
Untersuchung und Evaluierung verschiedener Methoden der metrischen Distanzmessung

Reimar Stier

28.03.2013

Diese Bachelorarbeit entstand am
Lehrstuhl für Mathematik mit Schwerpunkt Digitale Bildverarbeitung
Prof. Dr. Tomas Sauer
Fakultät für Informatik und Mathematik
Universität Passau

Kurzreferat

In dieser Arbeit werden drei Methoden zur metrischen Distanzmessung vorgestellt und untersucht. Diese Methoden verwenden preisgünstige Hilfsmittel zur Bestimmung von Gebäudehöhen und können bei der Verbesserung der freien Weltkarte (OpenStreetMap) helfen. Als Messgerät wird hierbei entweder ein Laser oder ein Smartphone verwendet. Der Schwerpunkt dieser Arbeit liegt auf einer Methode, die die Gebäudehöhe in einem errechneten 3D-Modell misst. Dieses Modell wird aus mehreren Bildern einer Smartphone-Kamera rekonstruiert.

Stichworte: Laser, Smartphone, OpenStreetMap, GPS, Kamerakalibrierung, 3D-Rekonstruktion, Optimierung

Inhaltsverzeichnis

Kurzreferat	II
Inhalt	III
1 Einleitung	1
1.1 OpenStreetMap	1
1.2 OSM2World	1
1.3 Zielsetzung	1
2 Messmethoden und Szenarien	3
2.1 Gebäudetypen	3
2.2 Umgebung	3
2.3 Laser	4
2.4 Smart-Measure	5
2.5 3D-Rekonstruktion	6
3 Grundlagen der 3D-Rekonstruktion	8
3.1 Lochkamera	8
3.2 Das Lochkameramodell	8
3.2.1 Die Lochkameraabbildung	8
3.2.2 Die Lochkameraabbildung in homogenen Koordinaten	9
3.2.3 Verzeichnung	10
3.3 Kamerakalibrierung	11
3.3.1 Parallele Mustererkennung	11
3.4 Geographische Koordinaten	12
3.4.1 Distanzmessung mit GPS	12
3.4.2 Großkreisdistanz Berechnung	12
3.4.3 Fehlermessung	12
4 3D-Rekonstruktion	14
4.1 Begriffsdefinitionen	14
4.1.1 Lochkamerastereosystem	14
4.1.2 Lochkameramultisystem	14
4.1.3 Markante Punkte	14
4.1.4 Korrespondierende Punkte	14
4.1.5 Koplanare Punkte	14
4.2 Von den Rohdaten zum 3D-Modell	15
4.3 Extraktion und Verknüpfung markanter Punkte	15
4.4 Berechnung der Transformation	15
4.4.1 Berechnung mit erkanntem Objekt	16
4.4.2 Normierung der Translation	16
4.5 Triangulierung	16
4.5.1 Startwertberechnung	17
4.5.2 Startwertberechnung mit der OpenCV-Bibliothek	17
4.5.3 Optimierung der Startwerte	18
4.6 Optimierungsstrategien	18
4.6.1 Optimierungsstrategie A	18
4.6.2 Optimierungsstrategie B	19
4.6.3 Optimierungsstrategie C	21

5	Implementierung	23
5.1	Persistenzmodell	23
5.2	Smartphone-Anwendung	23
5.2.1	Stativ für Kalibrierung und Testbilder	23
5.2.2	Kalibrierung der Smartphone-Kamera	25
5.3	Desktop-Anwendung	26
5.3.1	Extraktion und Verknüpfung markanter Punkte	26
5.3.2	Berechnung der Transformation und Triangulierung	26
5.3.3	Optimierung	27
5.3.4	Messung	27
5.3.5	Quellcode	27
6	Evaluation	28
6.1	Anwendbarkeit	28
6.2	Fehlerquellen, Messhindernisse	28
6.3	Aufwand	29
6.4	Kosten	29
6.5	Messungen	30
6.6	Bewertung der Distanzmessung mit GPS	31
6.7	Bewertung der Methoden	32
7	Fazit	33
7.1	Eintragung von Messungen bei OpenStreetMap	33
7.2	Ausblick	33
A	Literaturverzeichnis	35
B	Abbildungsverzeichnis	36
C	Tabellenverzeichnis	36
D	Glossar	37
E	Eidesstattliche Erklärung	38

1 Einleitung

In der heutigen Zeit sind Informationen omnipräsent. Geodaten können jederzeit von überall abgerufen werden. Mit einem Navigationsgerät, das mit Hilfe eines GPS-Sensors die eigene Position ermittelt, kann man Karten der Umgebung anzeigen lassen. Luftbilder von *GoogleMaps* können bei der Orientierung helfen. Grundsätzlich sind Navigationsgeräte in der Lage, viele brauchbare Informationen darzustellen. Hierbei ist man aber immer auf die Darstellung des Kartenanbieters (GoogleMaps) beschränkt. Im Gegensatz zu GoogleMaps bietet OpenStreetMap (OSM) auch Zugriff auf die Rohdaten (Geodaten) an. Die Rohdaten sind nützliche Informationen aus denen beliebige Kartenansichten erstellt werden können.

1.1 OpenStreetMap

OpenStreetMap [14] ist ein freies Projekt, das der Weiterentwicklung eines riesigen Geodatenbestandes gewidmet ist. Dieses Projekt hat sich zum Ziel gesetzt, Geodaten frei zugänglich zu machen und mit Hilfe einer großen Community mehr Daten zu sammeln und diese fortlaufend zu aktualisieren. Die Community-Teilnehmer, die Daten für OpenStreetMap sammeln, nennen sich *Mapper*. Weitere Informationen über OpenStreetMap und wie man sich am Projekt beteiligen kann, wird in dem Buch „Die freie Weltkarte nutzen und mitgestalten“ [16] beschrieben.

Für unterschiedliche Nutzergruppen oder Anwendungsgebiete gibt es unterschiedliche Kartentypen: Fahrradkarten, Wanderkarten, Autokarten, Stadtkarten, Bus- und Bahnkarten (Nahverkehr) und viele mehr. Alle Kartentypen haben unterschiedliche Anforderungen an die dargestellten Informationen. Fahrradkarten dienen dem Nutzer zur Auffindung geeigneter Radwege - zum Beispiel ruhigere Wege abseits von Kraftfahrtstraßen. Diese Karten haben einen kleineren Maßstab als Autokarten, da kleinere Straßen genutzt werden. Bei Autokarten liegt der Fokus auf Autobahnen oder allgemein dem schnellsten Weg zum Ziel. Aus den Rohdaten von OpenStreetMap können auch 3D-Ansichten erstellt werden. Mit diesem Thema beschäftigt sich Tobias Knerr, welcher OSM2World entwickelt.

1.2 OSM2World

OSM2World ist ein Konverter, der aus OSM-Rohdaten ein 3D-Modell erstellt.[10] Dieses Modell kann in verschiedene Formate exportiert werden: *POVRay* (Persistence of Vision Ray Tracer, ein Darstellungsprogramm für 3D-Szenen), *JOGL* (Java Bindings for OpenGL) und das Bildformat PNG. Auf der Webseite des Projektes befindet sich eine *Slippy Map*¹, welche von OSM2World generierte Bilder verwendet und eine 3D-Ansicht von Deutschland zeigt. Bisher sind in OpenStreetMap nur wenige Gebäude mit Höhen eingetragen, weswegen die meisten Gebäude noch mit einer Standardhöhe dargestellt werden und das 3D-Modell flach wirkt (vgl. Abbildung 1a und 1b). Als einfaches Hilfsmittel wird daher häufig die Anzahl der Stockwerke eines Gebäudes zur Schätzung der Höhe verwendet.

Idealer wäre eine Möglichkeit zur Messung der Höhe. In dieser Arbeit sollen drei Möglichkeiten zur Messung von Gebäudehöhen untersucht werden. Diese Methoden sollen Mappern helfen, Gebäudehöhen zu messen.

1.3 Zielsetzung

Ziel dieser Bachelor Arbeit ist, Methoden der metrischen Messung von Gebäudehöhen zu untersuchen. Diese können verwendet werden, um die freie Weltkarte zu verbessern. Hier sollen Schätzungen der Gebäudehöhen durch Messergebnisse ersetzt werden und bisher nicht erfasste Gebäudehöhen eingetragen werden. Es sollen drei Methoden vorgestellt und untersucht werden. Anschließend sollen diese Methoden nach verschiedenen Kriterien bewertet werden.

¹<http://maps.osm2world.org>



(a) Standardhöhen



(b) Vielgestaltiges Beispiel

Abbildung 1: OSM2World Kartenausschnitte

2 Messmethoden und Szenarien

In diesem Abschnitt werden die drei untersuchten Methoden vorgestellt. Dabei wird auf den jeweiligen Versuchsaufbau eingegangen und die Annahmen erklärt, welche einer Messung zu Grunde liegen. Hierbei sollen bereits mögliche Ursachen für Messfehler ausgemacht werden. Eine Bewertung folgt später. In dieser Arbeit bezeichnet die Gebäudehöhe die Firsthöhe.

2.1 Gebäudetypen

Im Folgenden werden Gebäude anhand ihrer Dachform[11] unterschieden, da diese verschiedene Ansätze beim Messen erfordern. Hierzu werden Gebäude in zwei Kategorien eingeteilt.

Kategorie I

Zu dieser Kategorie gehören alle Gebäude, deren höchster Punkt sich auf einer sichtbaren, frei zugänglichen, senkrechten Gebäudewand befindet. Die Höhe dieser Gebäude ist durch Vermessen einer senkrechten Strecke vom Terrain zum höchsten Punkt des Gebäudes bestimmbar (vgl. Abbildung 2). Bei einem Flachdach wird an einer beliebigen Wand, bei einem Pultdach an der Hohen Wand und bei einem Satteldach an der Giebelwand gemessen.

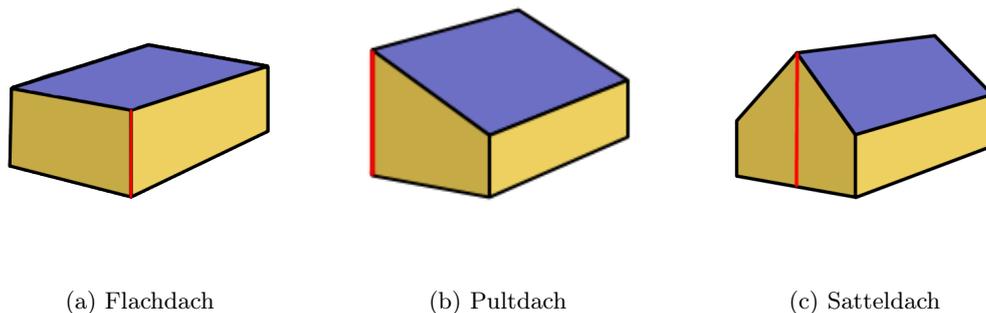


Abbildung 2: Beispiele für Gebäudetypen in Kategorie I

Kategorie II

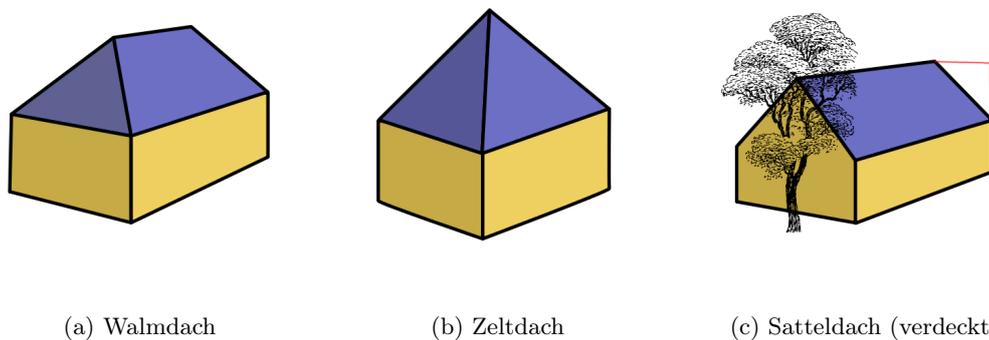
Alle sonstigen Gebäude werden in Kategorie II zusammengefasst (vgl. Abbildung 3). Man kann die Traufhöhe dieser Gebäude mit jeder der drei vorgestellten Methoden messen, jedoch ist es mit der Laser- und der Smart-Measure-Methode (vgl. Abschnitt 2.3 und 2.4) nicht möglich eine Messung der Firsthöhe vorzunehmen.

2.2 Umgebung

Neben der Dachform hat auch die Umgebung des Gebäudes Einfluss auf die Anwendbarkeit der Methoden. Grundlegend für eine Messung, ist die Sichtbarkeit der Gebäudewand und des Firstes. Versperren optische Hindernisse die Sicht, ist eine Messung nicht möglich. An dieser Stelle werden drei Positionierungen des Messgerätes unterschieden.

Ebenerdige Position

In dieser Position ist das Messgerät ebenerdig vor dem Gebäude platziert und es ist möglich mit dem Laser im rechtem Winkel auf die Hauswand zu zielen. Hierbei handelt es sich um die optimalste Position, da alle drei Methoden anwendbar sind.



(a) Walmdach

(b) Zeltdach

(c) Satteldach (verdeckt)

Abbildung 3: Beispiele für Gebäudetypen in Kategorie II

Höhere Position

Bei der höheren Position kann das Messgerät nur oberhalb des Gebäudefußes positioniert werden.

Tiefere Position

In diesem Fall kann das Messgerät nur unterhalb des Gebäudefußes positioniert werden.

2.3 Laser

Die *Lasermessmethode* ermöglicht nur die Messung von Gebäuden der Kategorie I. Es gibt drei Möglichkeiten eine Messung vorzunehmen:

- a) Direkte Distanzmessung:
Bei dieser Möglichkeit wird das Messgerät direkt unter dem Dachüberstand des Gebäudes platziert und es wird die Traufhöhe (vgl. Abbildung 4a) oder die Firsthöhe gemessen. Falls kein Dachüberstand vorhanden ist oder kein Zugang zum Grundstück möglich ist, kann diese Methode nicht angewendet werden. Die direkte Distanzmessung ist sehr genau, da die Abweichung der Lasermessung² typischerweise maximal 1,5 mm beträgt.
- b) Pythagoras-Möglichkeit I:
Die zweite Möglichkeit der Lasermessung wird in Abbildung 4b dargestellt. Für diese Methode platziert man das Messgerät in Laserreichweite vor dem Gebäude und misst im rechten Winkel den Abstand zur Gebäudewand. Anschließend neigt man das Messgerät, zielt auf den Traufpunkt des Gebäudes und misst dessen Abstand zum Traufpunkt. Die Traufhöhe kann schließlich mit Hilfe des Satzes von Pythagoras berechnet werden. Es wird angenommen, dass man einen rechten Winkel an der Gebäudewand gemessen hat und die Gebäudewand im Lot steht.
- c) Pythagoras-Möglichkeit II:
Für die dritte Möglichkeit werden drei Lasermessungen erstellt (vgl. Abbildung 4c). Eine Messung im rechten Winkel auf die Gebäudewand, eine Messung zum Traufpunkt oder zum First und eine Messung zum Fuß der Gebäudewand. Diese Möglichkeit erlaubt eine Anwendung der Laser-Methode bei einer erhöhten Position.

