



HOCHSCHULE MÜNCHEN

DIGITAL IMAGE PROCESSING

Image Stitching and Point Cloud Generation from Smartphone Pictures

Verfasst von

Adam Mahmoud

Michael Wimmer

Philipp Kronser

Vorgelegt bei

Prof. Dr. Claudius SCHNÖRR

27. Dezember 2018

Abkürzungsverzeichnis

BOM	Bill of Materials
BTC	Bitcoin
CAD	Computer-Aided Design
DApp	Dezentralisierte Apps
DRM	Design Research Methodology
ETH	Ether
ERP	Enterprise Ressource Planning
EVM	Ethereum Virtual Machine
PDM	Product Data Management
PLM	Product Lifecycle Management
PoS	Proof of Stake
PoW	Proof of Work
STEP	Standard for the exchange of product model data

Inhaltsverzeichnis

Abkürzungsverzeichnis	2
1. Einleitung	4
1.1. Motivation und Ziel der Arbeit	4
1.2. Aufbau der Arbeit	4
2. Deskriptive Phase I	5
2.1. Verwaltung von CAD-Daten heute	5
3. Section I	6
4. 3-D Punktewolke	7
4.1. Algorithmus	7
4.2. Kamerakalibrierung	8
4.3. Korrespondenzanalyse	10
4.4. Schätzen der Fundamentalmatrix	10
4.5. Schätzen der Kamerapose aus der Fundamentalmatrix	13
4.6. Triangulation	15
A. Appendix	16
Literaturverzeichnis	17

1. Einleitung

1.1. Motivation und Ziel der Arbeit

Lalala

1.2. Aufbau der Arbeit

Lalala

2. Deskriptive Phase I

2.1. Verwaltung von CAD-Daten heute

3. Section I

Lalala

4. 3-D Punktfolge

In dieser Aufgabe soll aus einer Reihe von Bildern eines Objekts, welche von verschiedenen Kamerastandpunkten aufgenommen wurden, eine 3D Punktfolge des Objekts berechnet werden.

4.1. Algorithmus

In diesem Kapitel wird der gesamte Algorithmus erklärt, der von der Bildserie zur 3D Punktfolge führt. Er enthält einige komplexe Unteralgorithmen die in den nächsten Kapiteln erläutert werden. Die grobe Vorgehensweise ist an **Mathworks** angelehnt. Zunächst müssen die Bilder des Objekts geladen werden. Dabei ist es wichtig dass sie in der richtigen Reihenfolge geladen werden, da aufeinanderfolgende Bilder eine Überlappung von mindestens 50% haben müssen. Außerdem müssen die Bilder in Grauwertbilder umgewandelt werden, da die darauffolgenden Algorithmen nur mit 1-kanaligen Bildern umgehen können. Zusätzlich müssen die Kameraparameter der Kamera, mit denen die Bilder aufgenommen wurden, geladen werden. Die Ermittlung der Kameraparameter wird in Kapitel 4.2 beschrieben. Nun wird das erste Bild verarbeitet, indem es über die Funktion `undistortImage` und die Kameraparameter verzeichnungsfrei (s. Kapitel 4.2) gemacht wird. Als nächstes werden auf dem ersten Bild markante Punkte über den Harris Corner Detector (s. Kapitel 4.3) ermittelt. Nun wird der erste Kamerastandpunkt, er ist der Referenzstandpunkt im Nullpunkt mit der Einheitsmatrix als Orientierung, mit- samt der markanten Punkte in eine Datenstruktur mit dem Namen `ViewSet` gespeichert. Diese speichert alle Parameter eines Kamerastandpunkts und ordnet diesen eine ID zu. Beim ersten Bild ist dies die ID 1. Es folgt eine Schleife die über alle Bilder iteriert. Mit dem nächsten Bild wird nun analog zum ersten Verfahren, es wird verzeichnungsfrei gerechnet und seine markanten Punkte werden ermittelt. Nun werden aber Punkte zwischen dem vorigen und dem gerade in der Schleife untersuchten Bild ermittelt die in beiden Bildern vorkommen. Solche Punkte werden als korrespondierende Punkte oder Matches bezeichnet. Die genaue Erläuterung der Berechnung von Matches folgt in Kapitel 4.3. Mit diesen ermittelten Matches kann nun eine sogenannte Fundamentalmatrix berechnet werden. Diese enthält Informationen über die intrinsischen und extrinsischen Kameraparameter. Das genaue Vorgehen zur Berechnung der Fundamentalmatrix sowie deren Eigenschaften werden in Kapitel 4.4 erläutert. Aus der Fundamentalmatrix kann durch Multiplikation der intrinsischen Kameraparameter beider Kameras (hier identisch

