

# Kamerakalibrierung

Messen in Videobildern, Leobots-Projekt 2006

Version 1.0

Matthias Jauernig, 03INB, HTWK Leipzig

## Inhaltsverzeichnis

<b>1 Abstract</b>	<b>2</b>
<b>2 Einleitung</b>	<b>2</b>
<b>3 Begriffe</b>	<b>2</b>
<b>4 Kamera-Kalibrierung</b>	<b>3</b>
<b>5 Praktische Überlegungen</b>	<b>4</b>
5.1 Festlegen des Ursprungs des Weltkoordinatensystems . . . . .	4
5.2 Ermitteln der Passpunkte . . . . .	5
<b>6 Die Klasse Calibration</b>	<b>8</b>

---

## 1 Abstract

Dieses kurze Dokument soll eine Einleitung zum Thema Kamerakalibrierung darstellen und eine Benutzungshilfe der Klasse *Calibration* darstellen, die beim Leobots-Projekt 2006 zum Einsatz kommen soll.

## 2 Einleitung

Das Messen von Entfernungen in Videobildern stellt einen elementaren Bestandteil eines kamera-orientierten Roboters dar, welcher die durch Bildverarbeitungs-Algorithmen ermittelten Informationen von Bild- in Weltkoordinaten umrechnen muss, um das weitere Vorgehen strategisch zu planen. Im Laufe der Jahre sind für Zwecke der Messung in Bildern sowie der 3D-Rekonstruktion eine Vielzahl von Verfahren entstanden, welche einen mathematischen Zusammenhang zwischen Welt- und Bildkoordinaten ermitteln können und somit eine wechselseitige Umrechnung (unter der Annahme  $Z_w = 0$ ) ermöglichen.

## 3 Begriffe

**Weltkoordinaten** Weltkoordinaten sind Koordinaten im uns umgebenden dreidimensionalen Raum. Das Koordinatensystem, in welchem sie angegeben werden, muss bei der Kalibrierung definiert werden, d.h. Nullpunkt und die Richtungen der 3 Achsen müssen definiert werden. Es erscheint sinnvoll, das folgende Koordinatensystem zu benutzen: Nullpunkt im Fußpunkt der Kamera, x- und y-Achse in Richtung der x-

bzw.  $y$ -Achse im Bildkoordinatensystem. Die  $z$ -Achse wird hier nicht benötigt, da wir nur in der  $Z=0$ -Ebene messen wollen.

**Bildkoordinaten** Bildkoordinaten sind die Koordinaten eines Punktes im Bild, welches die Kamera bei ihrer Aufnahme liefert. Das dazugehörige Koordinatensystem hat den Ursprung je nach Anwendungsgebiet meist in der linken oberen bzw. der linken unteren Ecke, die  $y$ -Koordinaten stellen die Zeilen, die  $x$ -Koordinaten die Spalten des Bildes dar.

**Kamerakalibrierung** Unter einer vollständigen Kamerakalibrierung versteht man die Bestimmung der Abbildungseigenschaften einer Kamera (*innere Orientierung*) sowie der Orientierung des Weltkoordinatensystems relativ zum Kamerastandpunkt (*äußere Orientierung*). Bei der hier verwendeten teilweisen Kalibrierung errechnet man nur den mathematischen Zusammenhang zwischen Welt- und Bildkoordinaten, da für unsere Zwecke etwas anderes nicht von Interesse ist. Die Parameter der äußeren bzw. inneren Orientierung ließen sich jedoch bei Bedarf jederzeit aus diesem Zusammenhang rück-substituieren.

**Passpunkte** Passpunkte sind Punkte im Weltkoordinatensystem, deren Bildkoordinaten bekannt sind. Mit diesen Punkten wird dann die Kamera-Kalibrierung durchgeführt, um den mathematischen Zusammenhang zwischen Welt- und Bildkoordinaten herzustellen. Die Passpunkte sind beim Erstellen der Kalibrierung zu übergeben, müssen also per Hand gemessen und eingegeben werden. Für das Messen in der  $Z=0$ -Ebene werden mind. 4 Passpunkte benötigt, die aus Genauigkeitsgründen ein möglichst großes Viereck aufspannen sollten.

**Photogrammetrie** Die Photogrammetrie beinhaltet die Rekonstruktion von dreidimensionalen Objekten aus perspektivischen Bildern, z.B. Fotografien. Ein Teil ihrer Algorithmen wird hier verwendet, um das Messen in Bildern zu realisieren.

