flächen, auf die Kalibriermarkierungen aufgebracht sind. In der folgenden Abbildung ist der Fall von zwei Ebenen zu sehen. Der Winkel zwischen den beiden Ebenen muß dabei genau vermessen werden.

Eine weitere Möglichkeit nicht koplanare 3D Punkte zu erhalten ist die Verwendung einer einzelnen Kalibrierfläche, welche mit Hilfe einer mechanischen Einrichtung exakt in Richtung der Normalen verschoben wird. Der Nachteil hierbei ist der erhöhte Aufwand, welcher durch die notwendige Verschiebeeinrichtung entsteht.



(a) 3D Kalibriermuster



(b) 2D Kalibriermuster

Abbildung 1.7: Dreidimensionales Kalibriermuster bestehend aus zwei einzelnen schiefwinkligen Ebenen (a) und ein einfaches 2D Kalibriermuster (b).

Ein linearer Algorithmus

Nehmen wir an, daß eine Anzahl von Korrespondenzen $\tilde{X}_j \leftrightarrow \tilde{x}_j$ zwischen Weltpunkten \tilde{X}_j und Bildpunkten \tilde{x}_j gegeben ist. Das Modell einer projektiven Kamera ist die 3×4 Projektionsmatrix \mathbf{P} . Seien $\mathbf{p}_1^T, \mathbf{p}_2^T, \mathbf{p}_3^T$ die drei Zeilenvektoren von \mathbf{P} , so läßt sich schreiben:

$$\tilde{x}_j \sim \mathbf{P}\tilde{X}_j = \begin{pmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{pmatrix} \tilde{X}_j \quad . \quad (1.24)$$

In dem Kontext der Kamerakalibrierung sind die Punkte \tilde{x}_j tatsächliche Messungen in dem Bild und liegen somit im Endlichen: \tilde{x}_j läßt sich schreiben als $\tilde{x}_j = (x_j, y_j, 1)^T$. Schreibt man die Äquivalenz $\tilde{x}_j \sim \mathbf{P}\tilde{X}_j$ als Gleichung $(\lambda x_j, \lambda y_j, \lambda)^T = \mathbf{P}\tilde{X}_j$, so folgt:

$$x_j = \frac{\mathbf{p}_1^T \tilde{X}_j}{\mathbf{p}_3^T \tilde{X}_j} \quad , \quad y_j = \frac{\mathbf{p}_2^T \tilde{X}_j}{\mathbf{p}_3^T \tilde{X}_j} \quad . \quad (1.25)$$

Diese Gleichungen lassen sich in folgendes Gleichungssystem umformen:

$$\begin{pmatrix} -\tilde{X}_j^T & \mathbf{0}^T & x_j \tilde{X}_j^T \\ \mathbf{0}^T & -\tilde{X}_j^T & y_j \tilde{X}_j^T \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \mathbf{0} \quad \Leftrightarrow \quad \mathbf{A}_j \mathbf{p} = \mathbf{0} \quad . \quad (1.26)$$

Beachten Sie die Ähnlichkeit dieser Gleichung mit der Gleichung 1.11. Die Elemente der 2×12 Matrix \mathbf{A}_j bestehen nur aus den Koordinaten der Bildmessung und der bekannten Koordinaten des Kalibrierusters. Der Lösungsvektor $\mathbf{p} = (\mathbf{p}_1^T, \mathbf{p}_2^T, \mathbf{p}_3^T)^T$ besteht aus den 12 Elementen von \mathbf{P} , welche zu bestimmen sind. Stehen n Korrespondenzen zur Verfügung, so kann man n Gleichungssysteme dieser Form übereinander stapeln und erhält das Gleichungssystem $\mathbf{A}\mathbf{p} = \mathbf{0}$, wobei \mathbf{A} eine $2n \times 12$ Matrix ist. Da die Matrix \mathbf{P} bis auf einen Skalierungsfaktor

definiert ist, enthält sie $12 - 1 = 11$ Unbekannte. 11 Gleichungen sind für eine eindeutige Lösung in \mathbf{p} notwendig. Jede Korrespondenz liefert zwei Gleichungen. Somit sind 6 Korrespondenzen notwendig, wobei eine Gleichung der insgesamt 12 bereitgestellten Gleichung ungenutzt bleiben kann.

Jede reale Messung ist mit Rauschen behaftet, und somit gilt nicht mehr die Projektionsgleichung $\tilde{x}_j = \lambda \mathbf{P} \tilde{\mathbf{X}}_j$. Daraus folgt, daß auch das Gleichungssystem $\mathbf{A}\mathbf{p} = \mathbf{0}$ für $n \geq 6$ Punkt Korrespondenzen keine nicht triviale Lösung mehr besitzt (Die triviale Lösung ist $\mathbf{p} = \mathbf{0}$). Stattdessen wird im Sinne des kleinsten Fehlerquadrates (siehe auch Abschnitt 1.5.1) nach einer Lösung gesucht, welche den Fehler $\|\mathbf{A}\mathbf{p}\|$ minimiert. Da die triviale Lösung auszuschließen ist, und da \mathbf{P} und somit auch \mathbf{p} bis auf einen Skalierungsfaktor definiert ist, wird \mathbf{p} auf eins normiert:

$$\mathbf{p} = \arg \min_{\mathbf{p}} \|\mathbf{A}\mathbf{p}\| \quad \text{mit} \quad \|\mathbf{p}\| = 1 \quad . \quad (1.27)$$

Dies ist der *Direct-Linear-Transformation* Algorithmus zur Berechnung der projektiven Kamera \mathbf{P} . Ausgehend von \mathbf{P} können dann die internen Kameraparameter \mathbf{K} und die externen Kameraparameter \mathbf{R}, \mathbf{t} berechnet werden. Für reale Kameras mit Linsenverzerrungen ist nur die nichtlineare Methode geeignet, welche im nächsten Abschnitt beschrieben wird.