da gleiche Kamera) die Essentielle Matrix berechnet werden, die nur noch Informationen über die extrinsischen Kameraparameter enthält. Im nächsten Schritt kann nun aus der Essentiellen Matrix die Lage und Orientierung der Kameras zueinander ermittelt werden. (siehe Kapitel 4.5). Die berechnete Kamerapose sowie die Matches werden mit der nächsten ID in die ViewSet Datenstruktur gespeichert. Da zwischen den Bildern jeweils nur relative Posen berechnet werden, müssen diese zuvor noch mit der Position und Orientierung des vorigen Standpunkts verechnet werden. Im letzten Schritt innerhalb der Schleife wird nun das aktuelle Bild zum vorigen Bild, sodass zwischen diesem und dem nächsten wieder eine Korrespondenzanalyse sowie die Ermittlung der Kamerapose berechnet werden kann. Sind alle Kameraposen und deren Matches im ViewSet gespeichert kann nun eine Triangulation der Matches stattfinden. Dazu werden mit der Funktion findTracks Punkte ermittelt die in mehreren Bildern (mind. 2) auftauchen. Über diese Tracks wird eine Triangulation wie im Kapitel 4.6 beschrieben durchgeführt. Ergebnis sind Punkte im 3D Raum die zum Schluss noch geplottet werden. Das gesamte Vorgehen wird hier noch einmal im Pseudocode zusammengefasst.

```

1 Lade alle Bilder
2 Finde markante Punkte auf erstem Bild
3 Über alle Bilder:
4   Finde markante Punkte auf diesem Bild
5   Finde Matches zwischen vorigem und diesem Bild
6   Berechne Fundamentalmatrix aus Matches
7   Berechne relative Pose aus Fundamentalmatrix
8   Rechne relative Pose auf Pose von vorigem Bild
9   Dieses Bild wird voriges Bild
10  Trianguliere Punkte die in mehreren Bildern zu finden sind

```

Abbildung 1: Pseudocode

4.2. Kamerakalibrierung

Die Kalibrierung der Kamera ist wichtig, da die Abbildung jeder Kamera ein wenig anders ist. So unterscheiden sich zum Beispiel die Brennweiten von Gerät zu Gerät geringfügig und jede Kamera hat ihre eigenen Verzeichnungsfehler. Es gibt zwei Arten von Verzeichnungsfehlern, die tangentiale und die radiale. Die radiale Verzeichnung entsteht durch die Krümmung der Linse bzw. kleine Unebenheiten die bei der Produktion entstanden sind. Die Tangentiale Verzeichnung entsteht dadurch dass der Fotochip nicht exakt parallel zur Linse eingebaut wurde. Die Folgenden Bilder zeigen die Auswirkungen von tangentialer und radialer Verzeichnung. **Verlag**

Matlab bietet eine Applikation die aus Bildern von einem Kalibrierungsboard die Ka-

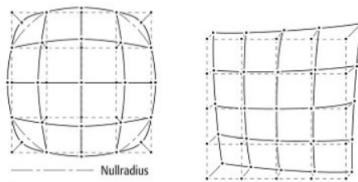


Abbildung 2: Auswirkung von Verzeichnung (links radial, rechts tangential) **Verlag**

meraparameter schätzt. Dazu muss das Board aus vielen verschiedenen Positionen (ca. 20) photographiert werden.