Alle drei Möglichkeiten setzen voraus, dass die Gebäudeoberfläche eine Lasermessung zulässt. Problematisch sind beispielsweise Glasfassaden, da diese den Laser reflektieren und die Messung

²http://www.bosch-pt.com/productspecials/professional/dle50/de/de/start/index.htm?WT.tsrc=Social_Bosch

verfälschen. Weiterhin sind hell gestrichene Wände bei starker Sonneneinstrahlung ein Messhindernis, da der Laserpunkt nicht mehr sichtbar ist. Außerdem ist mit einer Messung der Traufhöhe, außer bei einem Flachdach, noch nicht die Gebäudehöhe bestimmt.

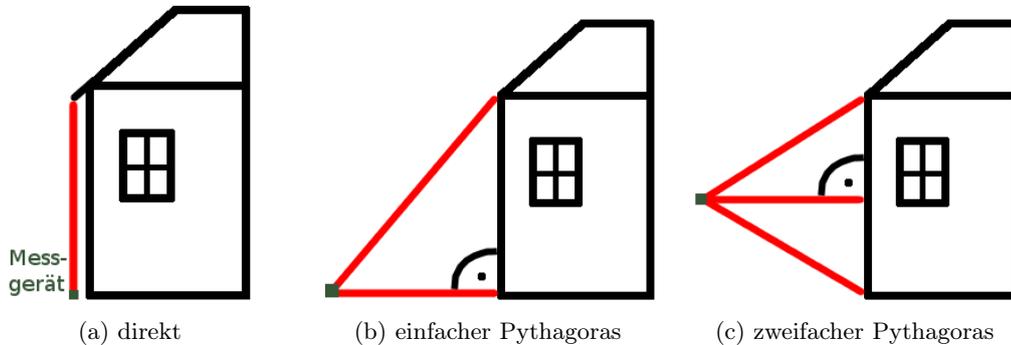


Abbildung 4: Lasermessung der Traufhöhe

Für die in dieser Arbeit enthaltenen Messergebnisse wurde das Lasermessgerät Bosch DLE 40 verwendet. Dieses hat eine Messreichweite von 40 Metern und keine “Pythagoras”-Funktion, im Gegensatz zum Bosch DLE 50, welches das Ergebnis der Messung mit Möglichkeit b sofort ausrechnet. Teurere Geräte haben zusätzlich eine Wasserwaage eingebaut, mit der eine genauere Messung denkbar wäre, da es dadurch einfacher ist im rechten Winkel zu messen.

2.4 Smart-Measure

Smart-Measure [2] ist eine Smartphone-Anwendung zur Entfernungsmessung. Mit der Anwendung ist es möglich, die Distanz zu einem Objekt sowie dessen Höhe und Breite zu messen. Im Folgenden soll nur die Funktion zur Höhenmessung betrachtet werden. Bevor man mit Smart-Measure eine Messung vornimmt, sollte die Haltehöhe h festgelegt werden. Erfolgt diese Angabe nicht, wird standardmäßig eine Höhe von $h = 1,5$ m angenommen. Außerdem ist eine Justierung des Lagesensors möglich, der zur Messung des Winkels verwendet wird. Hierzu wird das Smartphone an eine senkrechte Wand gehalten und die Anwendung passt den gemessenen Winkel auf einen rechten Winkel an. Dieser Vorgang wird „Kalibrierung“ der Anwendung genannt. Da diese Werte großen Einfluss auf die Genauigkeit der Messung haben, sollten sie vor der Messung erfolgen. Die Messung erfolgt schließlich in zwei Schritten:

1. Im ersten Schritt zielt man mit dem Fadenkreuz der Anwendung auf den Fuß der Mauer (siehe Abbildung 5a). Nun misst die Anwendung mit Hilfe des Lagesensors den Neigungswinkel α des Smartphones. Aus der bekannten Höhe und dem Winkel kann man die Entfernung zum Gebäude wie folgt berechnen:

$$\overline{AB} = \tan(\alpha) \cdot h$$

2. Im zweiten Schritt zielt man auf den Traufpunkt des Gebäudes (siehe Abbildung 5b). Die Applikation misst nun erneut den Neigungswinkel: β . Die Traufhöhe ergibt sich schliesslich wie folgt:

$$t_{\text{height}} = h + \tan(\beta) \cdot \overline{AB}$$

Eine Anwendung der Smart-Measure-Methode bei erhöhter Position ist mit der kostenpflichtigen Version möglich. Da eine Messung für ein solches Szenario von der Anwendung nicht vorgesehen ist, erfolgt die Messung mit zwei Methoden. Zunächst wird mit der einen Methode die Höhe vom Fuß der Gebäudewand bis auf die Haltehöhe gemessen (rechter Winkel zum Gebäude). Schließlich wird mit der oben beschriebenen Methode zur Gebäudehöhenmessung die verbleibende Höhe des Gebäudes gemessen. Diese Methode wurde nicht weiter untersucht.

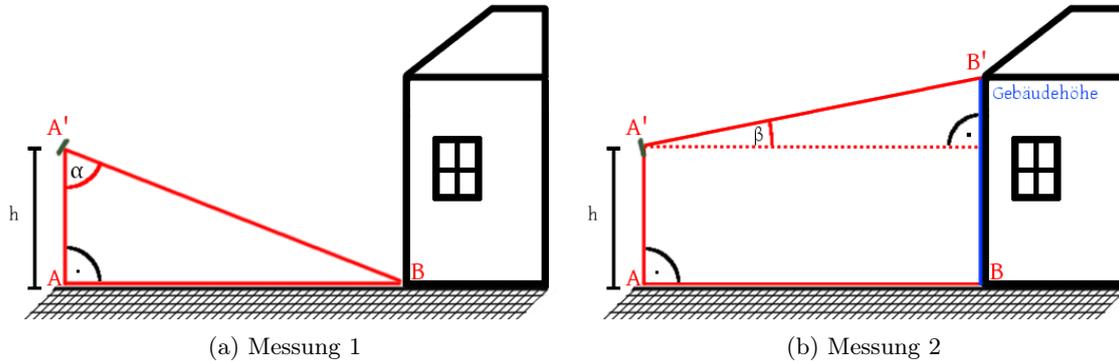


Abbildung 5: Smart-Measure

2.5 3D-Rekonstruktion

Die 3D-Rekonstruktion ist die aufwändigste der drei untersuchten Methoden. Hierbei erfolgt die Messung nicht vor Ort, sondern zu einem späteren Zeitpunkt mit einer Desktop-Anwendung. Das bedeutet, dass die Rekonstruktion in zwei Schritten erfolgt.

Sammeln der Messdaten

Zunächst werden Messdaten mit Hilfe einer selbst geschriebenen Smartphone-Anwendung gesammelt. Mit der Kamera des Smartphones werden Bilder von dem Gebäude erstellt. Es werden mindestens zwei Bilder benötigt. Hierbei ist zu beachten, dass das Gebäude in kompletter Höhe in allen Bildern zu erkennen ist. Jedes Bild wird versetzt zu den vorigen Bildern aufgenommen (vgl. Abbildung 6).

Gleichzeitig werden die Aufnahmeorte mit dem GPS-Sensor erfasst, um die Abstände zwischen den Kamerapositionen bestimmen zu können. Außerdem werden die Abstände zusätzlich mit einem Meterstab vermessen, als Referenz zu den GPS-Daten. Da Smartphones so konzipiert sind, dass sich das Smartphone beim Fotografieren auf Augenhöhe des Anwenders befindet, ist eine exakte Abstandsmessung mit dem Meterstab nicht möglich, da man hierzu das Smartphone direkt am Boden positionieren müsste. Als zusätzliches Hilfsmittel kann ein Schachbrett mit bekannten Maßen verwendet werden, um die Rotation und Translation der Kamerapositionen leichter berechnen zu können.



Abbildung 6: Bildsequenz der Messdaten

Messung

Die Messung erfolgt schließlich in einem errechneten 3D-Modell der fotografierten Szene. Hierzu werden zwei Punkte an der Hauswand ausgewählt: einer am Fuß der Gebäudewand und der andere am Traufpunkt (Punkte A und B in Abbildung 7). Anschließend wird noch ein dritter Punkt C ausgewählt, der den höchsten Punkt des Gebäudes markiert - die Firsthöhe. Dieser wird auf eine

Gerade durch die Punkte A und B projiziert. Der projizierte Punkt ist C'. Die Höhe des Gebäudes kann im Modell anhand der Strecke von Punkt A zu Punkt C' gemessen werden.

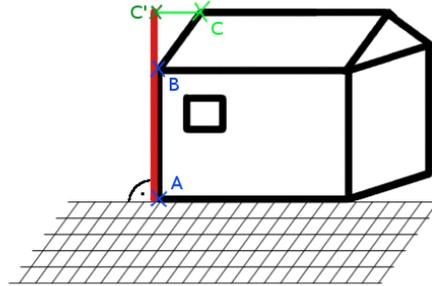


Abbildung 7: Messung der Gebäudehöhe im 3D Modell

Zur ausführlicheren Untersuchung dieser Methode werden im folgenden Kapitel die Grundlagen der 3D-Rekonstruktion eingeführt.

3 Grundlagen der 3D-Rekonstruktion

Zur Berechnung eines 3D-Modells benötigt man ein mathematisches Modell der Kamera, das die Abbildung dreidimensionaler Weltpunkte auf den Sensor beschreibt. Im Folgenden sollen die Grundlagen der Modellierung einer Kamera anhand eines Lochkameramodells vorgestellt werden.

3.1 Lochkamera

Eine Lochkamera ist ein einfaches optisches Gerät, in das durch eine kleine, verschließbare Öffnung Licht einfallen kann. Zum Beispiel ist in Abbildung 8 die Lochkamera in Form einer schwarzen Box abgebildet, in welche das Licht einfällt und auf dessen Rückseite ein seitenverkehrtes, auf dem Kopf stehendes, reelles Bild entsteht. Dieses Bild lässt sich mit Hilfe eines Bildsensors festhalten. Da sich bei der Lochkamera alle Lichtstrahlen in einem Punkt schneiden, kann das selbe Bild auch vor der Lochkamera beobachtet werden. Dieses Bild ist aufrecht und die Ebene, in der dieses beobachtet wird, wird $\{z = 1\}$ -Ebene genannt.

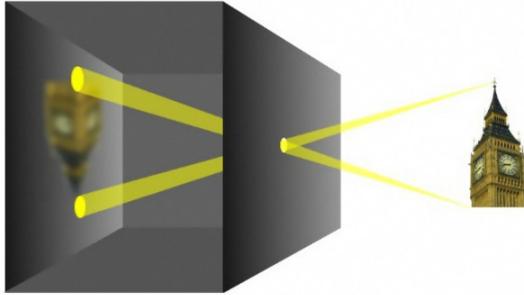


Abbildung 8: Lochkamera

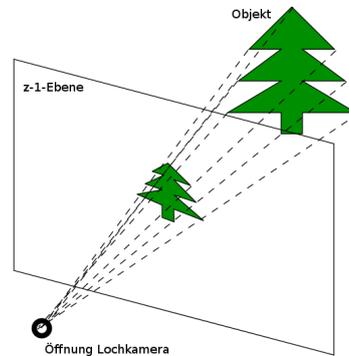


Abbildung 9: $\{z = 1\}$ -Ebene³

Da man mit einer Kamera Objekte in unterschiedlicher Entfernung scharf fotografieren möchte, besitzen die meisten Kameraobjektive zur Veränderung der Brennweite mehrere Feststofflinsen, oder eine Flüssiglins. Bei Flüssiglinsen ist es möglich durch das Anlegen einer Spannung die Brennweite der Linse zu variieren. Flüssiglinsen sind besonders widerstandsfähig und von kleiner Bauform, weshalb sie häufig in Smartphones verbaut werden. Im Modell werden nur Kameraobjektive mit fester Brennweite betrachtet.

3.2 Das Lochkameramodell

Das folgende Lochkameramodell ist entnommen aus “Rechnersehen” [6] von Dr. habil. Tobias Hanning, jedoch wird die Notation der OpenCV-Bibliothek [4] verwendet. Das Lochkameramodell erhält man, wenn man nur die Abbildungseigenschaften des Mittelpunktstrahls betrachtet und die optischen Eigenschaften des Objektivs durch die Eigenschaften einer einzelnen Feststofflinse approximiert. [7]

3.2.1 Die Lochkameraabbildung

Mehrere Bilder einer Szene stellen mehrere Perspektiven einer Kamera auf die Szene dar. Da für das 3D-Modell ein Koordinatensystem benötigt wird, wählt man hierfür das Koordinatensystem der ersten Perspektive. Koordinatensysteme weiterer Perspektiven werden bezüglich des Referenzkoordinatensystems der ersten Perspektive betrachtet. In diesem Sinne definiert man die Lochkameraabbildung K als eine Komposition dreier Abbildungen:

$$K = P \circ P_z \circ T$$

³Quelle: „Kamerakalibrierung und 3D-Rekonstruktion“[7]

Die Abbildung T stellt eine Transformation eines Punktes $p \in \mathbb{R}^3$ vom Referenzkoordinatensystem in das Kamerakoordinatensystem der aktuellen Kamera dar. Diese Transformation setzt sich aus einer Rotation $R \in \mathbb{R}^{3 \times 3}$ und einer Translation $t \in \mathbb{R}^3$ zusammen:

$$T : \mathbb{R}^3 \rightarrow \mathbb{R}^3, p \mapsto R p + t$$

Bei P_z handelt es sich um eine Projektion auf die $\{z = 1\}$ -Ebene:

$$P_z : \mathbb{R}^3 \setminus \{z = 0\} \rightarrow \{z = 1\}, (x, y, z) \mapsto \left(\frac{x}{z}, \frac{y}{z}, 1\right)$$

Das Urbild von P_z ist eine Gerade, welche *Sichtstrahl* genannt wird. Der Punkt $(0, 0, 1)^T$ bezüglich des Kamerakoordinatensystems heißt *Hauptpunkt*. Dieser ist der Schnittpunkt der optischen Achse mit der $\{z = 1\}$ -Ebene.

Die Abbildung P ist eine Koordinatentransformation von der $\{z = 1\}$ -Ebene in das Bildkoordinatensystem. Seien $f_x, f_y \in \mathbb{R}_+^*$ und $\gamma, c_x, c_y \in \mathbb{R}$, dann stellt P eine Scherung in zwei Richtungen und eine Translation um c dar:

$$P : \{z = 1\} \rightarrow \mathbb{R}^2, \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f_x & \gamma \\ 0 & f_y \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \end{pmatrix}$$

Die *intrinsischen Parameter* beschreiben in pixel-bezogenen Einheiten die Brennweite (f_x, f_y) und den Hauptpunkt (c_x, c_y) der Kamera. Der Hauptpunkt befindet sich typischerweise im Mittelpunkt des Bildes, jedoch kann es hierbei in Abhängigkeit vom Herstellungsprozess der Kamera zu Abweichungen kommen. Mit einem γ -Wert ungleich Null kann eine Scherung zwischen den Koordinatenachsen des Bildkoordinatensystems beschrieben werden.