Ein nichtlinearer Algorithmus

Lineare Methoden können immer dann eingesetzt werden, wenn die bestimmenden Gleichungen linear in den gesuchten Größen sind. Sind die bestimmenden Gleichungen nicht linear in den gesuchten Größen, kann zur Lösung auch nur ein nichtlinearer Algorithmus benutzt werden. Vergleichbar mit dem oben beschriebenen linearen Ansatz wird auch der nichtlineare Ansatz in Form eines Minimierungsproblems formuliert. Basis ist daher die geeignete Wahl einer Fehlerfunktion. Für das Problem der Kamerakalibrierung wird der euklidische Abstand d_j zwischen dem gemessenen Bildpunkt \mathbf{x}_j und dem Bild $\hat{\mathbf{x}}_j$ des Weltpunktes \mathbf{X}_j gewählt: $d_j = \|\mathbf{x}_j - \hat{\mathbf{x}}_j\| = \sqrt{(x_j - \hat{x}_j)^2 + (y_j - \hat{y}_j)^2}$, wobei zum Beispiel für die Lochkamera $\hat{\mathbf{x}}_j \sim \mathbf{P}\tilde{\mathbf{X}}_j$ gilt. Das Fehlermaß d_j wird auch als geometrischer Fehler bezeichnet.

Sei \mathbf{p} der Satz an Parametern, welche ein gegebenes Kameramodell beschreiben, so ist der projizierte Punkt \hat{x} eine Funktion dieser Parameter: $\hat{x} = \hat{x}(\mathbf{p}; \mathbf{X})$. Kamerakalibrierung kann dann als Optimierungsproblem über dem Parameterraum \mathbf{p} formuliert werden:

$$\mathbf{p} = \arg \min_{\mathbf{p}} \sum_j d_j^2 = \arg \min_{\mathbf{p}} \sum_j d(\mathbf{x}_j, \hat{x}(\mathbf{p}; \mathbf{X}_j))^2 = \arg \min_{\mathbf{p}} \sum_j \|\mathbf{x}_j - \hat{x}(\mathbf{p}; \mathbf{X}_j)\|^2 \quad . \quad (1.28)$$

Der Parameterraum \mathbf{p} kann zum Beispiel aus den Elementen der projektiven Kamera \mathbf{P} bestehen. Wird die Kamera einschließlich Linsenverzerrung kalibriert, so gilt $\mathbf{p} = (\boldsymbol{\kappa}, \mathbf{K}, \mathbf{R}, \mathbf{t})$. Das Optimierungsproblem selbst wird dann iterativ gelöst, zum Beispiel durch den Levenberg-Marquardt Algorithmus. Jeder iterative Algorithmus bedarf einer Startlösung. Diese wird typischerweise durch eine lineare Lösung bereitgestellt. Da für Kameras mit Linsenverzerrung kein linearer Ansatz hergeleitet werden kann, wird als Startlösung die lineare Lösung der projektiven Kamera benutzt. Ausgehend von \mathbf{P} werden dann $\mathbf{K}, \mathbf{R}, \mathbf{t}$ berechnet. Die erste Iteration startet dann mit Werten von $\boldsymbol{\kappa} = \mathbf{0}$. Im Gegensatz zu einem linearen Ansatz garantiert eine iterative Optimierungsmethode im allgemeinen nicht das Finden der global richtigen Lösung. Abhängig von der verwendeten Startlösung wird nur lokale Optimalität erreicht. In dem hier vorliegenden Fall der Kamerakalibrierung kann allerdings davon ausgegangen werden, daß die lineare Startlösung, welche ja nur die Linsenverzerrung nicht in der Modellierung mit einschließt, globale Optimalität gewährleistet.

1.4.4 2D Kalibrierung

Kalibriervorgang

Wie in Gleichung 1.20 gezeigt wurde, ist die Abbildung von koplanaren Punkten (alle Punkte liegen in einer Ebene) ein besonderer Fall. Die Projektionsmatrix \mathbf{P} kann dann auf eine Homographie \mathbf{H} reduziert werden. Schreibt man die Projektionsmatrix in ihren Spaltenvektoren $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4)$ so ergibt sich daraus für die Homographie: $\mathbf{H} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4)$. Betrachten man bekannte Kalibrierpunkte auf einer Ebene so kann mit Hilfe dieser 2D-2D Korrespondenzen die Homographie berechnet werden. Eine komplette Kalibrierung von \mathbf{P} ist nicht möglich, da nur 8 der insgesamt 11 Parameter bestimmt werden können. Was passiert nun, wenn man mehrere ebene Kalibriermuster der Kamera präsentiert, wobei die Position und Orientierung der einzelnen Ebenen unbekannt ist? In diesem Abschnitt wird gezeigt, daß drei Ebenen zur Berechnung der internen Kameraparameter ausreichend sind.

Der Einfachheit halber wird für den Kalibriervorgang ein einziges ebenes Kalibriermuster aufgebaut. Dieses wird von drei unterschiedlichen Richtungen aus aufgenommen, wie in folgender Abbildung zu sehen ist.