## 4 Kamera-Kalibrierung

Ziel ist es, einen mathematischen Zusammenhang zwischen Welt- und Bildkoordinaten herzustellen, um die gegenseitige Umrechnung von Punkten zu gewährleisten. Aus der Entwicklung der Photogrammetrie sind eine Vielzahl von Algorithmen entsprungen, die genau realisieren. Die meisten von ihnen erfordern erweiterte Angaben z.B. über interne Abbildungseigenschaften der Kamera und zeichnen sich durch eine erhöhte Komplexität aus.

Hier zum Einsatz kommen soll das Verfahren der *Direkten Linearen Transformation (DLT)*. Die Besonderheit dieses Verfahrens ist, dass man für unsere Zwecke des Messens in der  $Z=0$ -Ebene lediglich 4 Passpunkte benötigt und dass zur Durchführung der Kalibrierung lediglich 4 Passpunkte benötigt werden. Diese Einfachheit hat den Preis,

dass komplexe Eigenschaften wie radiale Linsenverzerrung <sup>1</sup> nicht berücksichtigt werden können. Diese sollen hier nicht weiter berücksichtigt werden und sind ggf. durch empirische Messungen mit individuellen Faktoren auszugleichen (ich denke nicht, dass der Fehler zu groß wird, das muss allerdings in der Praxis getestet werden).

Auf die konkrete Funktionsweise der DLT soll an dieser Stelle nicht eingegangen werden, hier sei beispielsweise auf die exzellenten Bücher von „Kraus“ verwiesen ([KRA97], [KRA96]). Es sei vielleicht soviel gesagt: um einen Welt- auf einen Bildpunkt abbilden zu können, muss zunächst das Weltkoordinatensystem in der Kamerakoordinatensystem transformiert werden. Dies geschieht durch Rotation und Translation des Weltkoordinatensystems. Anschließend wird eine Zentralprojektion durchgeführt, um den korrespondierenden Bildpunkt zu erhalten. Diese Vorgehensweise führt zu 2 Formeln (eine für die x- und eine für die y-Koordinate), durch Substitution der Abbildungsparameter der Kamera erhält man für diese 8 Unbekannte und somit ein Gleichungssystem mit 2 Gleichungen und 8 Unbekannten, welches sich mit 4 Passpunkten z.B. über die QR-Zerlegung lösen lässt.

Man muss den DLT-Algorithmus also mit mind. 4 Passpunkten füttern und erhält als Ergebnis der Lösung des linearen Gleichungssystems wie gewünscht den mathematischen Zusammenhang zwischen Bild- und Weltkoordinaten, mit dem man anschließend beliebige Punkturnrechnungen durchführen kann.

Die eigentlich interessante Frage ist, wie man am besten seine 4 Passpunkte bestimmt - dazu habe ich mir Gedanken gemacht, welche ich im kommenden Abschnitt zusammentrage.

## 5 Praktische Überlegungen

### 5.1 Festlegen des Ursprungs des Weltkoordinatensystems

Um die Passpunkte im Weltkoordinatensystem angeben zu können und eine korrekte Interpretation errechneter Weltpunkte zu besitzen, muss man das Weltkoordinatensystem definieren. Die x- und y-Achsen werden dabei nach den Bildachsen gesetzt, z-Achse wird keine benötigt aufgrund der Messung in der  $Z=0$ -Ebene.

Was noch fehlt, ist der Koordinatenursprung. Diesen könnte man beliebig legen, es bietet sich jedoch an, diesen an den Fußpunkt der Kamera bzw. des Roboters zu setzen. Dies hat den Sinn, dass man dann zur Entfernungsmessung nur noch den Betrag des Vektors zum fragten Weltpunkt errechnen muss.

---

<sup>1</sup>Hierunter versteht man den sogenannten „Fischaugen-Effekt“, der bei stärkeren Weitwinkel-Objektiven auftritt. Dadurch erscheint z.B. eine Gerade, welche sich am linken Rand eines Bildes befindet, mehr als öffnende Klammer (Wölbung nach links).

Es ist zu beachten, dass die in der Klasse *Calibration* umgesetzte Entfernungsmessung die Entfernung eines Punktes vom Punkt  $(0,0)$  misst. D.h. der Punkt an dem man seinen Ursprung platziert, gilt als Ausgangspunkt der Entfernungsmessung. Platziert man ihn z.B. direkt vor dem Roboter, so wird die Entfernung von dort ausgehend gemessen usw.