3.2.2 Die Lochkameraabbildung in homogenen Koordinaten

Zur effizienten Berechnung der Abbildungen verwendet man homogene Koordinaten, welche durch Erweiterung um eine Dimension konstruiert werden. Diese ermöglichen die Darstellung der Abbildungen als reine Matrizenmultiplikation. Ein Weltpunkt $p \in \mathbb{R}^3$ wird dabei auf einen Punkt $p' \in \mathbb{R}^4$ abgebildet:

$$h : \mathbb{R}^3 \rightarrow \mathbb{R}^4, \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Seien f_x, f_y, c_x, c_y und γ wie in Abschnitt 3.2.1, so definiert man:

$$\tilde{P} := \begin{pmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{Projektionsmatrix})$$

$$\tilde{T} := (R|t) \in \mathbb{R}^{3 \times 4} \quad (\text{Transformationmatrix})$$

$$\tilde{K} := \tilde{P}\tilde{T} \quad (\text{Kameraabbildung})$$

Für alle $p = (x, y, z)^T \in \mathbb{R}^3$ gilt:

$$K \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} \Leftrightarrow \exists s \in \mathbb{R} \setminus \{0\} : \tilde{K} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

Mit der Umkehrung der Projektionsmatrix \tilde{P} können normalisierte Koordinaten in der $\{z = 1\}$ -Ebene berechnet werden. Sei $(u, v, 1)^T$ eine homogene Bildkoordinate, so gilt:

$$\tilde{P}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

3.2.3 Verzeichnung

Als einzige Art der Abbildungsfehler optischer Geräte, wird *Verzeichnung* betrachtet, da diese den Mittelpunktstrahl betreffen. In OpenCV werden zwei Arten von *Verzeichnung* betrachtet: *Radiale Verzeichnung* und *tangentiale Verzeichnung*. Wenn man tangentiale Verzeichnung modelliert, setzt man in der Abbildung P typischerweise den Wert $\gamma = 0$. Die Einführung von Verzeichnung in das Kameramodell verhindert die Formulierung als projektive Abbildung.[7] Das bisherige Modell wird erweitert um die Verzeichnungsparameter der OpenCV-Bibliothek:

T	$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t$
P_z	$x' = x/z$ $y' = y/z$
Verzeichnung	k_1, k_2, k_3 sind radiale Verzeichnungsparameter p_1, p_2 sind tangentiale Verzeichnungsparameter $x'' = x'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x' y' + p_2 (r^2 + 2x'^2)$ $y'' = y'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_2 x' y' + p_1 (r^2 + 2y'^2)$ mit $r^2 = x'^2 + y'^2$
P	$u = f_x \cdot x'' + c_x$ $v = f_y \cdot y'' + c_y$

Tabelle 1: Verzeichnungsmodellierung der OpenCV-Bibliothek

Die Verzeichnungsparameter $(k_1, k_2, k_3, p_1, p_2)$ sind unabhängig von der betrachteten Szene, weshalb sie ebenfalls zu den intrinsischen Parametern gehören. Die homogenisierte Kameraabbildung verliert hierdurch aber nicht an Bedeutung, da sie für die Startwertberechnung wichtig ist.

3.3 Kamerakalibrierung

Der Vorgang zur Ermittlung der intrinsischen Kameraparameter wird *Kamerakalibrierung* genannt. Bei günstigen Kameras, wie sie in Smartphones verbaut werden, kann nicht angenommen werden, dass jede gefertigte Linse in dem verwendeten Kameratyp gleich sein muss. Daher ist eine Kalibrierung sinnvoll. Hierzu werden Kalibriermuster verwendet.

Kalibriermuster sind Objekte deren genaue Maße bekannt sind. Gut geeignet ist zum Beispiel ein Schachbrett mit abwechselnd schwarz-weißen Quadraten und bekannten Seitenlängen. Ein Schachbrett mit n Quadraten in x-Richtung und m Quadraten in y-Richtung, wobei $n \neq m$ gilt, nennt man ein $(n \times m)$ asymmetrisches Schachbrett. Gilt hingegen $n = m$ spricht man von einem symmetrischen Schachbrett. Für ein solches Schachbrett sagt man, dass es $(n - 1) \cdot (m - 1)$ Ecken hat. Als Ecken bezeichnet man die Punkte des Schachbretts, in denen sich jeweils vier Quadrate berühren (vgl. grüne Markierungen in Abbildung 10). Es sind auch andere Kalibriermuster wie zum Beispiel Kreismuster möglich. Grundsätzlich lässt sich die Ausrichtung asymmetrischer Muster leichter erkennen als die symmetrischer Muster.

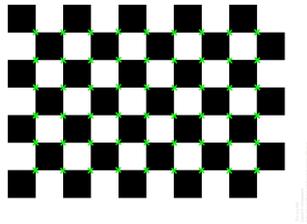


Abbildung 10: Schachbrett

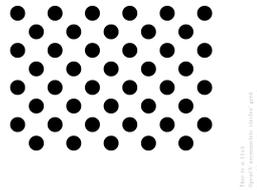


Abbildung 11: Kreismuster

Die Kalibrierung mit Hilfe der OpenCV-Bibliothek erfolgt in vier Schritten⁴:

1. In jedem Bild wird eine Mustererkennung durchgeführt und im Fall eines erkannten Schachbrettmusters eine Methode zur Verfeinerung der gefundenen Ecken aufgerufen. Im Anschluss sind die Bildkoordinaten (u, v) aller Ecken bekannt.
2. Im zweiten Schritt werden Objektkoordinaten für die Ecken des Musters erstellt. Im Falle eines $n \times m$ Kalibrierusters erhält die linke obere Ecke die Koordinate $(0, 0, 0)$ und die rechte untere Ecke $(n - 1, m - 1, 0)$.
3. Alle erkannten Muster werden markiert und die Ausrichtung der Muster wird überprüft.
4. Schließlich berechnet OpenCV mit den Bild- und Objektkoordinaten der Muster die intrinsischen Kameraparameter.

3.3.1 Parallele Mustererkennung

Da Muster in mehreren Bildern unabhängig voneinander erkannt werden können, lässt sich dieser Teil des Programmes parallelisieren. Zur Messung des parallelen SpeedUps wurde ein Quad-Core-Prozessor verwendet. Der parallele Speedup S_p ist der Quotient der sequentiellen Ausführungszeit T_1 und der parallelen Ausführungszeit T_p , wobei p die Anzahl der Prozessorkerne bezeichnet:

$$S_p := \frac{T_1}{T_p} \quad (\text{paralleler Speedup})$$

In Tabelle 2 ist ersichtlich, dass bei vier Kernen zweifacher Speedup erreicht wird. Für die Kamerakalibrierung auf dem Smartphone ist dies bereits ausreichend und die Ausführungszeit beträgt weniger als fünf Minuten (vgl. 5.2.2).

⁴ http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

Anzahl Bilder	T_1 (ms)	T_4 (ms)	S_4
2	2527	1373,5	1,83
3	3915,75	1577	2,48
4	5194,25	1638,75	3,16
5	6410,5	2530,5	2,53
6	8098	3126,25	2,59
10	22530,25	12489	1,80

Tabelle 2: Speedup

3.4 Geographische Koordinaten

Zur Bestimmung der Position auf der Erde werden geographische Koordinaten verwendet. Bei diesen Koordinaten handelt es sich um gedachte Kugelkoordinaten. Die Erde wurde in Längengrade und Breitengrade eingeteilt: Die Breitengrade (Latitude) beschreiben die Nord-Süd-Position und die Längengrade (Longitude) beschreiben die Ost-West-Position. Es gibt 360 Längengrade parallel zum Null-Meridian. Der Null-Meridian ist der Längengrad der vom Nordpol zum Südpol durch Greenwich (England) verläuft. Für Positionen westlich des Null-Meridians verwendet man negative Koordinaten (-180 bis 0) und für Positionen östlich verwendet man positive Koordinaten (0 bis 180). Breitengrade gibt es 180. Diese werden vom Äquator aus nach Norden bis 90 hochgezählt. Auf der Südhalbkugel sind die Breitengrade negativ.

3.4.1 Distanzmessung mit GPS

Diese Koordinaten kann man mit dem Global Positioning System (kurz GPS) messen. GPS ist ein globales satellitenbasiertes Navigationssystem, dessen Daten man mit einem GPS-Empfänger (GPS-Sensor) auswerten kann.

3.4.2 Großkreisdistanz Berechnung

Ein Großkreis beschreibt einen größtmöglichen Kreis auf einer Kugeloberfläche. Ein Orthodrome beschreibt die kürzeste Distanz zwischen zwei Punkten auf dem Großkreis. Mit der *Formel von Haversine* kann diese Distanz berechnet werden. Dabei wird für die Erde eine idealisierte Kugelform mit einem Umfang von 40000 km und einem Radius von 6367 km angenommen. Da diese leicht zu berechnen ist, und für kurze Distanzen wenig von der tatsächlichen Entfernung abweicht, wurde sie bei der Fehlermessung eingesetzt. Ein genaueres Ergebnis liefert das Referenzsystem des *World Geodetic System 1984* (kurz WGS84), das ein Bezugssystem zur Berechnung beschreibt.

3.4.3 Fehlermessung

Für die erste Messung wurde eine GPS-Maus (GPS-Sensor mit USB-Anschluss, Modell: Navilock NL-302U) verwendet. Die GPS-Maus wurde dabei nicht bewegt und es wurde alle fünf Sekunden die gemessene GPS-Position gespeichert. Anforderung war hier, dass mindestens fünf Satelliten für die Berechnung verwendet werden können. Dabei wurden circa 10000 qualitative Messwerte (ausreichend Satelliten) im Verlauf von 14 Stunden gesammelt. Im Schaubild sieht man die Verteilung dieser Messwerte innerhalb von 14 Stunden. Das Schaubild ist zentriert um den Mittelwert der Messwerte. Der durchschnittliche Abstand zum Mittelwert betrug 6,479 m. Der maximale Abstand zum Mittelwert betrug 33,787 m.

Für die zweite Messung wurde der GPS-Sensor eines Samsung "Galaxy Nexus"-Smartphones verwendet. Es wurden 30 Stunden alle zehn Sekunden gemessen und 8074 Messungen mit einer Abweichung von 10 m oder weniger gesammelt. Der durchschnittliche Abstand zum Mittelwert betrug 9,951 m. Der maximale Abstand zum Mittelwert betrug 51,273 m.

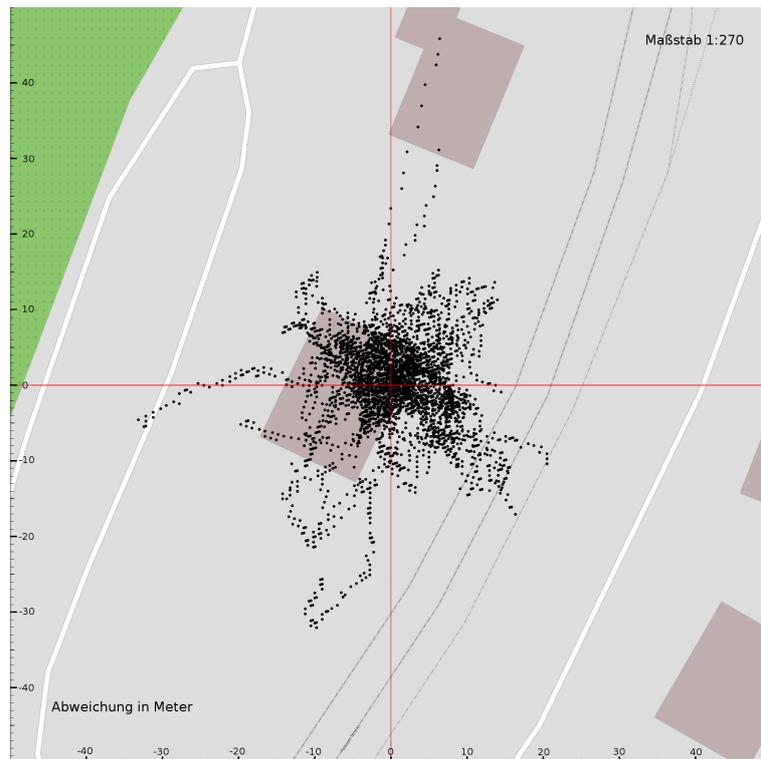


Abbildung 12: GPS-Maus

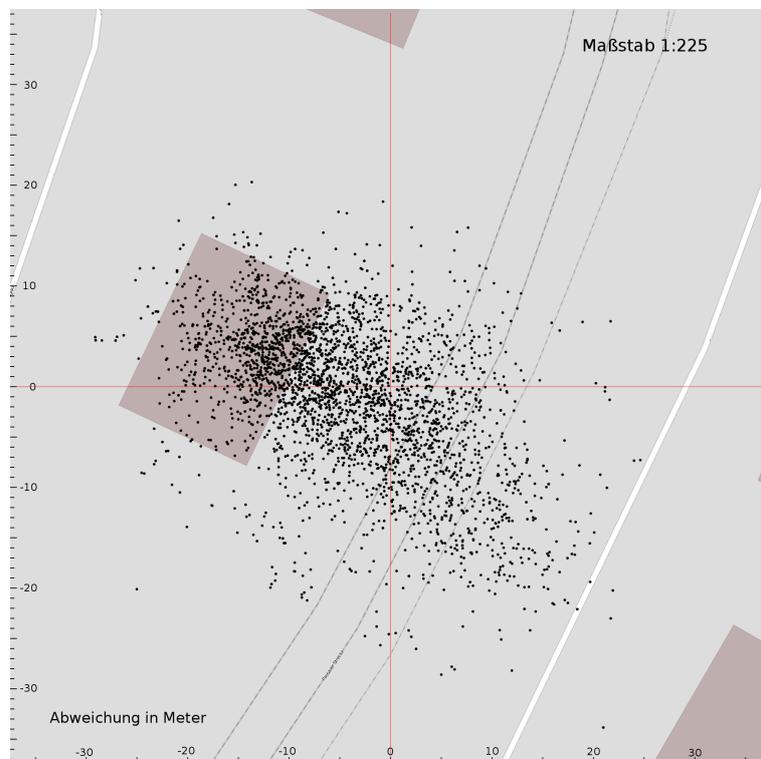


Abbildung 13: Smartphone-GPS-Sensor

4 3D-Rekonstruktion

Die Berechnung eines 3D-Modells aus zwei Bildern nennt man Stereorekonstruktion. Im allgemeineren Fall von mehr als zwei Bildern spricht man einfach von 3D-Rekonstruktion. Häufig beginnt man mit der Stereorekonstruktion, bevor man weitere Bilder derselben Szene betrachtet. Im Spezialfall einer einzelnen Kamera, mit der sequentiell Bilder einer statischen Szene aufgenommen werden, kann man in allen Kameraabbildungen dieselben intrinsischen Parameter verwenden.