Lineare Lösung

Jede Aufnahme der Kalibrierfläche stellt eine Homographie \mathbf{H} zur Verfügung. Schaut man sich noch einmal Gleichung 1.22 an, so ergibt sich für die beiden

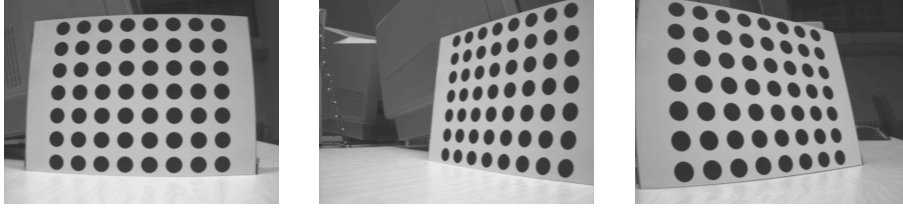


Abbildung 1.8: Drei Aufnahmen eines ebenen Kalibrierungsmusters unter verschiedenen Orientierungen.

Vektoren $\mathbf{r}_1, \mathbf{r}_2$:

$$\mathbf{r}_1 = \lambda^{-1} \mathbf{K}^{-1} \mathbf{h}_1 \quad , \quad \mathbf{r}_2 = \lambda^{-1} \mathbf{K}^{-1} \mathbf{h}_2 \quad . \quad (1.29)$$

Die beiden Vektoren $\mathbf{r}_1, \mathbf{r}_2$ sind orthonormal mit $\mathbf{r}_1^T \mathbf{r}_2 = 0$ und $\|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1$. Eingesetzt in diese beiden Bedingungen ergibt sich:

$$\mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 = 0 \quad (1.30)$$

$$\mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 \quad . \quad (1.31)$$

Dies sind zwei Bedingungen für die internen Kameraparameter in \mathbf{K} . Da die Homographie 8 Freiheitsgrade besitzt und 6 externe Kameraparameter existieren, kann es auch nur zwei Bedingungen für die internen Parameter geben. Im allgemeinen Fall sind 5 interne Kameraparameter zu bestimmen. Dafür sind dann drei Homographien nötig. Ist die Kamera zum Beispiel frei von Scherung, dann reichen zwei Homographien zur Berechnung von \mathbf{K} aus. Die Matrix $\mathbf{K}^{-T} \mathbf{K}^{-1}$ beschreibt das Bild des absoluten Kegelschnitts; mehr dazu ist im Abschnitt zur Selbstkalibrierung (1.4.5) zu finden.

Da die Matrix $\mathbf{B} = \mathbf{K}^{-T} \mathbf{K}^{-1}$ symmetrisch ist, enthält sie 6 unabhängige homogene Elemente:

$$\mathbf{K}^{-T} \mathbf{K}^{-1} = \mathbf{B} = \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{pmatrix} \quad , \quad (1.32)$$

welche in dem Vektor $\mathbf{b} = (B_{11}, B_{12}, B_{13}, B_{22}, B_{23}, B_{33})$ zusammengefaßt werden können. Sei der Vektor \mathbf{v}_{ij} folgendermaßen definiert:

$$\mathbf{v}_{ij} = (h_{i,1}h_{j,1}, h_{i,1}h_{j,2} + h_{i,2}h_{j,1}, h_{i,1}h_{j,3} + h_{i,3}h_{j,1}, h_{i,2}h_{j,2}, h_{i,2}h_{j,3} + h_{i,3}h_{j,2}, h_{i,3}h_{j,3})^T \quad .$$

Die Bedingungen 1.30 und 1.31 lassen sich dann schreiben:

$$\begin{pmatrix} \mathbf{v}_{12}^T \\ \mathbf{v}_{11}^T - \mathbf{v}_{22}^T \end{pmatrix} \mathbf{b} = \mathbf{0} \quad . \quad (1.33)$$

Stehen n Homographien zur Verfügung, so erhält man durch Übereinanderstapelung eine $2n \times 6$ Matrix \mathbf{V} , so daß gilt:

$$\mathbf{V} \mathbf{b} = \mathbf{0} \quad . \quad (1.34)$$

Für ideale rauschfreie Messungen ist dieses System eindeutig lösbar. Für rauschbehaftete Messungen ist die Lösung im Sinne des kleinsten Fehlerquadrats (siehe Abschnitt 1.5.1) zu berechnen.

Die Inverse der Kalibriermatrix \mathbf{K} ist ebenfalls eine obere Dreiecksmatrix (alle Elemente unterhalb der Diagonalen sind Null), und sei mit \mathbf{G} bezeichnet. Es gilt, daß das Produkt $\mathbf{G}^T \mathbf{G}$ einer oberen Dreiecksmatrix eine symmetrische und positive definite Matrix ergibt (Pos. Definitheit bedeutet, daß alle Eigenwerte positiv sind.). Umgekehrt gilt, daß jede symmetrische, positiv definite Matrix sich als Produkt $\mathbf{G}^T \mathbf{G}$ einer oberen Dreiecksmatrix schreiben läßt. Dies ist die sogenannte Cholesky Faktorisierung. Haben wir \mathbf{G} aus \mathbf{B} berechnet, so folgt durch Invertierung die Kalibriermatrix $\mathbf{K} = \mathbf{G}^{-1}$.

Wichtig bei dieser Art der Kalibrierung ist, daß die Kalibrierflächen nicht parallel zueinander sind, da sonst die Rotationsmatrizen \mathbf{R} identisch sind und die Gleichungen 1.30 und 1.31 keine zusätzlichen Bedingungen liefern.