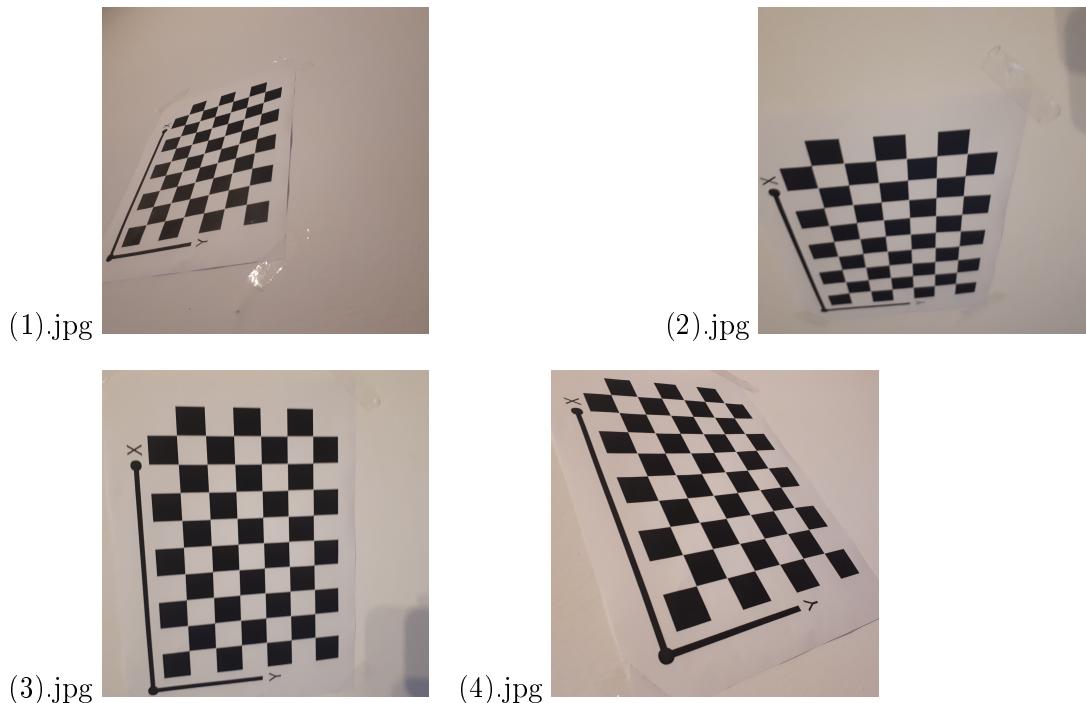


Abbildung 3: Bilder des Kalibrierungsboards

Aus diesen Bildern und der Information wie groß die Abstände der einzelnen Quadrate sind ermittelt die Applikation extrinsische und intrinsische Kameraparameter. Diese können als CameraParameters Objekt gespeichert und an anderer Stelle wieder geladen werden. Das CameraParameters Objekt enthält viele Informationen über die Kamera und die Kalibrierung. Von besonderer Bedeutung für diesen Einsatzweck sind hier die Einträge RadialDistortion und TangentialDistortion, welche für das Entzerren des Bildes wichtig sind. Außerdem wird die IntrinsicMatrix benötigt um aus der Fundamentalmatrix die Essentielle Matrix zu berechnen. **Mathworks**

4.3. Korrespondenzanalyse

4.4. Schätzen der Fundamentalmatrix

Die Fundamentalmatrix ist eine Matrix, welche die Geometrie zweier Kameräne zueinander sowie deren intrinsische Parameter beschreibt. Sie stellt die algebraische Repräsentation der Epipolargeometrie dar. **Richard2000**

Epipolargeometrie Die Epipolargeometrie liefert eine geometrische Beschreibung für zwei Kameraperspektiven auf einen Punkt in der Realität. Der Vorteil dieses Modells im Gegensatz zu anderen Modellen wie der Standardstereogeometrie ist, dass die Kameräne eine beliebige Brennweite haben können und sie nicht in einer Ebene liegen müssen. Dieses Modell passt also gut zur gestellten Aufgabe, da das Objekt von vielen unterschiedlichen frei positionierten Kamerastandpunkten aufgenommen wird. C und