4.1 Begriffsdefinitionen

Zunächst werden einige Begriffe für die Beschreibung der Berechnung des Modelles eingeführt.

4.1.1 Lochkamerastereosystem

Seien K_1 und K_2 zwei Lochkameraabbildungen mit $K_1 = P \circ P_z \circ T_1$ und $K_2 = P \circ P_z \circ T_2$, wobei $T_1 = Id$ und $T_2 \neq Id$ (Id steht für die Identische Abbildung). Dann heißt (K_1, K_2) ein *Lochkamerastereosystem*. [7]

4.1.2 Lochkameramultisystem

Das Lochkamerastereosystem kann nun auf mehr als zwei Kamerapositionen erweitert werden. Sei $n \in \mathbb{N}, n > 2$ und seien K_1, K_2, \dots, K_n Lochkameraabbildungen, für die gilt:

$$K_i = P \circ P_z \circ T_i \text{ mit } T_1 = Id, T_i \neq Id, 2 \leq i \leq n$$

In diesem Fall wird (K_1, K_2, \dots, K_n) ein *Lochkameramultisystem* genannt. Als Ursprung für das Koordinatensystem wird o.B.d.A. die erste Kamera gewählt (vgl. Abbildung 14).

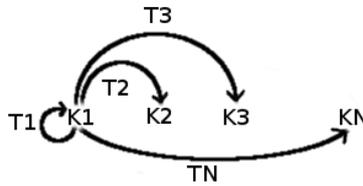


Abbildung 14: Kameratransformationen

4.1.3 Markante Punkte

Bildkoordinaten mit hohem Wiedererkennungswert nennt man *markante Punkte*. Diese werden oft auch einfach „Features“ (engl. Merkmale) genannt.

4.1.4 Korrespondierende Punkte

Markante Punkte in einem oder mehreren Bildern, die den gleichen Weltpunkt (3D-Punkt) beschreiben, nennt man *korrespondierende Punkte*. Die Bildkoordinaten eines Fensterecks in zwei Bildern sind ein Beispiel für *korrespondierende Punkte*.

4.1.5 Koplanare Punkte

Eine Menge von 3D-Punkten nennt man *koplanar*, wenn diese in der selben Ebene liegen. Dies trifft zum Beispiel auf alle Ecken des Schachbrett-Kalibrierungsmusters zu. Seien die Punkte $p_1 = (x_1, y_1, z_1)$, $p_2 = (x_2, y_2, z_2)$ und $p_3 = (x_3, y_3, z_3)$ paarweise verschieden und es gelte:

$$(x_1 = x_2 = x_3) \vee (y_1 = y_2 = y_3) \vee (z_1 = z_2 = z_3)$$

In diesem Fall sind die drei Punkte p_1, p_2 und p_3 koplanar.

4.2 Von den Rohdaten zum 3D-Modell

Um aus den Bildrohdaten einer kalibrierten Kamera ein 3D-Modell zu erstellen sind folgende Schritte notwendig:

1. Extraktion markanter Punkte
2. Verknüpfung markanter Punkte
3. Berechnung der Transformation
4. Triangulierung
5. Optimierung des Modells

4.3 Extraktion und Verknüpfung markanter Punkte

Ein *Feature-Detektor* ist eine implementierte Methode zur Extraktion markanter Punkte. Markante Punkte können ebenfalls automatisch verknüpft werden. Diese Methoden heißen *Feature-Matcher*. Folgende in der OpenCV implementierte Feature-Detektoren wurden untersucht: *SURF*, *SIFT*, *ORB* und *GoodFeaturesToTrack*. Außerdem wurden zur Verknüpfung der gefundenen markanten Punkte die Feature-Matcher *FLANN* (Fast Approximate Nearest Neighbor Search Library) und *BFMatcher* (BruteForceMatcher), sowie *OpticalFlow* untersucht. Bei kleiner Stereo-Basis (Kameraverschiebung um weniger als 50 cm) wurden markante Punkte gefunden. Darüber hinaus konnte bei größerer Stereo-Basis keine der genannten Methoden ausreichend viele Punkte erkennen. Dies hängt unter anderem mit der Entfernung und der Größe des Gebäudes zusammen. Bei sehr großen Gebäuden sind Ergebnisse mit Verschiebung von bis zu zwei Meter möglich. Da aber auch kleinere Gebäude vermessen werden sollen, gibt es keine Lösung, die in allen Szenarien anwendbar ist. Diese Thematik hat sich somit als wesentlich komplexer herausgestellt, als zunächst angenommen. Daher wurde die Extraktion und Verknüpfung markanter Punkte manuell durchgeführt.

4.4 Berechnung der Transformation

Es gibt viele Möglichkeiten, die Berechnung der Transformation vorzunehmen. Bei allen betrachteten Möglichkeiten wurde eine kalibrierte Kamera vorausgesetzt. Folgende Möglichkeiten wurden betrachtet:

- a) Kalibrierobjekt in der Szene
- b) Rechteck in der Szene
- c) Drei abstandsgleiche Punkte auf einer Gerade [15]
- d) Zerlegung der Essentiellen Matrix mit gegebener Fundamentalmatrix [8]
- e) Optimierungsstrategie C (siehe 4.6.3)

Die Möglichkeiten a, b und c setzen koplanare, äquidistante Punkte voraus. Die Möglichkeiten a und b werden in 4.4.1 betrachtet. Möglichkeit c wird nicht weiter betrachtet, da die Erkennung drei geeigneter Punkte schwierig ist. Möglichkeit d erfordert mindestens sieben korrespondierende Punkte zur Berechnung der Fundamentalmatrix. Das Ergebnis der anschließenden Transformationsberechnung hängt von der berechneten Fundamentalmatrix ab. Möglichkeit e benötigt einen geeigneten Startwert. Bei Bildern von Gebäuden aus einem Abstand von mehr als 15 Metern ist eine Identitäts-Rotation häufig ausreichend. Als Startwert für die Translation wird der gespeicherte Abstand der Kamerapositionen verwendet.

4.4.1 Berechnung mit erkanntem Objekt

Für die Erkennung ist zum Beispiel eine Hauswand geeignet, auf der ein Fenster beobachtet wird (vgl. “Camera Models and Calibration” und “Homography“ in [1]). Die Ecken des Fensters werden wie bei einem Schachbrett von links oben beginnend durchnummeriert. Die linke obere Ecke wird mit $(0,0,0)$ und die rechte untere Ecke mit $(1,1,0)$ nummeriert. Nun kann mit Hilfe von OpenCV die Pose der Kamera zum Objekt berechnet werden. Das Verfahren wird in Abbildung 15 dargestellt, wobei die Kamerapose der ersten Kamera mit der Abbildung T^* und die der zweiten Kamera mit T^{**} bezeichnet ist.

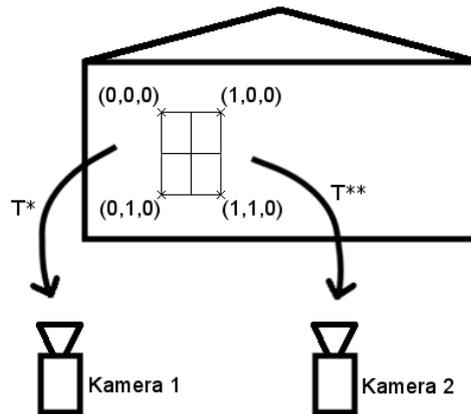


Abbildung 15: Abbildungen T^* und T^{**}

Die Abbildungen T^* und T^{**} transferieren das Objektkoordinatensystem in das Kamerakoordinatensystem der jeweiligen Kamera. Da die Objektpunkte $P_{obj} = \{(0,0,0), (1,0,0), (0,1,0), (1,1,0)\}$ bekannt sind, ist es möglich eine Transformation T zwischen den beiden Kamerakoordinatensystemen zu konstruieren:

$$T = (T^*)^{-1} \circ T^{**}$$

4.4.2 Normierung der Translation

Wenn der Abstand d zwischen den beiden Kameras bekannt ist, lässt sich der Skalierungsfaktor s der Kamerabbildung berechnen. Sei t die errechnete Translation der Kamerapositionen. Somit gilt:

$$s = \frac{d}{\|t\|} = \sqrt{\left(\frac{d}{\sqrt{x^2 + y^2 + z^2}}\right)^2} = \sqrt{\frac{d^2}{x^2 + y^2 + z^2}}$$

In diesem Sinne lässt sich die Translation durch $t_0 = s \cdot t$ normieren.

4.5 Triangulierung

Die Berechnung der Weltpunkte aus korrespondierenden Punkten wird Triangulierung genannt. Dabei wird der Schnittpunkt von zwei Sichtstrahlen im Raum berechnet. Dieser Schnittpunkt muss nicht zwangsläufig existieren. In diesem Fall wird der Punkt mit dem kürzesten Abstand zu beiden Sichtstrahlen ermittelt. Für die Triangulierung müssen die Kameras kalibriert und die Transformationen bekannt sein. Zunächst wird die Verzeichnung der Kamera außer Acht gelassen und eine lineare Schätzung des Startwertes der Weltpunkte vorgenommen. Somit ergibt sich der Startwert als Lösung eines überbestimmten linearen Gleichungssystem. Im Anschluss wird dieser Startwert unter Betrachtung der Verzeichnung der Kamera optimiert.

4.5.1 Startwertberechnung

Es wird eine lineare Schätzung des Weltpunktes für alle korrespondierenden Bildkoordinaten vorgenommen (vgl. „Startwert für die Stereorekonstruktion“ in [7]). Seien $i_{p,1} = (u_1, v_1)^T$, $i_{p,2} = (u_2, v_2)^T$ und $p = (x, y, z)^T$. Dann folgt aus der Kamerabbildung (siehe 3.2.2), die Existenz zweier Skalare $\lambda_1, \lambda_2 \in \mathbb{R}$ für die gilt:

$$\lambda_1 \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \tilde{K} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$\lambda_2 \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = \tilde{K}' \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Die beiden Skalare ergeben sich jeweils aus den dritten Zeilen der Kameramatrizen:

$$\lambda_1 = (\tilde{K}_{3,1}, \tilde{K}_{3,2}, \tilde{K}_{3,3}) \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \tilde{K}_{3,4}$$

$$\lambda_2 = (\tilde{K}'_{3,1}, \tilde{K}'_{3,2}, \tilde{K}'_{3,3}) \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \tilde{K}'_{3,4}$$

Somit ergibt sich ein lineares Gleichungssystem:

$$A \begin{pmatrix} x \\ y \\ z \end{pmatrix} = d \quad \text{mit}$$

$$A = \begin{pmatrix} \tilde{K}_{1,1} - u_1 \tilde{K}_{3,1} & \tilde{K}_{1,2} - u_1 \tilde{K}_{3,2} & \tilde{K}_{1,3} - u_1 \tilde{K}_{3,3} \\ \tilde{K}_{2,1} - v_1 \tilde{K}_{3,1} & \tilde{K}_{2,2} - v_1 \tilde{K}_{3,2} & \tilde{K}_{2,3} - v_1 \tilde{K}_{3,3} \\ \tilde{K}'_{1,1} - u_2 \tilde{K}'_{3,1} & \tilde{K}'_{1,2} - u_2 \tilde{K}'_{3,2} & \tilde{K}'_{1,3} - u_2 \tilde{K}'_{3,3} \\ \tilde{K}'_{2,1} - v_2 \tilde{K}'_{3,1} & \tilde{K}'_{2,2} - v_2 \tilde{K}'_{3,2} & \tilde{K}'_{2,3} - v_2 \tilde{K}'_{3,3} \end{pmatrix} \quad \text{und} \quad d = \begin{pmatrix} u_1 \tilde{K}_{3,4} - \tilde{K}_{1,4} \\ v_1 \tilde{K}_{3,4} - \tilde{K}_{2,4} \\ u_2 \tilde{K}'_{3,4} - \tilde{K}'_{1,4} \\ v_2 \tilde{K}'_{3,4} - \tilde{K}'_{2,4} \end{pmatrix}$$

Daraus ergeben sich vier Gleichungen für drei Unbekannte. Ein solches überbestimmtes LGS ist im Allgemeinen nicht lösbar. Man bestimmt daher ein p , mit der Methode der kleinsten Quadrate, das folgende Gleichung minimiert:

$$p = \operatorname{argmin}_{q \in \mathbb{R}^3} \|Aq - d\|^2$$

Die Lösung dieses lin. Ausgleichsproblems erhält man für reguläre Matrizen $A^T \cdot A$ wie folgt:

$$p = (A^T \cdot A)^{-1} A^T d$$

4.5.2 Startwertberechnung mit der OpenCV-Bibliothek

Die OpenCV-Bibliothek stellt eine Methode zur Singulärwertzerlegung zur Verfügung, welche die Berechnung der Inverse erlaubt. Außerdem ist eine Methode zur „optimalen Triangulierung“ [8, 9] bereits implementiert. Diese Methode optimiert zunächst die korrespondierenden Bildkoordinaten und berechnet aus den optimierten Bildkoordinaten die Weltpunkte.

4.5.3 Optimierung der Startwerte

Im Folgenden werden die Startwerte der Weltpunkte unter Betrachtung der Verzeichnung optimiert. Voraussetzung ist eine bekannte Transformation zwischen den beiden Kameras und außerdem ein Startwert für den Weltpunkt $p \in \mathbb{R}^3$. Die Transformation wird nicht optimiert, sondern nur die Position des Weltpunktes. Hierzu werden die Weltpunkte in das Bild der jeweiligen Kamera projiziert und der Abstand zu den jeweils bekannten, korrespondierenden Bildkoordinaten gemessen. Seien K_1, K_2 Kameraabbildungen, p der zu optimierende Weltpunkt und i_1, i_2 die korrespondierenden Bildkoordinaten des Weltpunktes p in der ersten bzw. zweiten Kamera. Für die zu minimierende Funktion gilt:

$$f_0 : \mathbb{R}^3 \rightarrow \mathbb{R}, p \mapsto \|K_1(p) - i_1\|^2 + \|K_2(p) - i_2\|^2$$

4.6 Optimierungsstrategien

In diesem Abschnitt werden drei Methoden zur nicht-linearen Optimierung des 3D-Modells vorgestellt. Die Methoden setzen voraus, dass Startwerte für die Transformation, sowie für die Weltpunkte der Szene, vorliegen. Weiterhin wird angenommen, dass jeder Weltpunkt in allen Bildern sichtbar ist. In diesem Abschnitt wird folgende Notation verwendet: K_k bezeichne die k -te Kameraabbildung. $i_{1,j}$ und $i_{2,j}$ bezeichnen zwei korrespondierende Punkte im Bild der ersten und zweiten Kamera. Aus diesen Bildpunkten wurde zuvor der j -te Weltpunkt p_j berechnet.