Nichtlineare Lösung

Ausgehend von der oben gewonnenen linearen Lösung kann eine nichtlineare Optimierung angehängt werden, welche die Bestimmung der Linsenverzerrung mit einschließt. Man verwendet wieder den geometrischen Fehler wie in Gleichung 1.28 definiert. Für jede Aufnahme i der Kalibrierfläche gibt es einen Satz von externen Kameraparametern $\mathbf{R}_i, \mathbf{t}_i$. Der Kalibrierpunkt \mathbf{X}_j wird in dem Bild i auf den Bildpunkt $\hat{\mathbf{x}}_{ij} = \hat{\mathbf{x}}(\mathbf{K}, \mathbf{k}, \mathbf{R}_i, \mathbf{t}_i; \mathbf{X}_j)$ abgebildet. \mathbf{x}_{ij} ist der im Bild gemessene Punkt. Der zu minimierende Gesamtfehler ist dann die Summation über alle Bilder und über alle Bildpunkte:

$$\min_{(\mathbf{K}, \mathbf{k}, \mathbf{R}_i, \mathbf{t}_i)} \sum_i \sum_j \|\mathbf{x}_{ij} - \hat{\mathbf{x}}(\mathbf{K}, \mathbf{k}, \mathbf{R}_i, \mathbf{t}_i; \mathbf{X}_j)\|^2 \quad . \quad (1.35)$$

Dieses Problem wird mit einem nichtlinearen Optimierungsalgorithmus, zum Beispiel vom Typ Levenberg-Marquardt, gelöst.

Genauigkeit der Markierungen

Um bewerten zu können, wie gut die berechnete Lösung ist, wird der geometrische Fehler mit der Genauigkeit verglichen, mit welcher die einzelnen Markierungspunkte gemessen wurden. Der Meßfehler ist die untere Grenze für die Genauigkeit der berechneten Lösung.

Sei im Bild eine schwarze Kreisscheibe vor weißem Hintergrund mit dem Radius r gegeben, so gibt es $2\pi r$ Randpixel. Mittels einer Schwellwertbildung wird die Kreisscheibe segmentiert. Diese Diskretisierung verursacht einen Fehler von ± 0.5

Pixeln in den Randpixeln. Die Standardabweichung als Maß für diesen Fehler beträgt dann circa 0.3 Pixel. Berechnen wir nun den Schwerpunkt aller Randpixel, so ist die Standardabweichung des Schwerpunktes $0.3/\sqrt{2\pi r}$. Da die inneren Pixel der Kreisscheibe fehlerfrei segmentiert werden, ist der Fehler des Flächenschwerpunktes der Kreisscheibe $0.3/\sqrt{2\pi r}(2\pi r)/(\pi r^2)$. Bei einem Radius von 10 Pixeln ergibt sich, daß der Schwerpunkt fast sicher mit einer Genauigkeit von einigen 10^{-2} Pixeln berechnet werden kann.

Weiterhin ist noch der systematische Fehler zu untersuchen, welcher entsteht, wenn Kreise projektiv verzerrt werden. Unter einer allgemeinen projektiven Transformation wird ein Kreis auf eine Ellipse abgebildet. Dabei gilt, daß der Schwerpunkt des transformierten Kreises nicht identisch mit der Transformierten des ursprünglichen Schwerpunkt ist. Das heißt, der berechnete Schwerpunkt ist nicht das exakte Bild des Schwerpunkts auf dem Kalibriermuster. Dieser so entstandene Fehler ist abhängig von dem Abstand Z des Musters zur Kamera und dem Radius R des Kreises. Eine obere Schranke für diesen (relativen) Fehler ist $\frac{R^2}{Z^2}$. Dies gilt für den Flächenschwerpunkt; die obere Fehlerschranke für den Linienschwerpunkt, das ist der Schwerpunkt der Randpunkte, liegt nur bei $\frac{R}{Z}$. Man sieht auch hier wieder, daß als Markierung der Flächenschwerpunkt dem Linienschwerpunkt vorzuziehen ist. Beträgt zum Beispiel der mittlere Abstand des Kalibrierusters 30cm und der Radius des Kreises 1cm, so liegt der Fehler bei 10^{-3} . Bei der hier eingesetzten Webcam ist der Abbildungsmaßstab circa 1cm zu 10 Pixel. Das heißt, der systematische Fehler durch die projektive Verzerrung beträgt größenordnungsmäßig 10^{-2} Pixel.

1.4.5 Selbstkalibrierung

Die hier vorgestellten Kalibrierverfahren beruhen auf der Kenntnis der Position von 3D Weltpunkten. Jetzt stellt sich natürlich folgende Frage: Ist eine Kalibrierung der Kamera auch dann noch möglich, wenn keine Kenntnis über die Weltpunkte vorliegt? Die Antwort ist ja! Der Vorgang der Berechnung von Kameraparametern ohne Wissen über die 3D Geometrie wird Selbstkalibrierung genannt. Selbstkalibrierung wird immer dann eingesetzt, wenn kein Kalibriermuster zur Verfügung steht. Steht ein Kalibriermuster zur Verfügung, dann sollte dies zur Kalibrierung auch verwendet werden, da die Ausnutzung der zusätzlichen Information in Form der bekannten Weltkoordinaten die Genauigkeit der Kalibrierung erhöht. Auch ist die Implementierung der Kalibrierung dann einfacher.