## 5.2 Ermitteln der Passpunkte

Die 4 Passpunkte, welche zur Ebenen-Kalibrierung mittels DLT benötigt werden, müssen durch Messung im Raum und im Bild ermittelt werden. D.h. i.A. nimmt man 4 Punkte im Raum, welche im Bild zu sehen sein müssen und misst ihre Raumposition bzgl. des festgelegten Koordinatensystems sowie ihre Position im Bild. Die 4 Passpunkte muss man immer dann neu ermitteln, wenn sich die Position der Kamera verändert hat, da dann der Zusammenhang zwischen Welt- und Bildkoordinaten nicht mehr derselbe ist.

Die Prozedur der Rekalibrierung macht natürlich etwas Arbeit, vor allem wenn man die Punkte manuell im Bild vermisst. Zunächst einmal benötigt man sowieso eine Art „Helferprogramm“, welches zumindest gewährleisten muss, dass man die Position eines Pixels im Bild ermitteln kann, welcher mit einem Weltpunkt korrespondieren soll. Sinnvoll, sicher aber auch ein wenig aufwendig wäre es, in das Bild hineinzuzoomen zu können, um auf Subpixel-Ebene zu messen (erhöhte Genauigkeit).

Wie sollten die 4 Passpunkte gewählt werden? Man sollte sie so wählen, dass sie eine möglichst große Ebene aufspannen. D.h. ein Punkt relativ links in der unteren Bildhälfte, einen relativ rechts, einen oben links und einen oben rechts. „Unten“ heißt natürlich relativ nah am Roboter, „oben“ heißt relativ weit entfernt. Auf keinen Fall sollten alle Punkte auf einer Linie liegen, da sonst durch die linearen Abhängigkeiten das Gleichungssystem nicht lösbar ist.

Abbildung 1 soll die Skizze eines von der Kamera aufgenommenen Bildes sein, in welchem 4 mögliche Punkte eingezeichnet sind.

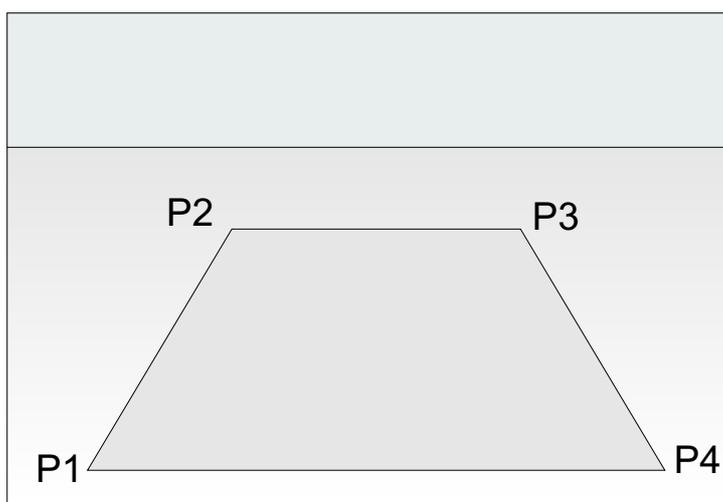


Abbildung 1: Kamerabild mit möglichen Passpunkten

Zur Vereinfachung der Passpunkte-Bestimmung kann man sich zunächst einmal im Weltkoordinatensystem mit 4 Punkten ein Rechteck oder Quadrat aufspannen. Auch in Abbildung 1 wurde dies getan, die Trapezform resultiert im Kamerabild durch die perspektivische Verzerrung. Durch diese Strategie lässt sich schon einmal viel Arbeit sparen.

Weiterhin soll angenommen werden, dass der Ursprung des Weltkoordinatensystems wie besprochen am Fußpunkt der Kamera liegt. Würde man nun die  $y$ -Achse des Weltkoordinatensystems einzeichnen, läge sie in der Mitte eines Kamerabildes und würde mit der  $y$ -Achse des Bildkoordinatensystems übereinstimmen. Diese Annahme führt dazu, dass man nur noch 2 Passpunkte messen muss, z.B. auf der linken Bildhälfte. Die anderen 2 ließen sich durch Spiegelung an der  $y$ -Achse erhalten, was programmtechnisch realisierbar ist. Abbildung 2 verdeutlicht dieses Konzept.