4.6.1 Optimierungsstrategie A

Optimierungsstrategie A wird in der Literatur als *Bundle Adjustment* bezeichnet. Bundle Adjustment ist die vorherrschende Methodik für das allgemeine Rekonstruktionsproblem. [8]

Lochkamerastereosystem

Mit einem Lochkamerastereosystem (K_1, K_2) wird zunächst jeder Weltpunkt in die erste Kamera abgebildet und es wird der Abstand des projizierten Bildpunktes zu der bekannten Bildkoordinate $i_{1,j}$ gemessen (vgl. Abbildung 16). Für die Projektion in der zweiten Kamera werden die Weltpunkte zunächst in das Kamerakoordinatensystem der zweiten Kamera transformiert und schließlich abgebildet. Hier wird der Abstand zu der bekannten Bildkoordinate $i_{2,j}$ gemessen. Der Fehler ergibt sich aus der Summe der quadrierten Abstände.

Die Startwerte für die Rotation $r := (r_1, r_2, r_3)^T$ und die Translation $t := (t_1, t_2, t_3)^T$ der Kameraposition sowie die m Weltpunkte $p_j := (p_{jx}, p_{jy}, p_{jz}) \in \mathbb{R}^3, 1 \leq j \leq m, m \geq 6, m \in \mathbb{N}$ sind Parameter des Optimierers. Das vollständige Tupel ϕ_1 in der Fehlerfunktion lautet wie folgt:

$$\phi_1 = \left(\overbrace{r_1, r_2, r_3}^r, \overbrace{t_1, t_2, t_3}^t, \overbrace{p_{1x}, p_{1y}, p_{1z}}^{p_1}, \overbrace{p_{2x}, p_{2y}, p_{2z}}^{p_2}, \dots, \overbrace{p_{mx}, p_{my}, p_{mz}}^{p_m} \right) \in \mathbb{R}^{6+3m}$$

Man optimiert folgende Fehlerfunktion:

$$f_1 : \mathbb{R}^{6+3m} \rightarrow \mathbb{R}, \phi_1 \mapsto \sum_{j=1}^m \|K_1(p_j) - i_{1,j}\|^2 + \|K_2(p_j) - i_{2,j}\|^2$$

Lochkameramultisystem

Das Bundle Adjustment kann auf ein Multilochkamerasystem (K_1, K_2, \dots, K_n) angewendet werden. Für n Kameras beschreiben $n-1$ Transformationen T_2, \dots, T_n deren Anordnung. Die Rotation $r_k := (r_{k1}, r_{k2}, r_{k3}), k \in \mathbb{N}, 2 \leq k \leq n$ und die Translation $t_k := (t_{k1}, t_{k2}, t_{k3}), k \in \mathbb{N}, 2 \leq k \leq n$ beschreiben die Transformation in das Koordinatensystem der k -ten Kamera. Auch hier sind mindestens sechs Weltpunkte erforderlich: $p_j := (p_{jx}, p_{jy}, p_{jz}) \in \mathbb{R}^3, 1 \leq j \leq m, m \geq 6, m \in \mathbb{N}$. Dabei bezeichnet ϕ_2 das Tupel der Form:

$$\phi_2 = \left(\overbrace{r_{21}, r_{22}, r_{23}}^{r_2}, \overbrace{t_{21}, t_{22}, t_{23}}^{t_2}, \overbrace{r_{31}, r_{32}, r_{33}}^{r_3}, \overbrace{t_{31}, t_{32}, t_{33}}^{t_3}, \dots, \overbrace{r_{n1}, r_{n2}, r_{n3}}^{r_n}, \overbrace{t_{n1}, t_{n2}, t_{n3}}^{t_n}, \right. \\ \left. \overbrace{p_{1x}, p_{1y}, p_{1z}}^{p_1}, \overbrace{p_{2x}, p_{2y}, p_{2z}}^{p_2}, \dots, \overbrace{p_{mx}, p_{my}, p_{mz}}^{p_m} \right) \in \mathbb{R}^{6(n-1)+3m}$$

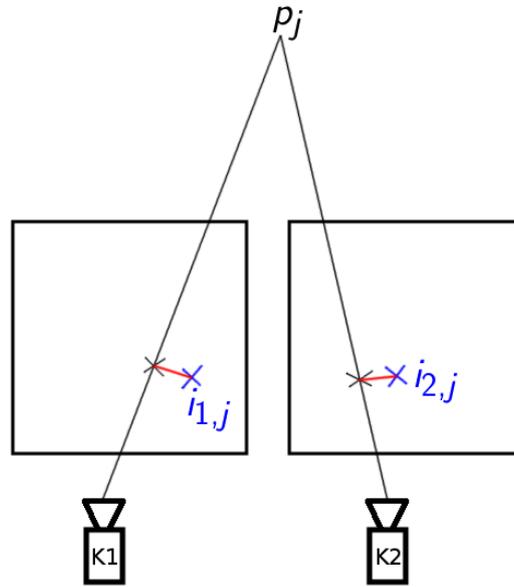


Abbildung 16: Bundle Adjustment

Optimiert wird folgende Fehlerfunktion:

$$f_2 : \mathbb{R}^{6(n-1)+3m} \rightarrow \mathbb{R}, \phi_2 \mapsto \sum_{k=1}^n \sum_{j=1}^m \|K_k(p_j) - i_{k,j}\|^2$$

4.6.2 Optimierungsstrategie B

Aus Kapitel 3.2 ist bekannt, dass das Urbild von P_z eine Ursprungsgerade ist. Kennt man den Abstand der angenommenen Position zum Kamerazentrum, lässt sich der Weltpunkt rekonstruieren. Da in allen Perspektiven dieselbe Kamera und dieselbe Projektionsmatrix verwendet wird, kann eine für alle Perspektiven gültige Funktion definiert werden. Sei $\mathbf{M} \left(\begin{pmatrix} u \\ v \\ 1 \end{pmatrix}; \mathbf{d} \right)$ die Funktion, die eine homogenisierte Bildkoordinate (u, v) in Abstand d auf den Weltpunkt p abbildet. Es sind folgende drei Schritte notwendig:

1. Normalisierung der Bildkoordinate $i = (u, v)$:

$$\tilde{P}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = s \cdot p_0$$

2. Bestimmung des Skalierungsfaktors s für p_0 :

$$s = \frac{d}{\|p_0\|} = \sqrt{\frac{d^2}{x^2 + y^2 + 1}}$$

3. Berechnung des Weltpunktes p :

$$p = s \cdot p_0 = s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$$

Lochkamerastereosystem

Bei der Anwendung dieser Optimierungsstrategie auf ein Lochkamerastereosystem werden sechs Parameter für die Transformation des Standortes von Kamera 1 zu Kamera 2 und zwei Distanzparameter je Weltkoordinate verwendet. Die Distanzparameter geben den Abstand des Weltpunktes zum Kamerazentrum der jeweiligen Kamera an. Es wird der Fehler der Projektion aller Bildkoordinaten in das jeweils andere Bild gemessen. Zur Berechnung des Fehlers einer Bildkoordinate wird der zugehörige Weltpunkt p mit Abstand d_1 berechnet. Dieser wird in das Kamerakoordinatensystem der zweiten Kamera transformiert und dort auf die Bildfläche der zweiten Kamera projiziert. Auf der Bildfläche wird der Abstand der projizierten Bildkoordinaten zu der bekannten korrespondierenden Bildkoordinate gemessen. Umgekehrt wird ebenfalls für die Bildkoordinate der zweiten Kamera der Weltpunkt p mit Abstand d_2 berechnet, in Kamera 1 zurücktransformiert und auf die Bildfläche projiziert. Wieder wird der Abstand der projizierten Bildkoordinaten zu der bekannten korrespondierenden Bildkoordinate gemessen.

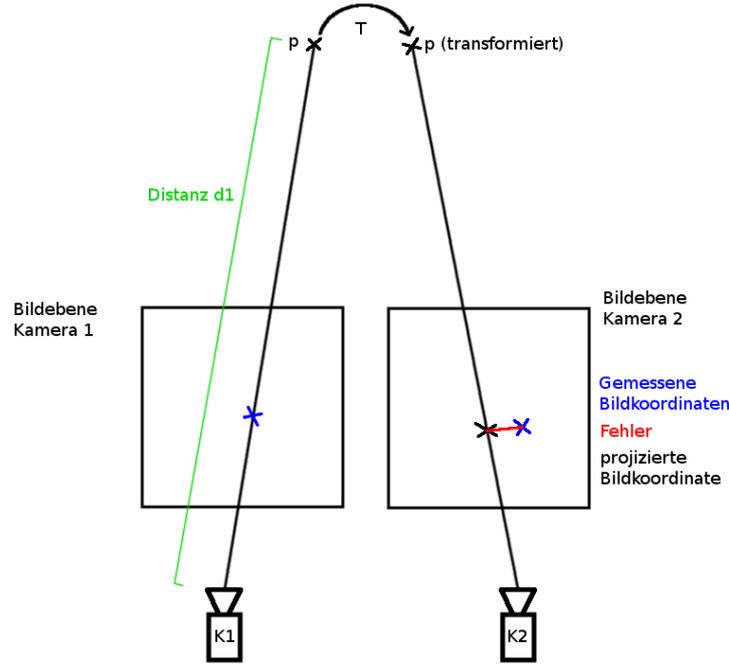


Abbildung 17: Distanzminimierung projizierter Punkte

Seien $r := (r_1, r_2, r_3)^T$ die Rotation und $t := (t_1, t_2, t_3)^T$ die Translation des Referenzkoordinatensystems von Kamera 1 in das Koordinatensystem von Kamera 2. Mit der Rodrigues-Formel wird die Rotationsmatrix R zur gegebenen Rotation r berechnet. Seien $i_{1,j}$ und $i_{2,j}$ korrespondierende Bildkoordinaten in Kamera 1 und 2. Da die Funktion M Weltkoordinaten bezüglich des jeweiligen Kamerakoordinatensystems konstruiert, muss die in der zweiten Kamera errechnete Weltkoordinate in das Koordinatensystem der ersten Kamera transformiert werden. Dann ist mit folgender Summe die zu minimierende Fehlerfunktion für Strategie B gegeben:

$$\phi_3 = \left(\overbrace{r_1, r_2, r_3}^r, \overbrace{t_1, t_2, t_3}^t, \overbrace{d_{1,1}, d_{2,1}}^{d_1}, \overbrace{d_{1,2}, d_{2,2}}^{d_2}, \dots, \overbrace{d_{1,m}, d_{2,m}}^{d_m} \right)$$

$$f_3 : \mathbb{R}^{6+2m} \rightarrow \mathbb{R}, \phi_3 \mapsto \sum_{j=1}^m \|K_2(M(i_{1,j}; d_{1,j})) - i_{2,j}\|^2 + \|K_1(R^{-1} \cdot (M(i_{2,j}; d_{2,j}) - t)) - i_{1,j}\|^2$$

Multilochkamarasystem (K_1, K_2, \dots, K_n)

Optimierungsstrategie B kann ebenfalls auf n Perspektiven (Bilder) erweitert werden. Als Eingabe dienen nun die Transformationen T_2, \dots, T_n und mindestens drei Weltpunkte: $p_j \in \mathbb{R}^3, j \in \{3, \dots, m\}$, sowie $n \cdot m$ Distanzparameter. Die zu minimierende Funktion f_4 ist folgendermaßen gegeben:

$$\phi_4 = \left(\overbrace{r_{21}, r_{22}, r_{23}}^{r_2}, \overbrace{t_{21}, t_{22}, t_{23}}^{t_2}, \overbrace{r_{31}, r_{32}, r_{33}}^{r_3}, \overbrace{t_{31}, t_{32}, t_{33}}^{t_3}, \dots, \overbrace{r_{n1}, r_{n2}, r_{n3}}^{r_n}, \overbrace{t_{n1}, t_{n2}, t_{n3}}^{t_n}, \right. \\ \left. \overbrace{d_{1,1}, d_{2,1}}^{d_1}, \overbrace{d_{1,2}, d_{2,2}}^{d_2}, \dots, \overbrace{d_{1,m}, d_{2,m}}^{d_m} \right) \in \mathbb{R}^{6(n-1)+2m}$$

$$f_4 : \mathbb{R}^{6(n-1)+2m} \rightarrow \mathbb{R}, \phi_4 \mapsto \sum_{k=2}^n \sum_{j=1}^m \|K_k(M(i_{1,j}; d_{1,j})) - i_{k,j}\|^2 + \|K_1(T_k^{-1} \cdot M(i_{k,j}; d_{k,j})) - i_{1,j}\|^2$$

4.6.3 Optimierungsstrategie C

Diese Strategie minimiert den Abstand zweier Sichtstrahlen im \mathbb{R}^3 . Damit unterscheidet sie sich wesentlich von den beiden anderen Strategien, welche die Abweichung mehrerer projizierter Punkte auf der Bildfläche messen. Optimierungsstrategie C optimiert nur die sechs Parameter für die Transformation des Kamerastandortes.

Optimierungsstrategie C wird auf das Lochkamerastereosystem angewandt. Seien $(u_{1,j}, v_{1,j})$ und $(u_{2,j}, v_{2,j})$ korrespondierende Bildkoordinaten in der ersten bzw. der zweiten Kamera. Seien A_j, B_j Sichtstrahlen durch diese korrespondierende Bildkoordinaten. Seien $r := (r_1, r_2, r_3)^T$ die Rotation und $t := (t_1, t_2, t_3)^T$ die Translation, die die Bewegung vom Referenzkoordinatensystem der ersten Kamera in das Koordinatensystem der zweiten Kamera beschreiben. Dann gilt für Fehlerfunktion:

$$f_5 : \mathbb{R}^6 \rightarrow \mathbb{R}, (r_1, r_2, r_3, t_1, t_2, t_3) \mapsto \sum_{j=1}^m \min_{x \in A_j} \min_{y \in B_j} \|x - y\|$$

Hier ergibt sich das Problem, dass der nicht-lineare Optimierer die Kamerapositionen zusammenschieben kann und die Sichtstrahlen sich somit im Ursprung schneiden. Dies ist natürlich nicht gewünscht, daher wurde der Umkehrwert der Entfernung des Schnittpunktes der Sichtstrahlen als Strafterm eingeführt. Sei $d_j \neq 0$ die Distanz des j -ten Weltpunktes vom Kamerazentrum.