Nehmen wir an, daß eine Menge von Weltpunkten \tilde{X}_j mittels mehrerer Kameras \mathbf{P}_i auf Bildpunkte \tilde{x}_{ij} abgebildet wird: $\tilde{x}_{ij} \sim \mathbf{P}_i \tilde{X}_j$. Ist die Anzahl der Kameras mindestens zwei, so ist es möglich nur auf Grund der Bildpunkte \tilde{x}_{ij} die Kameras \mathbf{P}_i zu berechnen. Mit bekannten Kameras können dann die Weltpunkte rekonstru-

iert werden. Dies bezeichnet man dann als Tiefen- oder Stereosehen. Allerdings ist es nun so, daß bei nur zwei Aufnahmen sowohl die Kameras \mathbf{P}_i wie auch die rekonstruierten Punkte $\tilde{\mathbf{X}}_j$ nur bis auf eine projektive Transformation bestimmt werden können. Eine metrische oder euklidische Rekonstruktion ist nicht möglich.

Werden nun alle Aufnahmen von derselben Kamera gemacht, so ist \mathbf{K} konstant und es gilt: $\mathbf{P}_i = \mathbf{K}(\mathbf{R}_i|\mathbf{t}_i)$. Die internen Kameraparameter sind für alle Aufnahmen konstant, während die externen Kameraparameter die euklidische Bewegung zwischen den Kameras beschreiben. Glücklicherweise gibt es nun eine geometrische Größe im projektiven 3D Raum, welche unabhängig von den externen Parametern konstant bleibt und daher zur Kalibrierung der Kamera genutzt werden kann. Diese geometrische Größe ist der sogenannte *absolute Kegelschnitt* $\mathbf{\Omega}_\infty$, abgekürzt *AC* (engl. *Absolute Conic*). Ein Kegelschnitt ist eine quadratische Form für Punkte der projektiven Ebene. Sei zum Beispiel \mathbf{C} ein Kegelschnitt, so erfüllen die Punkte $\tilde{\mathbf{x}}$ folgende Gleichung: $\tilde{\mathbf{x}}^T \mathbf{C} \tilde{\mathbf{x}} = 0$. Der absolute Kegelschnitt ist ein Kegelschnitt, welcher in der unendlich fernen Ebene liegt. Das heißt, daß Punkte, welche auf dem Kegelschnitt liegen, die Form $\tilde{\mathbf{X}} = (X, Y, Z, 0)^T = (\mathbf{X}^T, 0)^T$ aufweisen. Der absolute Kegelschnitt ist definiert als: $\mathbf{\Omega}_\infty = \mathbf{I}$ (\mathbf{I} ist die 3×3 Einheitsmatrix.). Die Kegelschnittgleichung lautet dann:

$$\mathbf{X}^T \mathbf{\Omega}_\infty \mathbf{X} = 0 \quad \text{mit} \quad \mathbf{\Omega}_\infty = \mathbf{I} \quad .$$

Punkte $\tilde{\mathbf{X}} = (\mathbf{X}^T, 0)^T$ werden durch eine euklidische Transformation bestehend aus Rotation \mathbf{R}_i und Translation \mathbf{t}_i auf Punkte $\tilde{\mathbf{X}}' = (\mathbf{X}'^T, 0)^T$ abgebildet, für die gilt: $\mathbf{X}' \sim \mathbf{R}_i \mathbf{X}$. Setzt man dies in obige Gleichung ein, so erhält man: $\mathbf{X}'^T \mathbf{\Omega}_\infty \mathbf{X}' = (\mathbf{R}_i^{-1} \mathbf{X}')^T \mathbf{I} (\mathbf{R}_i^{-1} \mathbf{X}') = \mathbf{X}'^T \mathbf{R}_i^{-T} \mathbf{I} \mathbf{R}_i^{-1} \mathbf{X}' = \mathbf{X}'^T \mathbf{X}' = \mathbf{X}'^T \mathbf{\Omega}_\infty \mathbf{X}' = 0$. Man sieht, der absolute Kegelschnitt ist invariant unter euklidischen Transformationen.

Weltpunkte der Form $\tilde{\mathbf{X}} = (\mathbf{X}^T, 0)^T$ werden auf Bildpunkte \tilde{x}_i gemäß $\tilde{x}_i \sim \mathbf{K} \mathbf{R}_i \mathbf{X}$ abgebildet. Da \mathbf{K} und \mathbf{R}_i invertierbar sind gilt: $\mathbf{X} \sim (\mathbf{K} \mathbf{R}_i)^{-1} \tilde{x}_i$. Setzt man dies oben ein, ergibt sich: $((\mathbf{K} \mathbf{R}_i)^{-1} \tilde{x}_i)^T \mathbf{I} (\mathbf{K} \mathbf{R}_i)^{-1} \tilde{x}_i = \tilde{x}_i^T \mathbf{K}^{-T} \mathbf{R}_i^{-1} \mathbf{K}^{-1} \tilde{x}_i = \tilde{x}_i^T \mathbf{\omega} \tilde{x}_i = 0$. Die Matrix $\mathbf{\omega} = \mathbf{K}^{-T} \mathbf{K}^{-1}$ ist das *Bild des absoluten Kegelschnitts*, abgekürzt *IAC* (engl. *Image of the Absolute Conic*). Da $\mathbf{\Omega}_\infty$ invariant bezüglich der Kamerabewegung ist, gilt dies auch für dessen Bild $\mathbf{\omega}$:

$$\mathbf{\omega} = \mathbf{K}^{-T} \mathbf{K}^{-1} = \text{constant} \quad \text{für beliebige Kamerabewegungen}(\mathbf{R}_i, \mathbf{t}_i) \quad .$$

Jede Kamera außer der Ersten \mathbf{P}_i , $i = 2, 3, \dots$ liefert Bedingungen bezüglich des IACs; wie diese Bedingungen im Detail aussehen, soll hier nicht weiter erörtert werden. Wichtig ist nur die Tatsache, daß bereits drei Bilder mit einer identischen Kamera \mathbf{K} aufgenommen, ausreichend zur Bestimmung des IACs und somit zur Kalibrierung sind.