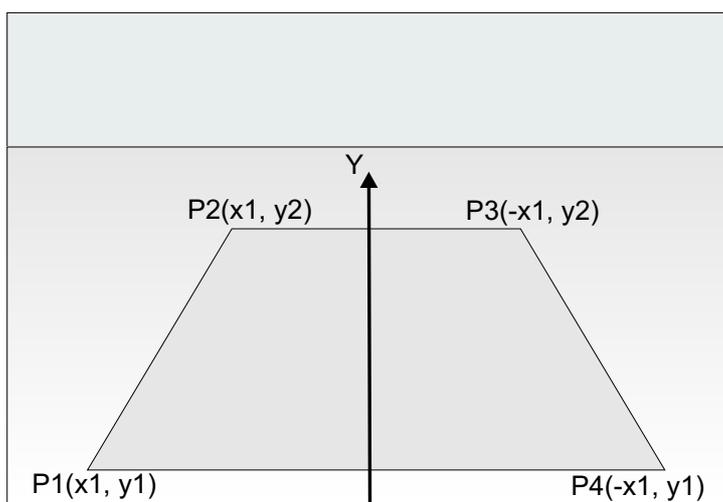


Abbildung 2: Passpunkte durch Spiegelung

Diese Passpunkte wurden unter der Annahme gemessen, dass man ein Rechteck aufgespannt hat. Dies führt dazu, dass man im Weltkoordinatensystem lediglich zwei y-Koordinaten ( $y_1, y_2$ ) und eine x-Koordinate ( $x_1$ ) messen muss. Im Bildkoordinatensystem müsste man noch die Position der Punkte  $P_1$  und  $P_2$  ermitteln und könnte sowohl  $P_3$  als auch  $P_4$  durch Spiegelung an der Bildmitte erhalten.

Bei Benutzung eines aufspannenden Quadrats vereinfacht sich die Messung nochmals, da lediglich eine y- sowie eine x-Koordinate in Weltkoordinaten zu ermitteln sind, woraus die anderen Punkte folgen:

$$\begin{aligned} P_1 &= (x_1, y_1) \\ P_2 &= (x_1, y_1 + 2 * |x_1|) \\ P_3 &= (-x_1, y_1 + 2 * |x_1|) \\ P_4 &= (-x_1, y_1) \end{aligned}$$

Nur in Bildkoordinaten sind wieder 2 Punkte zu ermitteln, nämlich die Projektionen von  $P_1$  und  $P_2$ .

Schön wäre es, wenn man gar keine Weltpunkte mehr messen müsste, sondern diese fest im Programm speichern könnte. Ich schlage folgendes Verfahren vor: man erstellt sich eine Schablone aus Holz- oder Metallstäben, wie sie Abbildung 3 zeigt.

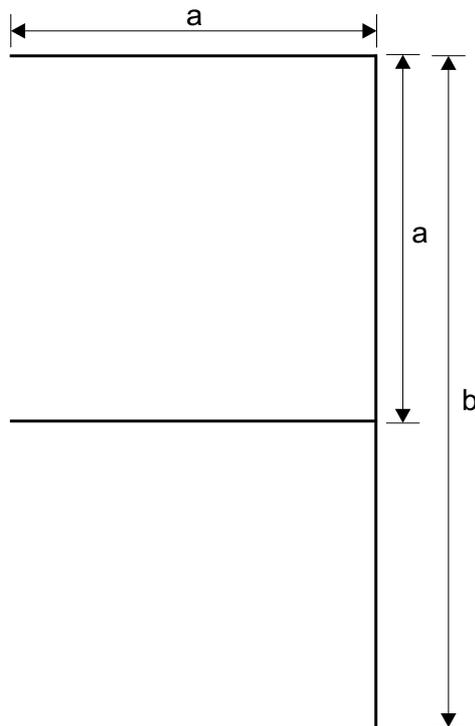


Abbildung 3: Schablone als Hilfe zur Passpunkte-Bestimmung

Der Fußpunkt dieser Schablone wird in den gewünschten Ursprung des Weltkoordinatensystems gelegt, die 4 Weltpunkte ergeben sich dann wie folgt (siehe zur Verdeutlichung

Abbildung 4):

$$P_1 = (-a, b - a)$$

$$P_2 = (-a, b)$$

$$P_3 = (a, b)$$

$$P_4 = (a, b - a)$$

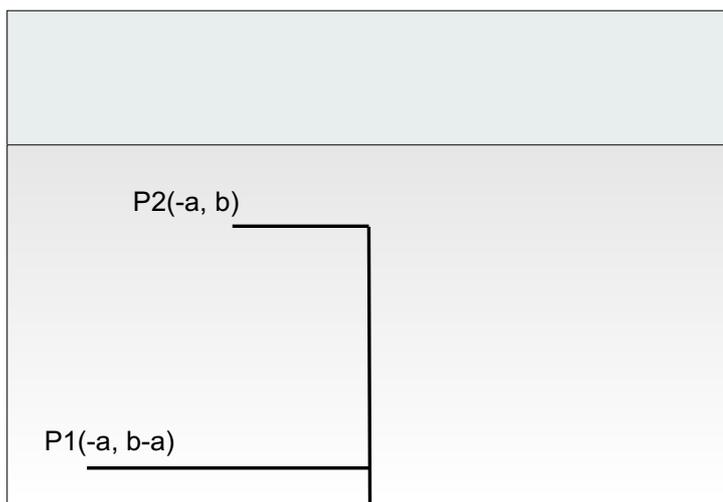


Abbildung 4: Messen mittels Schablone

Hiermit muss man keine Weltpunkte mehr messen, sondern nur noch die Projektionen von  $P_1$  und  $P_2$  im Bild - diese werden dann durch Spiegelung an der vertikalen Bildmitte zu  $P_4$  respektive  $P_3$ .