Die erweiterte Fehlerfunktion lautet nun wie folgt:

$$f_6 : \mathbb{R}^6 \rightarrow \mathbb{R}, (r_1, r_2, r_3, t_1, t_2, t_3) \mapsto \sum_{j=1}^m \frac{1}{d_j} \min_{x \in A_j} \min_{y \in B_j} \|x - y\|$$

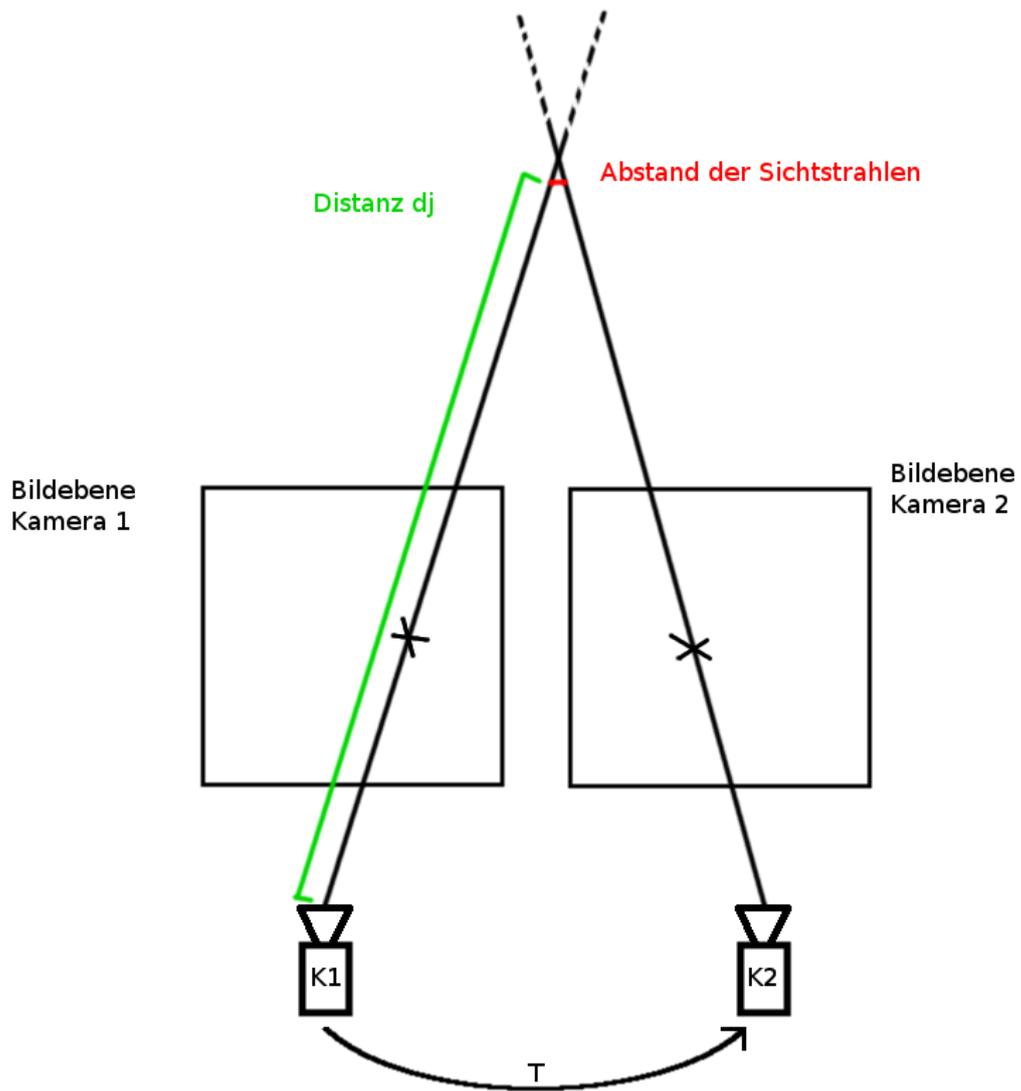


Abbildung 18: Sichtstrahl-Distanzminimierung

5 Implementierung

Im Rahmen dieser Bachelor Arbeit entstanden zwei Anwendungen. Die erste Anwendung ist eine Smartphone-Anwendung, welche die Aufnahme einer Szene ermöglicht und die zweite Anwendung ist eine Desktop-Anwendung, die aus der aufgenommenen Szene ein 3D-Modell berechnet. Beide Anwendungen wurden in Java geschrieben und verwenden die OpenCV-Bibliothek. Als Nachschlagewerk und zur Einarbeitung waren folgende Bücher und Webseiten hilfreich:

- Java concurrency in practice [5]
- Learning OpenCV [1]
- OpenCV 2 Computer Vision Application Programming Cookbook [12]
- OpenCV Dokumentation [3, 4]
- OpenCV Tutorials ⁵

5.1 Persistenzmodell

Beiden Anwendungen liegt ein gemeinsames Persistenzmodell zu Grunde, welches in Abbildung 19 in Form eines ER-Diagramms dargestellt wird. Die Kamera-Entität im Diagramm repräsentiert die Kamera, die die Bilder aufgenommen hat. Von dieser Kamera sind nach der Kalibrierung die intrinsischen Kameraparameter und die Verzeichnungsparameter bekannt. Die Kamera nimmt eine Szene auf. Diese Szene besteht aus mehreren Bildern. Zu den Bilddaten wird die Reihenfolge (Nummer), die GPS-Positionen und der Abstand der Kamerapositionen gespeichert. Bilder können mit Transformationen verknüpft werden. Die gemessene Distanz zwischen den Kamerapositionen wird zur Skalierung der Translation verwendet. Korrespondierende Punkte verknüpfen die beobachteten Bildkoordinaten in den verschiedenen Perspektiven mit dem gemeinsamen Welt-punkt. Ein Objekt in der Szene wird von mehreren korrespondierenden Punkten beschrieben. Ein erkanntes Schachbrett oder ein ausgewähltes Fenster kann als solches Objekt gespeichert werden.

5.2 Smartphone-Anwendung

Für die Entwicklung einer eigenen Smartphone-Anwendung gibt es einige wichtige Gründe. Zunächst muss der **Fokus der Kamera konstant** sein, da das verwendete Lochkameramodell eine feste Brennweite der Kamera fordert. Dazu wird diese konstant auf *unendlich* gesetzt. Bei der kleinen Brennweite der Smartphone-Kamera beginnt der Bereich der Schärfentiefe bereits nach wenigen Metern und somit stellt diese Einstellung keine Einschränkung für die Anwendung dar. Weiterhin muss bei der Aufnahme die aktuelle Position in **geographischen Koordinaten** gespeichert werden. Aus diesen Informationen wird der Abstand zwischen den Standorten berechnet werden. Zusätzlich kann ein **Name** und eine **Beschreibung für die Szene** vergeben werden, um die Adresse des Gebäudes zu speichern. Dies ist für die Auswertung der Messdaten und die Eintragung der Höhe in OpenStreetMap unabdingbar. Der **Zeitauslöser** der Anwendung minimiert das Risiko von verwackelten Bildern. Diese nützliche Funktion fehlt in vielen Standard-Anwendungen von Android. Überdies hinaus ist eine **Serienbildfunktion** bei einem routinierten Messablauf nützlich.

5.2.1 Stativ für Kalibrierung und Testbilder

Für die Kalibrierung ist ein Smartphone Stativ-Adapter (vgl. Abbildung 20) sehr hilfreich. Der Stativ-Adapter wurde zusätzlich auf ein 15 cm langes Aluminium-Winkelprofil geklebt, welches

⁵http://docs.opencv.org/doc/tutorials/features2d/table_of_content_features2d/table_of_content_features2d.html,
http://docs.opencv.org/doc/tutorials/calib3d/table_of_content_calib3d/table_of_content_calib3d.html

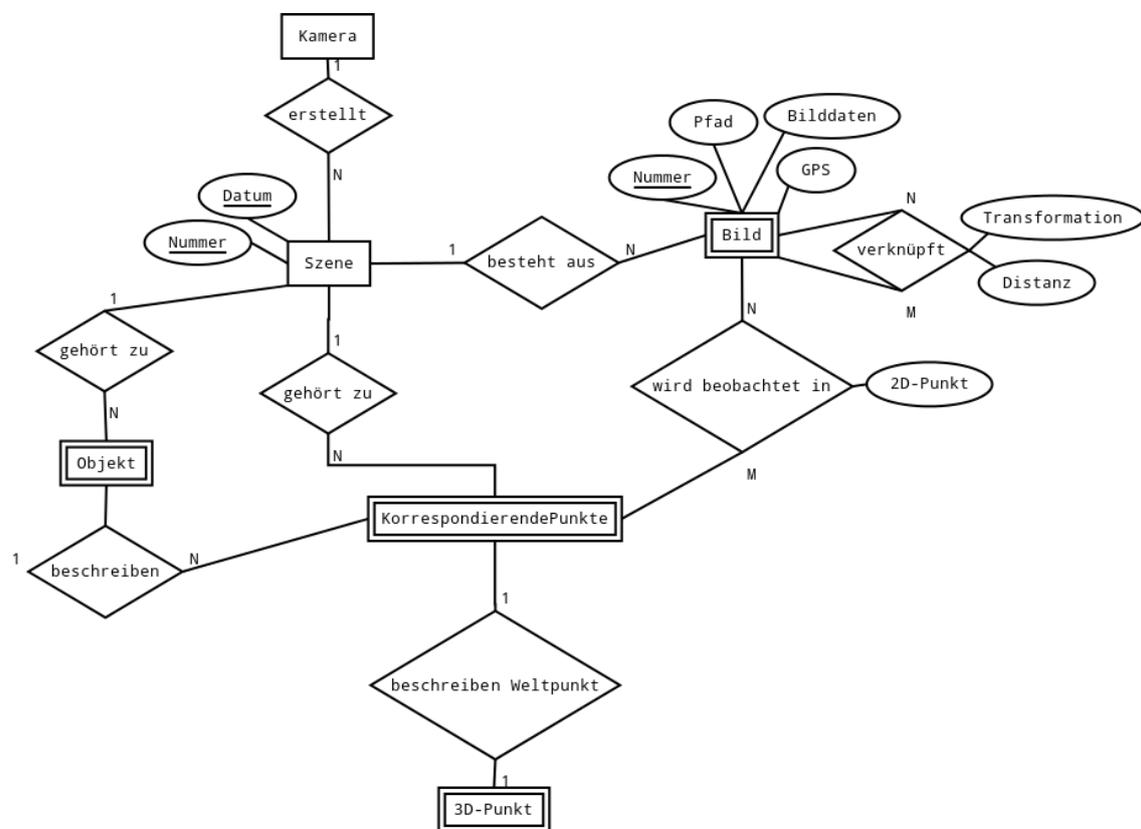


Abbildung 19: ER-Diagramm Szene

ein Zuschnitt eines größeren Winkelprofils mit den Abmessungen 20x30x2 mm und 1 Meter Länge ist. Der restliche Teil des Winkelprofils wurde als Leiste verwendet, die man mit Hilfe einer Schraubzwinge auf einem normalen Fotostativ befestigen kann (Abbildung 21). Hiermit lassen sich Testbilder mit reiner Translation (Rotation entspricht nahezu der Identität) der Kameraposition erstellen, welche zu Testzwecken der Algorithmen verwendet wurden.



Abbildung 20: Smartphone-Stativ-Adapter



Abbildung 21: Adapter auf Aluminiumleiste

5.2.2 Kalibrierung der Smartphone-Kamera

Der Stativ-Adapter und die Serienbildfunktion der Smartphone-Anwendung sind praktische Hilfsmittel bei der Kamerakalibrierung. Es kann eine Verzögerung festgelegt werden, die dem Anwender genügend Zeit lässt das Kalibrieremuster neu zu positionieren. Ein akustisches Signal weist den Benutzer darauf hin, wann ein Bild aufgenommen wird. Dabei werden ungefähr zehn Bilder des Kalibrieremusters benötigt. Idealerweise nimmt man einige Bilder mehr auf und wählt später ein gutes Set aus. Es ist darauf zu achten, dass das Kalibrieremuster in möglichst unterschiedlichen Positionen und Entfernungen zur Kamera aufgenommen wird.

Die Bilder können zur Kalibrierung in der Desktop-Anwendung geladen werden. Mit der Smartphone-Anwendung ist ebenfalls eine Kalibrierung möglich. Dies ist aber noch nicht ratsam, weil die Ausrichtung der Kalibrieremuster bisher noch nicht visuell kontrolliert werden kann. Auf dem *Samsung-Galaxy-Nexus*-Smartphone hat die Kalibrierung mit fünf Bildern 35 Minuten benötigt. Bei dem aktuelleren *LG-Nexus-4*-Smartphone mit Quad-Core-Prozessor und einer parallelen Mustererkennung ist eine Kalibrierung mit zehn Bildern in fünf Minuten fertig. Eine Kalibrierung ist also nur mit einem neueren, leistungsstärkeren Smartphone in angemessener Zeit möglich.



Abbildung 22: Kamerakalibrierung

5.3 Desktop-Anwendung

Die graphische Benutzeroberfläche ist in fünf Teile mit einer Übersicht eingeteilt. Die Aufteilung orientiert sich dabei an die in 4.2 aufgeführten Schritte zur Berechnung des 3D-Modells.

5.3.1 Extraktion und Verknüpfung markanter Punkte

Im ersten Reiter (vgl. Abbildung 23) werden korrespondierende Bildpunkte ausgewählt und verknüpft. Die Position der Bilder kann verschoben und skaliert werden. Weiterhin kann mit den Pfeiltasten der Tastatur zu weiteren Bildern gewechselt werden, wenn die Szene mehr als zwei Bilder enthält. Zusätzlich wird eine markierte Bildkoordinate in einem kleinen Fenster im linken oberen Eck des jeweiligen Bildes in fünffacher Vergrößerung dargestellt. Dieses kleine Fenster hilft bei der genauen Auswahl der Bildkoordinate. Gespeicherte korrespondierende Bildkoordinaten sind editierbar und werden in einer Liste angezeigt. Für die spätere Messung müssen an dieser Stelle bereits drei geeignete Punkte ausgewählt werden.

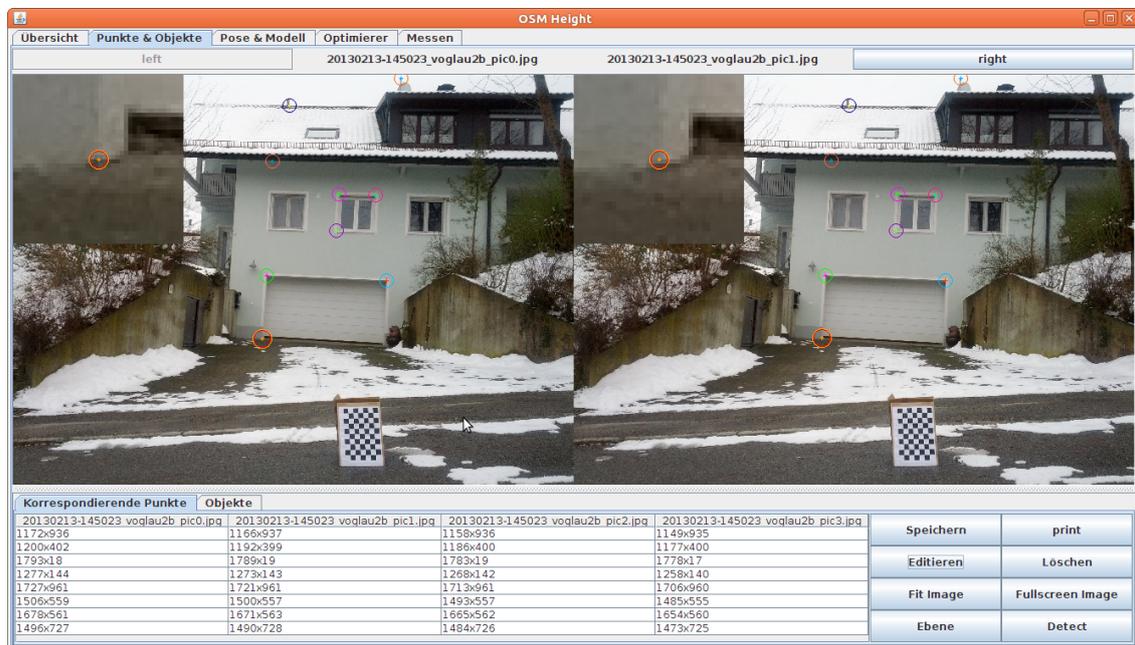


Abbildung 23: Punkte markieren und verknüpfen

Mehrere korrespondierende Punkte sind als Objekt markierbar. Dieses Objekt kann im nächsten Reiter zur Berechnung der Transformation verwendet werden.