1.5 Numerisches Handwerkszeug

1.5.1 Fehlerquadrat Methode

An bisher zwei Stellen war eine lineares Problem im Sinne der kleinsten Fehlerquadratmethode zu lösen: Zur Berchnung der Homographie basierend auf 2D/2D Punktkorrespondenzen und zur Berechnung der projektiven Kamera durch 3D/2D Punktkorrespondenzen. Hier beschreiben wir jetzt die numerische Lösung für dieses Problem.

Sei folgendes homogene lineare Gleichungssystem gegeben:

$$\mathbf{Ax} = \mathbf{0} \quad ,$$

wobei \mathbf{A} eine $m \times n$ Matrix und \mathbf{x} ein n -dimensionaler Vektor ist. Besitzt die Matrix den Rang r , so hat der Lösungsraum die Dimension $n - r$. Daraus folgt, daß für den Fall, daß der Rang der Dimension des Vektors entspricht $n - r = 0$, die Lösung nur die triviale Lösung $\mathbf{x} = \mathbf{0}$ ist. Eine nicht triviale Lösung gibt es nur für $n > r$. Gibt es mehr Gleichungen m als Unbekannte n und hat die Matrix vollen Rang $r = n$, so ist das System überbestimmt. Dies ist der Fall, wenn mehr Meßpunkte oder Korrespondenzen als Unbekannte vorliegen und die Messungen nicht exakt, also rauschbehaftet sind.

Ist das System überbestimmt, so gibt es keine Lösung im eigentlichen Sinne. Stattdessen wird nach einer Lösung \mathbf{p} gesucht, welche den Fehlervektor $\mathbf{e} = \mathbf{Ap}$ minimiert. Der Fehler \mathbf{e} wird algebraischer Fehler genannt, weil er im Gegensatz zum sogenannten geometrischen Fehler (siehe auch Abschnitt 1.4.3) keiner geometrischen Größe entspricht. Der Abstand des Fehlers zum Ursprung ist seine euklidische Norm $e = \|\mathbf{e}\|$. Die kleinste Fehlerquadratmethode ist formal definiert als:

$$\mathbf{x} = \arg \min_{\mathbf{x}} \|\mathbf{Ax}\| \quad \text{mit} \quad \|\mathbf{x}\| = 1 \quad .$$

Die Normierung mit $\|\mathbf{x}\| = 1$ ist notwendig, um die triviale Lösung auszuschließen. Dieses Optimierungsproblem kann zum Beispiel mit Hilfe der Lagrangeschen Multiplikatormethode gelöst werden. Obiges Minimierungsproblem ist äquivalent zur Minimierung der Lagrangefunktion L :

$$L(\mathbf{x}, \lambda) = \|\mathbf{Ax}\|^2 + \lambda(1 - \|\mathbf{x}\|^2) \quad .$$

Eine notwendige Bedingung für ein Minimum ist:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{x}} &= \mathbf{0} \Leftrightarrow \\ \frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} + \lambda(1 - \mathbf{x}^T \mathbf{x})) &= \mathbf{0} \Leftrightarrow \\ 2(\mathbf{A}^T \mathbf{A}) \mathbf{x} - 2\lambda \mathbf{x} &= \mathbf{0} \Leftrightarrow \\ (\mathbf{A}^T \mathbf{A}) \mathbf{x} &= \lambda \mathbf{x} \quad . \end{aligned}$$

Die letzte Zeile ist die Eigenwert- bzw. Eigenvektorgleichung. Alle Eigenvektoren \mathbf{x} erfüllen die notwendige Bedingung für ein Minimum. Der Wert von $\mathbf{x}_i^T (\mathbf{A}^T \mathbf{A}) \mathbf{x}_i$ für die Eigenvektoren \mathbf{x}_i ist $e_i^2 = \mathbf{x}_i^T (\mathbf{A}^T \mathbf{A}) \mathbf{x}_i = \mathbf{x}_i^T \lambda_i \mathbf{x}_i = \lambda_i$. Das Minimum wird also für den Eigenvektor angenommen, welcher zu dem kleinsten Eigenwert λ_i gehört. Die numerische Berechnung des kleinsten Fehlerquadrates ist folglich ganz einfach durch die Eigenwertberechnung der Matrix $\mathbf{A}^T \mathbf{A}$ gegeben: $\mathbf{A}^T \mathbf{A} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T$, wobei die Spaltenvektoren von \mathbf{V} die Eigenvektoren sind und $\mathbf{\Lambda}$ eine Diagonalmatrix ist, welche die Eigenwerte enthält.