Zu beachten ist, dass man die Schablone so exakt wie möglich ausrichtet, d.h. dass die Winkel zu den Ausläufern wirklich  $90^\circ$  betragen und dass der Träger exakt in der Bildmitte vertikal nach oben verläuft - dies kann ein adäquates Hilfsmittel darstellen, welches einem mühevollen Arbeit abnimmt.

Gänzlich automatisieren lässt sich die Passpunkte-Bestimmung nicht. Mit der Schablonen-Methode wäre es denkbar, ein Programm zu schreiben, welches die Linien der Schablone erkennt und automatisch die Position der linken Spitzen ermittelt. Somit hätte man eine Passpunkte-Bestimmung, zu welcher man nur noch die Schablone im Bild ausrichten müsste. Dies würde eine sinnvolle Erweiterung eines Kalibrierungs-Tools darstellen, welche realisierbar erscheint.

## 6 Die Klasse Calibration

Diese C++ Klasse ist dafür gedacht, die Kamera-Kalibrierung mittels DLT durchzuführen um anschließend beliebige Umrechnungen von Welt- und Bildkoordinaten zu machen.

Für diesen Zweck wird die Kalibrierung bereits im Klassen-Konstruktor vorgenommen. Wie bereits beschrieben werden mindestens 4 Passpunkte benötigt, um die DLT durchführen zu können:

- **Calibration()**: Liest die Passpunkte vom Benutzer per Tastatur ein.
- **Calibration(PairPoint\*\* points, int count)**: Erstellt die Kalibrierung mittels der Passpunkte, welche sich im PairPoint-Array verbergen. count gibt die Anzahl der Passpunkte in points an, muss also mindestens 4 betragen. PairPoint ist eine Datenstruktur, welche ein Paar aus Welt- und Bildpunkt (also einen Passpunkt) darstellt.
- Mögliche Erweiterungen: Konstruktor, welcher die Passpunkte aus einer Datei liest. Zusätzlich eine Methode Recalibrate(string filename), womit eine Kalibrierung erneut durchgeführt werden kann, z.B. wenn man in der Datei neue Passpunkte abspeichert.

Die Umwandlung von Welt- und Bildkoordinaten findet dann in 2 Methoden statt:

- **Point2D\* ImageToWorldPoint(Point2D\* p)**: Bildet den übergebenen Bildpunkt p auf den zugehörigen Weltpunkt ab und gibt diesen zurück. Die Lage des Weltpunktes im verwendeten Weltkoordinatensystem ist dabei zu berücksichtigen.
- **Point2D\* WorldToImagePoint(Point2D\* p)**: Bildet den übergebenen Weltpunkt p auf den zugehörigen Bildpunkt ab und gibt diesen zurück.

Beim Leobots-Projekt sind vor allem reale Entfernungen vom Roboter zu einem Bildpunkt von Interesse. Dies kann nicht über die Methode ImageToWorldPoint() realisiert werden, da diese nur den Punkt im Koordinatensystem zurückgibt, nicht aber die Entfernung vom Fußpunkt der Kamera. Allerdings kommt die Länge des Vektors (Kamera-Fußpunkt) → (Zum Bildpunkt gehörender Weltpunkt) für die Distanzmessung in Frage. Entspricht der Kamera-Fußpunkt wie vorgeschlagen dem Ursprung des Weltkoordinatensystems, so ist die Entfernung einfach nur der Betrag des Vektors vom Ursprung zum ermittelten Weltpunkt.

Genau dies realisiert die Methode **double ImagePointToDistance(Point2D\* p)**, welche die Entfernung vom Ursprung des Weltkoordinatensystems zum Weltpunkt, der aus dem Bildpunkt p resultiert, zurückgibt.

## Literatur

- [KRA97] Karl Kraus: „Photogrammetrie, Band 1: Grundlagen und Standardverfahren“, 6. Auflage, Dümmler/Bonn 1997
- [KRA96] Karl Kraus: „Photogrammetrie, Band 2: Verfeinerte Methoden und Anwendungen“, 3. Auflage, Dümmler/Bonn 1996