5.3.2 Berechnung der Transformation und Triangulierung

Im zweiten Reiter (vgl. Abbildung 24) wird zunächst ein Bild als *Ursprungsbild* markiert. Weiterhin muss ein zweites Bild als *Referenzbild* markiert werden. Dies geschieht mit einem linken bzw. rechten Mausklick auf die Bilder in der oberen Leiste. Das Referenzbild ist notwendig, da die Methoden zur Triangulierung zwei Bilder verwenden: das Ursprungsbild und das Referenzbild. Anschließend kann ein Schachbrett oder ein anderes Objekt zur Berechnung der Transformation verwendet werden. Alternativ kann ein Startwert für die Transformation manuell eingegeben werden. Nun wird eine der Methoden zur Triangulierung verwendet, um Startwerte für die Weltpunkte zu berechnen. Danach wird Optimierungsstrategie C zur Optimierung der Transformation verwendet. Nachdem die Optimierungsstrategie beendet ist, wird die errechnete Transformation auf die gemessene Distanz skaliert. Schließlich wird erneut eine Methode zur Triangulierung angewendet, um Startwerte für die Weltpunkte mit optimierter Transformation zu berechnen.

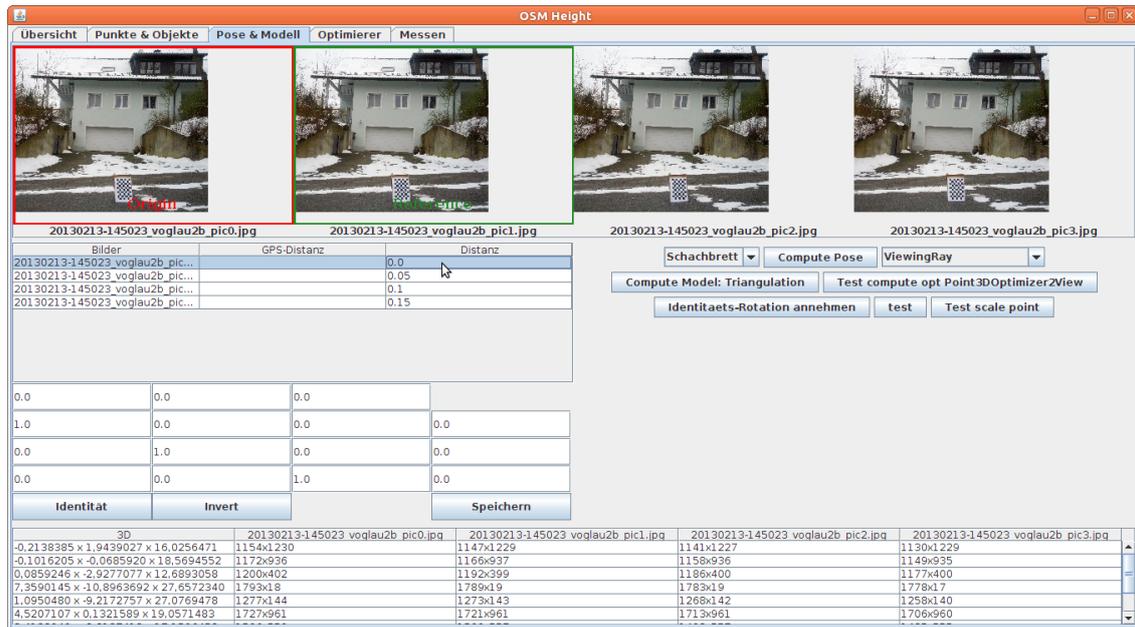


Abbildung 24: Transformation und Triangulierung

5.3.3 Optimierung

Für die Optimierungsstrategien wurde der nicht-lineare Optimierer *Nelder-Mead* aus der *Apache-Commons-Math-Bibliothek* verwendet. Im nächsten Schritt wurden die Weltpunkte mit Funktion f_0 einzeln optimiert (vgl. 4.5.3). Der letzte Schritt der Modelloptimierung ist ein Aufruf von Optimierungsstrategie A (Bundle Adjustment, vgl. 4.6.1).

5.3.4 Messung

Im fünften Reiter der Anwendung können nun die zuvor für die Messung erstellten Punkte markiert werden. Aus den Weltpunkten der korrespondierenden Punkte wird die Höhe errechnet.

5.3.5 Quellcode

Für die Android-Anwendung wurden 2361 Zeilen Code benötigt. Davon waren 609 Zeilen für den parallelen Kalibrierungsalgorithmus notwendig, welcher mit einem Java-Thread-Pool umgesetzt wurde. Weitere 142 Zeilen Code wurden benötigt, um die Messdaten in 3.4.3 zu speichern. Die Anwendungsfenster mit denen der Benutzer die Android-Anwendung bedient, werden *Activities* genannt. Der Rest des Codes wurde zur Entwicklung dieser *Activities* aufgewendet.

Zur Programmierung der Desktop-Anwendung wurden knapp 10000 Zeilen Code benötigt, aber der Kern der Optimierungsstrategien benötigte dabei jeweils nur 50 bis 100 Zeilen Code. Die meisten Codezeilen wurden für die Entwicklung der verschiedenen Reiter, Panels, Tabellen und Bild Darstellungen gebraucht. Der Hauptteil des Codes für die Optimierung lag auf durchzuführenden Kontrollen (zusätzlich zu den 50 bis 100 Zeilen Code). Diese prüften ob alle geforderten Startwerte vorhanden sind, und ob das Ursprungsbild und das Referenzbild markiert wurden. Zusätzlich gibt es je eine Implementierung für zwei Perspektiven und eine für mehrere Perspektiven.

6 Evaluation

In diesem Kapitel werden die Vor- und Nachteile der Methoden anhand folgender Kriterien evaluiert:

1. **Anwendbarkeit**
2. **Fehlerquellen, Messhindernisse**
3. **Aufwand**
4. **Kosten**
5. **Messungen**

6.1 Anwendbarkeit

Durch Kombination der in Abschnitt 2.1 und 2.2 angesprochenen Gebäudekategorien und Möglichkeiten der Positionierung des Messgerätes ergeben sich verschiedene Messszenarien. Tabelle 3 stellt die Anwendbarkeit der untersuchten Methoden auf diese Szenarien dar.

Szenario	Gebäude-kategorie	Position	Laser	Smart-Measure	3D-Rekonstruktion
1	I	ebenerdig	x	x	x
2	I	höher	x	(x)	x
3	I	tiefer			x
4	II	ebenerdig			x
5	II	höher			x
6	II	tiefer			x

Tabelle 3: Anwendbarkeit der untersuchten Methoden

Aus der Tabelle wird ersichtlich, dass die Laser-Methode und die Smart-Measure-Methode auf die Szenarien drei bis sechs nicht anwendbar sind, während die 3D-Rekonstruktion in allen Szenarien eine Messung ermöglicht. Die allgemeine Anwendbarkeit der 3D-Rekonstruktion zur Messung von Gebäudehöhen ist ein klarer Vorteil dieser Methode.

6.2 Fehlerquellen, Messhindernisse

Eine freie Sicht auf die zu messende Wand, ist bei allen drei Methoden erforderlich. Daher sind optische Barrieren, die die Sicht versperren, ein häufiges Messhindernis.

Laser-Methode

Bei Anwendung der direkten Laser-Methode ist ein Zugang zur Gebäudewand unerlässlich, und es muss ein Dachüberstand vorhanden sein. Die anderen beiden Varianten der Laser-Methode benötigen eine Möglichkeit den Laser im rechten Winkel auf das Gebäude zu richten. Weiterhin darf das Messobjekt nicht zu weit entfernt sein, da trotz einer Reichweite des Lasermessgerätes von 40 Metern, ist der Laserpunkt ab 20 Metern häufig nur schwer auffindbar und eine Messung daher nicht möglich. Befindet sich am Gebäude eine Fassadenbegrünung, so verhindert dies eine Messung, da der Laser unter diesen Umständen nicht gesehen wird. Darüber hinaus verursachen Reflektionen in Glasfenstern Messfehler. Ebenso ist auch helle Sonneneinstrahlung problematisch, da sie die Sicht auf den Laserpunkt verhindert. Ein bewölkter Himmel, während der Messung, ist daher von Vorteil. Bei Nacht ist der Laser wesentlich besser sichtbar und größere Distanzen sind messbar.

Smart-Measure-Methode

Es ist nur eine ebenerdige Messung möglich, daher ist die Wahl des Messstandortes ausschlaggebend für die Genauigkeit der Messung. Für die zweite Messung muss das Smartphone auf die Traufe gerichtet werden. Bei der Neigung des Smartphones sollte dieses nicht von der Stelle bewegt werden, da ansonsten das Messergebnis verfälscht wird.

3D-Rekonstruktions-Methode

Aus der allgemeinen Anwendbarkeit der Methode folgt, dass es für diese Methode nur wenige Messhindernisse gibt. Im Gegensatz zu den anderen Methoden gibt es bei dieser mehr Fehlerquellen, da wesentlich mehr Schritte notwendig sind. Eine ungenaue Abstandsmessung zwischen den Kamerapositionen vor Ort, verfälscht das Ergebnis. Bei Verwendung eines Feature Detektors können markante Punkte falsch verknüpft werden. Bei Gebäudebildern in dunklen Gassen mit einem Ausschnitt des Himmels auf den Bildern, können aufgrund der Mittelung der Helligkeit sehr dunkle Bereiche in den Bildern entstehen. Daher ist eine Auswahl markanter Punkte in diesen Bereichen nicht möglich. Bei hoher Sonneneinstrahlung oder Gegenlicht können die Bilder überbelichtet sein.

6.3 Aufwand

In diesem Abschnitt wurde der zeitliche Aufwand für eine Messung bewertet.

Laser-Methode

Bei der direkten Distanzmessung liegt ein Ergebnis in weniger als einer Minute vor. Der größte Aufwand ist, eine sinnvolle Messstelle zu finden, an der eine Messung zur Traufe möglich ist. Bei der Pythagoras-Messung ist die Messung ebenfalls schnell erstellt. Das Ergebnis kann mit dem Taschenrechner berechnet werden.

Smart-Measure-Methode

Ein Ergebnis liegt bei dieser Messmethode in weniger als einer Minute vor. Wie bei der Laser-Methode ist auch hier eine sinnvolle Messstelle zu finden. Die Berechnung der Höhe erledigt die Anwendung. Bei dieser Methode ist der Aufwand zur Messung am geringsten.

3D-Rekonstruktions-Methode

Die Erstellung der Bilder geht schnell (weniger als eine Minute). Zur Abstandsmessung kommt ein Meterstab zum Einsatz. Bei einer Messung mit einem Stativ dauert dieser Vorgang länger. Zusätzlich muss der Abstand zwischen den Kamerapositionen notiert werden. Dieser kann als Kommentar in der Szene vermerkt werden. Damit liegt aber noch kein Messergebnis vor. Dieses kann erst in der Desktop-Anwendung berechnet werden. Der Aufwand hierfür beträgt circa zehn Minuten. Es werden die markanten Punkte verknüpft, die Transformations-Berechnung durchgeführt und das erstellte Modell optimiert.

6.4 Kosten

Das verwendete Lasermessgerät Bosch DLE 40 kostet circa 100 Euro. Für die Smart-Measure-Methode und die 3D-Rekonstruktions-Methode wird ein Smartphone benötigt. Ein geeignetes Gerät kann bereits ab 100 Euro erworben werden. Die Standard-Version der Smart-Measure-Anwendung ist kostenlos.

6.5 Messungen

Abkürzungen:

- Laser, direkte Messung: LD
- Laser, Pythagoras-Messung: LP
- Smart-Measure: SM
- 3D-Rekonstruktion: 3D

Angaben in Meter, Abweichungen in Prozent.

Messung 1

Die erste Messung war von einem Gebäude der Kategorie II. Es wurden zwei Bilder in zwei Metern Abstand zueinander aufgenommen. In der Szene befand sich ein Schachbrett, das zur Transformationsberechnung verwendet wurde. Der Startwert aus der Transformationsberechnung wurde schließlich zur Optimierung des Modells verwendet.

Methode	Garage		Traufe		First
LD	2,11		5,691		
LP			5,733	+ 0,7 %	
SM	2,27	+ 7,6 %	6,23	+ 9,4 %	
3D	2,07	- 1,9 %	5,94	+ 4,4 %	9,49



Messung 2 (Innstraße 23, Innstegaula)

Das Gebäude hat ein Satteldach, dessen Giebelwand vermessen werden kann. Daher ist es in Kategorie I. Es wurden drei Bilder, mit Hilfe eines Stativs, gemacht. Der Abstand Kamerapositionen betrug 30 cm. Zur Berechnung des 3D-Modells wurden zunächst zwei Bilder, danach drei Bilder verwendet. Eine Lasermessung mit Pythagoras war nicht möglich, da helle Sonneneinstrahlung und das rote Backsteingebäude den Laserpunkt unsichtbar gemacht haben.

Methode		Traufe		First	
LD		5,743		9,796	
SM		6,5	+13 %	10,3	+ 5%
3D	2 Bilder	6,33	+10 %	8,79	+ 10 %
3D	3 Bilder	5,13	-11 %	8,93	- 9 %
	Mehrf. Ausf. d. Opt.				
3D	3 Bilder	5,81	+1,2 %	10,02	+2,3 %

Messung 3 (Stadtturm)

Der Stadtturm ist ein Gebäude mit Flachdach und daher in Kategorie I. Aufgrund der Höhe des Stadtturms (38 Meter⁶) war eine Messung mit der Laser-Methode nicht möglich.