1.5.2 Normalisierung

Der Direct-Linear-Transformation Algorithmus besteht numerisch betrachtet aus der Eigenwertzerlegung einer Matrix, deren Elemente sich aus Koordinaten korrespondierender Punkte zusammensetzen. Nehmen wir als Beispiel einen typischen Bildpunkt mit den Koordinaten (100,100). Da die Einträge in der Matrix \mathbf{A} in Gl. 1.11 quadratisch in den Bildkoordinaten sind, gibt es Einträge der typischen Größenordnung von 1, 10^2 und 10^4 ; in der Matrix $\mathbf{A}^T \mathbf{A}$ reicht die Größenordnung von 1 bis 10^8 . Man sieht, die Größenordnungen der einzelnen Einträge sind sehr verschieden. Dies resultiert in einer schlechten Konditionierung der Matrix $\mathbf{A}^T \mathbf{A}$. Die Kondition ist das Verhältnis von größtem zum kleinsten Eigenwert (Da bei unverrauschten Daten der kleinste Eigenwert Null ist, wird hier nicht der kleinste, sondern der zweitkleinste Eigenwert gewählt.). Bei den oben gewählten typischen Koordinaten kann die Kondition in der Größenordnung von 10^8 liegen. Eine schlechte Konditionierung führt zu ungenauen Ergebnissen in der Eigenwertberechnung. Im Extremfall wird die Rechengenauigkeit der Maschine erreicht, und eine Berechnung ist gar unmöglich. Daher ist es notwendig, die Daten geeignet zu normalisieren, sodaß die Konditionierung verbessert wird. Die Daten werden derart normalisiert, daß der Mittelwert Null ist, und die Standardabweichung $\sqrt{2}$ beträgt (Bei dreidimensionalen Daten ist $\sqrt{3}$ zu wählen.). Die Einträge der Matrix \mathbf{A} werden dann alle die gleiche Größenordnung aufweisen und die Kondition ist oft kleiner als 10. In dem praktischen Teil wird gezeigt werden, daß der DLT Algorithmus einschließlich Normalisierung deutlich bessere Ergebnisse bei verrauschten Daten liefert als der DLT Algorithmus ohne Normalisierung. In

Formeln schreibt sich die Normalisierung wie folgt:

$$\mathbf{x}_{\text{norm}} = \frac{\sqrt{2}}{\sigma}(\mathbf{x} - \boldsymbol{\mu}) \quad \text{mit} \quad \boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad , \quad \sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \boldsymbol{\mu}\|^2} \quad .$$

1.6 Literatur

Das in diesem Bericht beschriebene Kalibrierungsverfahren geht zurück auf Zhang und wurde 1998 bzw. 2000 veröffentlicht [5, 6]. Für Windows lauffähiger Code ist dazu im WWW zu finden [4]. Diese Methode ist von höherem praktischen Wert als das Verfahren nach Tsai [3], welches in der Vergangenheit häufig eingesetzt wurde. Der Direct-Linear-Transformation Algorithmus geht zurück auf Abdel-Aziz und Karara [1]. Ein ausgesprochen gut geschriebenes und verständliches Buch über das Computersehen ist das Buch von Hartley und Zisserman [2]. Dort werden angefangen mit den geometrische Grundlagen, Kameramodelle und Kalibrierung, einschließlich Selbstkalibrierung, bis hin zum Stereosehen alle wichtigen Themen in einer abgerundeten Form präsentiert.

1.7 Aufgaben

Die mit einem Stern * gekennzeichneten Aufgaben sind optional.

1.7.1 Theoretischer Teil

Projektive Geometrie

Aufgabe 1: Seien $\mathbf{m}_{11} = (0, 0)^T$ und $\mathbf{m}_{12} = (1, 1)^T$ zwei Punkte auf der Geraden \mathbf{l}_1 . Berechnen Sie die homogenen Koordinaten von \mathbf{l}_1 . Wie lautet der Normalen- und der Richtungsvektor von \mathbf{l}_1 ? Sei eine weitere Gerade \mathbf{l}_2 durch die Punkte $\mathbf{m}_{21} = (1, 0)^T$ und $\mathbf{m}_{22} = (2, 1)^T$ gegeben. Wie steht die Gerade \mathbf{l}_2 zur Geraden \mathbf{l}_1 ? Berechnen Sie den Schnittpunkt der beiden Geraden. Was ist die geometrische Bedeutung der Lösung?

Aufgabe 2: Wieviele Freiheitsgrade haben die projektive, affine, ähnliche und euklidische Transformation und wieviele Punktkorrespondenzen sind zu ihrer Berechnung notwendig?

Aufgabe 3: Zeigen Sie, daß die affine Transformation Punkte im Unendlichen bzw. Endlichen auf Punkte im Unendlichen respektive Endlichen abbildet.

Kameramodell

Aufgabe 4: Wie äußert sich die Eigenschaft von quadratischen Pixeln in der Kalibriermatrix?

Aufgabe 5: Welche Bedeutung haben die beiden Elemente x_0, y_0 der Kalibriermatrix?

Aufgabe 6: Was modelliert der Scherungsparameter s ?

Aufgabe 7: Was versteht man unter radialer Linsenverzerrung?

Kalibrierung

Aufgabe 8: Wie muß ein Kalibriermuster aufgebaut sein, damit eine Kalibrierung mit nur einer Aufnahme möglich ist?

Aufgabe 9: Ein ebenes Kalibriermuster wird in Richtung der Normalen verschoben. Muß die Länge der Verschiebung bekannt sein?

Aufgabe 10: Wieso sind 3D Kalibrierverfahren unhandlich?

Aufgabe 11: Wieviele Aufnahmen einer Kalibrierebene sind für eine komplette 2D Kalibrierung notwendig?

Aufgabe 12: Welche geometrische Größe wird bei der nicht linearen Optimierung minimiert?

Aufgabe 13: Beschreiben Sie das 2D Kalibrierverfahren.

1.7.2 Praktischer Teil

Verweise auf zu benutzende MATLAB Funktion sind in `typewriter` formatiert.