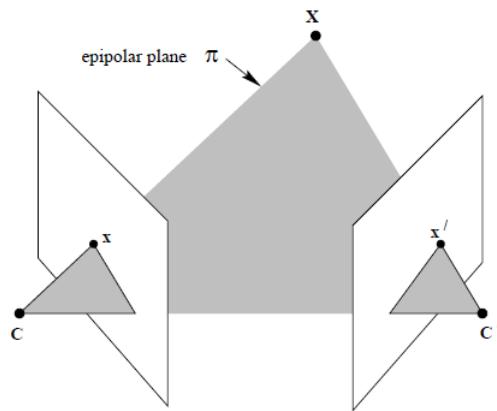


Abbildung 4: Epipolargeometrie **Richard2000**

C' sind hier die optischen Zentren der Kameräne. Sie blicken auf einen Punkt X in der realen Welt. Die Ebene die durch die Punkte C, C' und X aufgespannt wird nennt sich Epipolarebene. Die Punkte x und x', die auf die Bildebene projiziert werden, liegen ebenfalls auf dieser Ebene. Die Schnittlinien zwischen der Epipolarebene sowie den Bildebenden sind die Epipolarlinien. Korrespondierende Punkte müssen auf den jeweiligen Epipolarlinien liegen. Da ein anderer Punkt X eine andere Epipolarebene aufspannen würde, wären dessen Epipolarlinien auch andere. Die Schnittpunkte der Basislinie (C – C') mit der Bildebene sind die Epipole. Alle Epipolarlinien verlaufen durch die Epipole. **Richard2000**

Fundamentalmatrix Aus der Epipolargeometrie lässt sich die Fundamentalmatrix herleiten, die Informationen über die extrinsischen Kameraparameter (Position, Orientierung) sowie die Intrinsischen Parameter (Brennweite, Bildhauptpunkt) enthält. Sie stellt die algebraische Repräsentation der Epipolargeometrie dar. Die Fundamentalmatrix ist eine 3x3 Matrix und hat immer den Rang 2. Hat man einen korrespondierenden Punkt auf 2 Bildebenen x und x' , so muss mit der Fundamentalmatrix F gelten.

$$x'Fx = 0$$

Dies wird bei der Berechnung der Fundamentalmatrix noch wichtig. Sie wird über einen einen Datensatz an möglichen Punktkorrespondenzen über den 8-Punkte Algorithmus geschätzt. **Richard2000**

8-Punkte Algorithmus Der 8-Punkte Algorithmus ist der einfachste Algorithmus zum Schätzen der Fundamentalmatrix. Im speziellen wird hier der normalisierte 8-Punkte Algorithmus betrachtet, da die Normalisierung der Punkte zu besseren Ergebnissen führt. Bei der Normalisierung werden die Punkte so verschoben, dass ihr Mittelpunkt der Nullpunkt ist, und zusätzlich so skaliert, dass deren Abstand im quadratischen Mittel $\sqrt{2}$ vom Nullpunkt entspricht. Die Normierung führt dazu dass das zu lösende Problem des 8-Punkte Algorithmus wesentlich besser konditioniert ist.

Der 8-Punkte Algorithmus basiert auf der Eigenschaft:

$$[x, y, 1]F[x', y', 1] = 0$$

Ausgeschrieben entspricht dies folgendem Gleichungssystem:

$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0$ Diese Gleichung kann nun umgestellt werden sodass

$$Af = 0$$

mit

$$A = [x'x, x'y, x', y'x, y'y, y', x, y, 1]$$

und

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

Die Fundamentalmatrix hat trotz ihrer 9 Einträge lediglich 7 Freiheitsgrade, da sie nur bis auf ihren Skalierungsfaktor definiert ist und die Bedingung an sie gestellt wird dass sie den Rang 2 aufweist. Daher kann mit mindestens 8 korrespondierenden Punktpaaren über die Singulärwertzerlegung eine Lösung für f gefunden werden. Diese ist der Singulärvektor der zum kleinsten Singulärwert von A korrespondiert. Die Singulärwerte

befinden sich in der Diagonalmatrix D. Zerlegt man A mittels Singulärwertzerlegung in UDV, so ist die Lösung in der letzten Spalte von V zu finden. Für eine ideale Fundamentalmatrix müsste nach einer Singulärwertzerlegung gelten:

$$D_{ideal} = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Dies wird in der Realität nicht vorkommen, da hier mit diskreten Pixeln gearbeitet wird und so nie die ideale Fundamentalmatrix berechnet werden kann.

$$D_{real} = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix}$$

Daher wird der dritte Diagonaleintrag von D auf 0 gesetzt und die Fundamentalmatrix aus den drei Matrizen aus der Zerlegung erneut berechnet. **Richard2000 Schreer2005**