Methode	First	%
SM	41,0	+ 7,9 %
3D	41,21	+ 8,4 %

⁶http://regiowiki.pnp.de/index.php/Stadtturm_%28Passau%29

Messung 4

Dieses Gebäude hat ein Walmdach und zwei Stockwerke. Damit gehört es in Kategorie II. Da bei der Berechnung viele Markante Punkte am Rand der zwei Bilder ausgewählt wurden, ist eine deutliche Abweichung von der Lasermessung festzustellen. Es ist für Messungen mit der 3D-Rekonstruktions-Methode darauf zu achten, dass das Gebäude mittig im Bild aufgenommen wird, damit die Verzeichnung das Ergebnis nicht so stark beeinträchtigt.

Methode	Traufe		First
LD	6,819		
LP	6,919	+ 1,5 %	
LP2	5,656	- 17 %	
SM	7,3	+ 7 %	
3D	4,49	- 34 %	6,06

Messung 5

Messgebäude 5 hat ein Satteldach und steht giebelseitig zur Straße. Damit ist es in Kategorie I und mit allen Methoden messbar. Weiterhin hat es vier Stockwerke und es wurde ein Stativ verwendet.

Methode	Traufe		First	
LD	6,635		11,930	
LP	6,858	+ 3,4 %		
LP2	6,942	+ 4,6 %		
SM	7,5	+ 13,0 %	13,5	+ 13 %
3D	6,818	+ 2,8 %	12,043	+ 1 %

Messung 6

Messgebäude 6 hat ein Pyramidendach und steht traufseitig zur Straße. Damit ist es in Kategorie II. Weiterhin hat es zwei Stockwerke und es wurde zwei Bilder für die 3D-Rekonstruktion gemacht.

Methode	Traufe		First	
LD	7,423			
LP	7,432	+ 0,1 %		
LP2	6,528	- 12,0 %		
SM	7,3	- 1,7 %		
3D	6,78	- 8,7 %	11,50	

6.6 Bewertung der Distanzmessung mit GPS

In einer ausführlicheren Studie preiswerter GPS-Sensoren wurde eine Abweichung der Messung im Zentimeterbereich erreicht. [17] Diese guten Messergebnisse waren aber nur möglich, wenn das Messgerät mehrere Minuten bis zu einer Stunde auf einem Stativ still stand und die Rohdaten des Sensors für die spätere Nachbearbeitung der Daten gespeichert wurden. Diese Methodik ist für die Anwendung der 3D-Rekonstruktion unpraktikabel und außerdem bietet die Android-Entwickler-Schnittstelle keinen Zugriff auf die Rohdaten, weshalb diese Technik mit der Standardsoftware nicht umsetzbar ist. Da für die 3D-Rekonstruktion eine genaue Abstandsmessung erforderlich ist, kann man aus Abschnitt 3.4.3 und der Studie folgern, dass die GPS-Sensoren, wie sie aktuell in

Smartphones verbaut werden (bzw. andere preiswerte Lösungen), noch nicht exakt genug messen. Die Messungen wurden daher mit einem Meterstab bzw. einer Stativ-Leiste gemacht.

6.7 Bewertung der Methoden

Die direkte Laser-Methode liefert die zuverlässigsten Ergebnisse. Dafür schneidet die Möglichkeit mit doppeltem Pythagoras nicht so gut ab: Bei Messung 4 und 6 ist eine deutliche Abweichung erkennbar. Diese kommen jedoch in der Bedienbarkeit nicht an die Smart-Measure-Methode heran, welche sich als benutzerfreundlichste Anwendung herausgestellt hat. Beide Methoden können mit der Anwendbarkeit der 3D-Rekonstruktion nicht mithalten. Da aber bei der 3D-Rekonstruktion zu viele Schritte beachtet werden müssen, ist der Aufwand bisher unzumutbar. Die Ergebnisse einer 3D-Rekonstruktion, mit mehr als zwei Bildern, motivieren hingegen eine künftige Weiterentwicklung dieser Methode. Eine Vereinfachung der Methode oder eine ausführliche Anleitung zur Bedienung sind dabei dringend notwendig.

7 Fazit

Die Abstandsmessung mit GPS hat sich als ungenügend herausgestellt hat, somit muss eine Alternative gefunden werden. Eine einfache Variante besteht in der Messung mit dem Meterstab. Eine andere vielversprechende Möglichkeit zeigt die Anwendung *Measured* auf: Die Strecke wird, einen Fuß vor den anderen setzend, zurückgelegt. Somit ist die Strecke ein Vielfaches der bekannten Schuhgröße.

7.1 Eintragung von Messungen bei OpenStreetMap

Zur Eintragung von Geodaten in OpenStreetMap wird der Java-OpenStreetMap-Editor⁷ (JOSM) verwendet. JOSM ist ein Bearbeitungsprogramm für OpenStreetMap, welches Ausschnitte der OpenStreetMap-Datenbank (Kartenmaterial) herunterladen kann und dem Benutzer Hilfsmittel zur Bearbeitung bietet. Gebäude werden in OpenStreetMap als zusammenhängende Wege dargestellt, die die Grundfläche des Gebäudes markieren.⁸ Die Höhe des Gebäudes wird im Schlüssel *height* eingetragen.⁹ Zusätzlich sollte die Quelle der Höhenmessung angegeben werden, damit Mapper erkennen können, dass es sich dabei um eine Messung handelt, die von der realen Höhe abweichen kann. Dadurch werden die Angaben den Anforderungen einer ständigen Qualitätskontrolle von OpenStreetMap gerecht. Vorschlag für die Quellenangabe der Messung aus den untersuchten Methoden:

- source:height=Laser-Measure-Method
- source:height=Smart-Measure-Application
- source:height=3D-Reconstruction

Ein Gebäude mit 10,5 Meter Höhe wird wie folgt eingetragen:

building=*	Beschreibung des Gebäudetyps
height=10.5	Höhenangabe in Meter (Punkt als Dezimaltrennzeichen)
source:height=3D-Reconstruction	Quellenangabe der Höhe

Tabelle 4: Höheneintragung

Eine Beschreibung der Messmethodik kann auch im Wiki von OpenStreetMap hinterlegt werden und mit einer Verknüpfung (URL) im *source*-Schlüssel vermerkt werden. Diese Möglichkeit macht andere Mapper auf die Methoden aufmerksam und wäre daher zu bevorzugen.

7.2 Ausblick

Im letzten Abschnitt dieser Arbeit wird auf die zukünftige Entwicklung der Anwendung eingegangen. Dabei lassen sich zwei Ziele unterscheiden: eine selbständige Smartphone-Anwendung und Verbesserungen der metrischen Rekonstruktion.

Smartphone-Anwendung

Für eine Smartphone-Anwendung, die auch eine Berechnung im 3D-Modell durchführen kann, müssen noch einige Dinge umgesetzt werden. Zunächst muss eine vollautomatische Extraktion und Verknüpfung der markanten Punkte ermöglicht werden, die bei verschiedenen Entfernungen der neuen Kameraposition (Stereobasis) zuverlässig funktioniert. An dieser Stelle sollte es nicht

⁷<http://josm.openstreetmap.de/wiki/De%3AWikiStart>

⁸http://wiki.openstreetmap.org/wiki/Simple_3D_Buildings

⁹<http://wiki.openstreetmap.org/wiki/Key:height>

notwendig sein am Smartphone markante Punkte korrigieren zu müssen, da die Eingabemöglichkeiten durch das kleine Display, im Vergleich zu einem Desktop-Rechner, eingeschränkt sind. Außerdem sollten ausreichend viele markante Punkte gefunden werden, sodass drei davon für eine Messung ausgewählt werden können. Für die Verknüpfung der markanten Punkte ist eine Verwendung der Lucas-Kanade-Methode in Betracht zu ziehen. Diese beschreibt eine Verfolgung markanter Punkte, welche ein Bild so transformiert, dass es mit einem Teil eines weiteren Bildes möglichst gut übereinstimmt. [13] Überdies hinaus sollte die Laufzeit der Optimierungsstrategien verbessert werden, um eine schnelle Messung am Smartphone zu erlauben.

Metrische Rekonstruktion

Für eine metrische Rekonstruktion sind eine Kamerakalibrierung und die Transformation der Kameraposition wichtig. Darüber hinaus können weitere beobachtete Merkmale im Bild verwendet werden, um das 3D-Modell zu verbessern. Parallele Linien in der Szene sollen im Modell ebenfalls parallel sein. Gebäude stehen in der Regel senkrecht, daher kann ein rechter Winkel am Fuß der Fassade angenommen werden. Das gleiche gilt für Fenster und andere im Bild auffindbare Merkmale. Hier bietet es sich an, eine Merkmalerkennung zur Verbesserung des 3D-Modells in Betracht zu ziehen.

Grundsätzlich ist die 3D-Rekonstruktion eine gute Möglichkeit Gebäudehöhen zu bestimmen. Außerdem stellt die Entwicklung weiterer Optimierungsstrategien zur Verbesserung des 3D-Modells eine tolle Herausforderung dar.

A Literaturverzeichnis

- [1] by Gary Bradski and Adrian Kaehler. *Learning OpenCV*. O Reilly, first edition edition, September 2008.
- [2] Smart Tools Co. Smart measure. <https://play.google.com/store/apps/details?id=kr.sira.measure>. zuletzt abgerufen am 18.02.2013.
- [3] OpenCV Community. Opencv documentation. <http://opencv.willowgarage.com/documentation/index.html>. zuletzt abgerufen am 01.12.2012.
- [4] OpenCV Community. Opencv documentation 3d reconstruction. http://opencv.willowgarage.com/documentation/camera_calibration_and_3d_reconstruction.html. zuletzt abgerufen am 01.12.2012.
- [5] B. Goetz and T. Peierls. *Java concurrency in practice*. Addison-Wesley, 2006.
- [6] Tobias Hanning. *Rechnersehen*. Ralf Schuster Verlag Passau, 1. auflage edition, 2010.
- [7] Tobias Hanning. Kamerakalibrierung und 3D-Rekonstruktion, 27. Oktober 2011.
- [8] Richard Hartley and Andrew Zisserman. *Multiple View Geometry*. Cambridge University Press, 2., aufl. edition, 2003.
- [9] Richard I. Hartley and Peter Sturm. *Triangulation*, 1994.
- [10] Tobias Knerr. OSM2World Webseite. <http://osm2world.org/>. zuletzt abgerufen am 16.12.2012.
- [11] H. Koepf and G. Binding. *Bildwoerterbuch der Architektur*. Kroener, 2005.
- [12] Robert Laganière. *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing, 2011.
- [13] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision (ijcai). In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, April 1981.
- [14] Mitglieder der OpenStreetMap-Community OpenStreetMap Foundation. [Openstreetmap.de. http://www.openstreetmap.org/](http://www.openstreetmap.org/). zuletzt abgerufen am 15.12.2012.
- [15] S. Graf Tobias Hanning, R. Schoene. A closed form solution for monocular re-projective 3d pose estimation of regular planar patterns. In: International Conference of Image Processing (ICIP) . Atlanta, GA, 2006. , E-ISBN : 1-4244-0481-9, 2006.
- [16] Frederik Ramm und Jochen Topf. *OpenStreetMap - Die freie Weltkarte nutzen und mitgestalten*. Lehmanns Media, 3. auflage edition, 2010.
- [17] Josua Staehli und Stefan Keller. Praezise positionsbestimmung mit low-cost-gps und post-processing. Technical report, Geometa Lab des Instituts fuer Software an der HSR Hochschule fuer Technik, Rapperswil (Schweiz), Fachhochschule Ostschweiz, February 2013.

B Abbildungsverzeichnis

1	OSM2World Kartenausschnitte	2
2	Beispiele für Gebäudetypen in Kategorie I	3
3	Beispiele für Gebäudetypen in Kategorie II	4
4	Lasermessung der Traufhöhe	5
5	Smart-Measure	6
6	Bildsequenz der Messdaten	6
7	Messung der Gebäudehöhe im 3D Modell	7
8	Lochkamera	8
9	$\{z = 1\}$ -Ebene ¹⁰	8
10	Schachbrett	11
11	Kreismuster	11
12	GPS Maus	13
13	Smartphone-GPS-Sensor	13
14	Kameratransformationen	14
15	Abbildungen T^* und T^{**} von der Kamera ins Objektkoordinatensystem	16
16	Bundle Adjustment	19
17	Distanzminimierung projizierter Punkte	20
18	Sichtstrahl-Distanzminimierung	22
19	ER-Diagramm Szene	24
20	Smartphone-Stativ-Adapter	25
21	Adapter auf Aluminiumleiste	25
22	Kamerakalibrierung	25
23	Punkte markieren und verknüpfen	26
24	Transformation und Triangulierung	27
25	Satteldach	37

C Tabellenverzeichnis

1	Verzeichnungsmodellierung der OpenCV-Bibliothek	10
2	Speedup	12
3	Anwendbarkeit der untersuchten Methoden	28
4	Höheneintragung	33

D Glossar

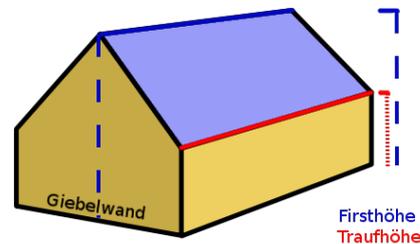


Abbildung 25: Satteldach

Android ist ein Betriebssystem für mobile Endgeräte (Smartphones). 23

Dachform Es gibt verschiedene Dachformen, welche nach der Anordnung der geneigten Dachflächen bestimmt sind. [11] In Abbildung 25 ist ein Satteldach abgebildet. Weitere Beispiele sind in Abschnitt 2.2 aufgeführt. 3

First First ist die obere, meist waagrechte Schnittlinie zweier geneigter Dachflächen [11] (vgl. Abbildung 25). 3

Firsthöhe Die Höhe des Firstes über dem umgebenden Terrain. 3

Flüssiglinse besteht aus verschiedenen Flüssigkeiten, die unter elektrischer Spannung die Brennweite variieren. Sie sind von sehr kleiner Bauform und energiesparsam und werden daher häufig in Smartphones eingebaut. 8

Giebelwand Bei einem Satteldach wird die Wand am First *Giebelwand* genannt (vgl. Abbildung 25). 3

Slippy Map ist ein moderner Begriff für eine verschiebbare, gekachelte Karte mit Zoom-Funktion im Web (engl. für flinke Karte). 1

SpeedUp paralleler Speedup beschreibt den Zusammenhang der Ausführungszeit eines sequentiellen und eines parallelen Programmes. Dies wird häufig als Bruchteil der Ausführungszeit der sequentiellen Version angegeben. 11

Traufe Die Traufe ist die untere waagrechte Begrenzung der Dachfläche, die parallel zum First verläuft [11]. 29

Traufhöhe Die Höhe der Traufe über dem umgebenden Terrain (vgl. Abbildung 25). 3

E Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt und durch meine Unterschrift, dass die vorliegende Arbeit von mir selbstständig angefertigt worden ist. Alle wörtlich oder sinngemäß übernommenen Ausführungen wurden als solche gekennzeichnet. Die Arbeit wurde bisher weder einer anderen Prüfungsbehörde vorgelegt noch veröffentlicht.

Passau, den 26.3.2013

Ort, Datum

(Reimar Stier)