Aufnahme

Aufgabe 1: Das Kalibrieremuster wird mit Hilfe einer der vorhandenen Kameras zunächst fünfmal aufgenommen. Dazu wird das Skript `grabFrames.m` verwendet, das das Kamera-Live-Bild anzeigt und bei Klick auf dieses Aufnahmen speichert. Die Kamera ist dabei so einzustellen, dass ein möglichst guter Kontrast erreicht wird, bei möglichst geringen Beleuchtungsartefakten um die weitere Verarbeitung zu erleichtern (Tip: Die Einstellung der Bildschärfe ist von großer Bedeutung). Was ist bei der Aufnahme zu beachten?

Labelling

Aufgabe 2: Anhand eines Bildes aus der aufgenommenen Sequenz soll der Vorgang des sogenannten Labellings behandelt werden. Ein Satz von gelabelten Punktkorrespondenzen ist der erste Schritt bei der Kamerakalibrierung.

An erster Stelle steht die Detektion der Kreise. Dazu wird zuerst die gewünschte Region des Kalibrieremusters ausgeschnitten ($R = \text{roipoly}(\text{Image})$). Dann werden die Kantenpunkte der Kreise gefunden ($E = \text{edge}(\text{Image}) \& R$). Damit die Kreise auch wirklich zusammenhängend sind, werden die Kanten dilatiert ($E = \text{imdilate}(E, \text{ones}(3,3))$), und dann die Kreise gefüllt ($E = \text{bfill}(E, 'holes')$). Manchmal kommt es vor, daß beim Ausschneiden Kreise angeschnitten werden. Diese Kreise sollten erkannt und aussortiert werden, da sie keine korrekte Markierung mehr darstellen

($[m,n] = \text{find}(E \& \sim R)$; $E(\text{bwselect}(E,n,m)) = 0$). Schließlich werden alle zusammenhängenden Gebiete durchnumeriert ($L = \text{bwlabel}(E)$), wobei anzumerken ist, daß diese Numerierung quasi willkürlich ist, d.h. die Numerierung noch keinen Rückschluß auf die Position der Markierung zuläßt.

Aufgabe 3: Der eigentliche Labelling Prozeß besteht nun daraus, den Markierungspunkten ihre Weltkoordinaten zuzuordnen. Dieser Vorgang soll automatisch geschehen und ist in der Funktion `labelling` implementiert. Dort werden zuerst die Schwerpunkte der Kreise berechnet. Die linke, obere Markierung wird als Koordinatenursprung gewählt. Die beiden nächstgelegenen (Schwer-) Punkte bestimmen willkürlich die x - und y -Achse des ebenen Koordinatensystems. Ausgehend davon werden sukzessiv alle Punkte in ihren x - und y -Koordinaten durchnumeriert. Der Abstand der Punkte ist dann auf Eins normiert. Soll die tatsächliche metrische Größe als Abstand in die Kalibrierung mit eingehen, so

erreicht man dies durch eine einfache Multiplikation der Koordinaten mit dieser Größe.

Je nach Wahl des Aufnahme-Winkels kann das automatische Labelling fehlschlagen. Um nicht weitere Aufnahmen machen zu müssen, gibt es die Möglichkeit in solchen Fällen ein manuelles Labelling durchzuführen. Dazu dient die Funktion `labellingInteractive`.

Berechnung der Homographien

Aufgabe 4: Programmieren Sie die fehlenden Teile in `computeHomography`; die Vorlage dazu ist in `computeHomography.template.m` zu finden.

Normalisierung

Anhand der Berechnung einer Homographie soll gezeigt werden, wie wichtig die Normalisierung der Daten ist. In der Datei `dataNormalizationData.mat` befindet sich ein Testdatensatz mit 9 Korrespondenzen. Die unverrauschten Punktkoordinaten sind für beide Bilder X_0 . Dazu wurde Rauschen, modelliert durch eine Gaußsche Normalverteilung mit einer Standardabweichung von 5, addiert, so daß sich die Korrespondenzen X_1, X_2 ergeben. Berechnen sie die Homographie mit und ohne Normalisierung (`help computeHomography`). Die beiden Ergebnisse sollen mit der tatsächlichen Homographie, nämlich der homogenen 3×3 Einheitsmatrix, verglichen werden.

Aufgabe 5*: Programmieren Sie die fehlenden Teile in `dataNormalization`; die Vorlage dazu ist in `dataNormalization.template.m` zu finden.

Aufgabe 6*: Berechnen Sie den geometrischen Fehler, welcher der euklidische Abstand zwischen den Punkten und ihren projizierten Korrespondenzen ist: $d_i = \|\mathbf{x}_i - \mathbf{H}^1 \mathbf{x}_i\|$. Der Gesamtfehler ist dann $\sqrt{\frac{1}{n} \sum_{i=1}^n d_i^2}$, welcher auch als Root-Mean-Square (RMS) Fehler bezeichnet wird.

Kameramodell

Aufgabe 7: Die Funktion `camera` bildet Weltpunkte einer Ebene auf Bildpunkte ab. Programmieren sie die Lücken in `camera.template.m`.

Kalibrierung

Aufgabe 8: Das Skript `calibration` führt die komplette Kamerakalibrierung durch. Schauen Sie sich das Skript an und vollziehen die einzelnen Schritte nach.

Literaturverzeichnis

- [1] Y.I. Abdel-Aziz and H.M. Karara. Direct linear transformation into object space coordinates in close-range photogrammetry. In *Proc. Symposium on Close-Range Photogrammetry*, pages 1–18, Urbana, Illinois, Jan 1971.
- [2] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.
- [3] R. 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*, 3(4):323–344, 1987.
- [4] Z. Zhang. Microsoft easy camera calibration tool. Microsoft Research.
- [5] Z. Zhang. A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research, 1998.
- [6] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 22(11):1330–1334, 2000.