Schätzen der Fundamentalmatrix mittels RANSAC Um eine möglichst gute Fundamentalmatrix zu erhalten, wird der RANSAC Algorithmus verwendet. Er nimmt sich aus der Menge der korrespondierenden Punktpaare 8 zufällige Punkte heraus. Damit wird über den 8-Punkte Algorithmus eine Fundamentalmatrix berechnet. Um die Güte dieser Fundamentalmatrix zu prüfen wird nun ermittelt wie viele korrespondierende Punktpaare zu der berechneten Fundamentalmatrix passen. Dazu wird die Sampson Distanz ermittelt, welche sich wie folgt berechnet.

$$d = \frac{(x'^T F x)^2}{(F x)_1^2 + (F x)_2^2 + (F x')_1^2 + (F x')_2^2}$$

$(F x)_j^2$ beschreibt hier den j-ten Eintrag im Vektor $(F x)^2$.

Liegt ein korrespondierendes Punktpaar unter einem gewissen Schwellwert, zählt er als Inlier und gilt als zum Modell passend. Diese Berechnung wird über viele Iterationen mit immer zufällig ausgewählten Punkten für die Berechnung der Fundamentalmatrix wiederholt. Die Fundamentalmatrix die die meisten Inlier hat, deren Modell also am besten zu den Daten passt wird jeweils gespeichert. Wurden alle Iterationen durchlaufen, wird die Fundamentalmatrix erneut über den 8-Punkte Algorithmus mit allen Inliern der besten Fundamentalmatrix berechnet. **Richard2000 Mathworks**

```

1 Über N Iterationen:
2   Wähle 8 zufällige Punkte aus Korrespondenzen
3   Berechne F über 8-Punkte Algorithmus
4   Berechne Sampson-Distanz für alle Korrespondenzen
5   Berechne Inlier (Sampson-Distanz < thresh)
6
7   Falls Anzahl Inlier > Anzahl Inlier zuvor
8     Speichere F temporär ab
9     Speichere Inlier temporär ab
10 end
11
12 Berechne F mit den temporär gespeicherten Inliern
13 return F

```

Abbildung 5: Pseudocode RANSAC für Fundamentalmatrix

4.5. Schätzen der Kamerapose aus der Fundamentalmatrix

Um aus der Fundamentalmatrix die Kamerapose zu schätzen, muss diese zunächst in die Essentielle Matrix umgewandelt werden. Diese steht in enger Verbindung zur Fundamentalmatrix, allerdings hat sie nur 5 Freiheitsgrade, nämlich 3 für die Rotation und 2 für die Translation (auch hier ohne Skalierungsfaktor). Um aus der Fundamentalmatrix die Essentielle Matrix zu berechnen, muss diese mit der Intrinsischen Matrix des CameraParams Objekt, welches aus der Kalibrierung generiert wurde, multipliziert werden. Die Intrinsische Matrix enthält die Informationen über die Brennweite sowie den Bildhauptpunkt. Da hier alle Bilder mit der gleichen Kamera aufgenommen wurden, wird die Fundamentalmatrix zwei mal mit der gleichen intrinsischen Matrix multipliziert.

$$E = I' * F * I$$

Zunächst soll die essentielle Matrix E in die schiefsymmetrische Matrix S und die Rotationsmatrix R zerlegt werden. Hierfür wird E mittels Singulärwertzerlegung in die Matrizen U,D und V zerlegt.

$$E = UDV$$

Zusätzlich werden die Matrizen W und Z benötigt:

$$W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$Z = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Dabei ist W eine Rotationsmatrix und Z schiefsymmetrisch. Hier ergeben sich nun 2 mögliche Lösungen für die Rotationsmatrix:

$$R_1 = UW'V'$$

$R_2 = UWV'$ Beide Lösungen sind gültig und müssen später verifiziert werden. Der Translationsvektor lässt sich einfach aus der dritten Spalte von U ablesen oder durch die Formel:

$t = UZU'$ ermitteln. Aufgrund des fehlenden Skalierungsfaktors könnte die mögliche Lösung aber auch

$$t = -t$$

sein. Daher ergeben sich für die Kamerapose nun 4 mögliche Lösungen:

$$R_1, t$$

$$R_1, -t$$

$$R_2, t$$

$$R_2, -t$$

Um zu prüfen welche Kameraposen die richtigen sind wird nun für jede Kombination eine Triangulierung der Inlier aus der Fundamentalmatrix Berechnung durchgeführt. Die erste Kamera befindet sich dabei im Nullpunkt mit der Orientierung $\text{eye}(3)$, die zweite hat jeweils eine der vier möglichen Lösungen. Die genaue Vorgehensweise der Triangulation wird im nächsten Kapitel beschrieben. Bei der der Realität entsprechende Lösung, müssten (fast) alle in den 3D Raum triangulierten Punkte vor beiden Kameras liegen, bei den falschen Lösungen müssten sie hinter mindestens einer Kamera liegen. Zum Schluss werden die als richtig ermittelte Lösung noch in das Koordinatensystem der ersten Kamera überführt, indem die Rotationsmatrix transponiert wird und diese auf den Translationsvektor multipliziert wird. Dann werden sie als Pose der zweiten Kamera relativ zur ersten zurückgegeben. **Richard2000 LTH2013**

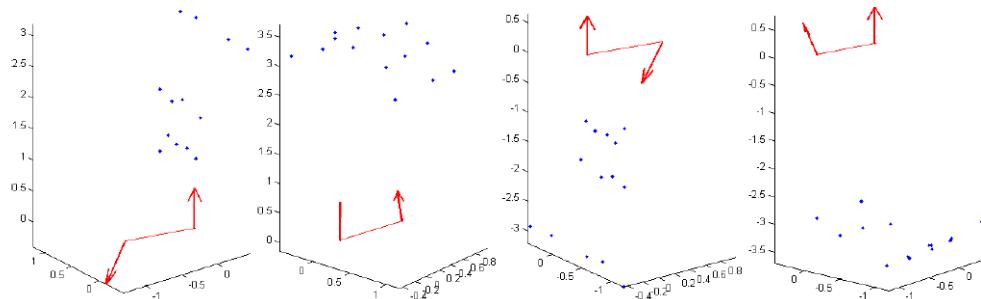


Abbildung 6: 4 mögliche Lösungen **LTH2013**

4.6. Triangulation

Die Triangulation ist eine Technik die aus Bildpunkten auf der Bildebene zweier Kameras Punkte im 3D Raum errechnet. Dafür wird eine Kameramatrix oder Projektionsmatrix benötigt. Sie ist eine 4x3 Matrix und enthält die interenen und externen Kameraparameter einer Kamera. Sie beschreibt die Projektion eines Punktes in der realen Welt auf die Bildebene der Kamera.

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix}$$

Hier wird die Datenstruktur CameraMatrix von Matlab verwendet, die aus den Kameraparametern sowie den zuvor berechneten Kameraposen die Kameramatrix errechnet.

Für die Triangulation wird die Form

$$x = PX$$

umgestellt in

$$AX = 0$$

A wird nach folgender Vorschrift aufgebaut:

$$A = \begin{pmatrix} xp^{3T} - p^{1T} \\ yp^{3T} - p^{2T} \\ x'p'^{3T} - p'^{1T} \\ y'p'^{3T} - p'^{2T} \end{pmatrix}$$

Mit p^i wird hier die i-te Zeile der Kameramatrix P bezeichnet. Die Punkte x,y und x',y' sind die korrespondierenden Bildpunkte in Pixeln. Wird nur zwischen zwei Kameras trianguliert, so kann A genau wie in der Formel erstellt werden. Wird zwischen mehreren Kameras trianguliert, wie am Ende des Gesamtalgorithmus so kann A analog zur oberen Formel aufgebaut werden, mit jeweils 2 weiteren Zeilen für jede weitere Kamera. Die Lösung für X wird über die homogene Methode ermittelt, indem A mittels Singulärwertzerlegung wieder in die 3 Matrizen U,D und V zerlegt wird. Die Lösung ist hier wieder korrespondierend zum kleinsten Singulärwert und daher, bei absteigender Größe der Singulärwerte in D, in der letzten Spalte von V zu finden. **Richard2000**

$$A = UDV$$

$$X = V(end)$$

A. Appendix

Erklärung

Hiermit erkläre ich gemäß der Rahmenprüfungsordnung der Hochschule München, dass ich die vorliegende Arbeit mit dem Titel “Implementierung einer Blockchain-Anwendung zur Steigerung der Integrität von CAD-Metadaten” selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

München, den _____

Unterschrift _____