



Diplomarbeit am Institut für Informatik der Freien Universität Berlin,

Arbeitsgruppe Intelligente Systeme und Robotik

Erstellung von Orthophotos anhand visueller Merkmale

Hannes Priehn

Matrikelnummer: 3978590

hpriehn@inf.fu-berlin.de

Betreuer: Prof. Dr. Marco Block-Berlitz, Prof. Dr. Raúl Rojas

Abgabedatum: 22. April 2014

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis oder ggf. in Fußnoten angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den 7. April 2014

Hannes Priehn

Inhaltsverzeichnis

1	Motivation und Einleitung	5
1.1	Motivation	5
1.2	Das Projekt	6
1.3	Die archäologische Arbeit	6
2	Grundlagen und Stand der Technik	9
2.1	Systemvoraussetzungen	9
2.2	Physik und Hardware	9
2.2.1	Multicopter	10
2.2.2	Der Weg des Lichts	11
2.2.3	Verzeichnung	12
2.3	Geometrie	13
2.3.1	Das Lochkameramodell	13
2.3.2	Homogene Koordinaten	14
2.3.3	Epipolargeometrie und Posenschätzung	17
2.4	Computer-Vision und Algorithmen	18
2.4.1	Merkmale	18
2.4.2	Random Sample Consensus	19
2.4.3	Bündelblockausgleich (Bundle Adjustment)	21
2.4.4	Multiview Stereo	23
2.4.5	Stitching	23
2.4.6	Farbtransfer	24
2.5	Verwendete APIs und Software	26
2.5.1	VisualSFM	26
2.5.2	Meshlab	27
2.6	Das Problem aus Sicht der Kartographen	27
3	Entwurf	30
3.1	Die Ausgangssituation	30

Inhaltsverzeichnis

3.2	Hardwarelimitierungen	31
3.3	Softwarebeschränkungen	32
3.4	Die Projektion	32
3.5	Gigapixelgroße Bilder	33
3.6	Farben	34
4	Implementierung	37
4.1	Die Kartenebene	37
4.2	GPS Impfung der Bilddaten	38
4.3	NVM Datei	38
4.4	Programmierte Module	39
4.4.1	NVMProject	39
4.4.2	Chopper	41
5	Leistungsbewertung	46
5.1	Projektion und Geometrie	46
5.2	Güte der erstellten Karten	47
5.3	Visueller Eindruck	48
6	Schlußfolgerungen, Fragen und Ausblicke	52
6.1	Die Geometrie	52
6.2	Der Farbausgleich	52
6.3	Geocoding	53
6.4	Echtzeiteinsatz	53
7	Zusammenfassung	55
	Literaturverzeichnis	56

1 Motivation und Einleitung

Die Arbeit ist in sieben Kapitel gegliedert. Das erste skizziert die Gründe für die Arbeit und das Umfeld in dem sie entstanden ist. Grundlagen und Stand der Technik stellt die Verbindung zu verwandten Arbeiten her. Außerdem werden die Systemvoraussetzungen und die zum Verständnis notwendigen physikalischen, mathematischen und algorithmischen Grundlagen erklärt. Im Kapitel Entwurf werden die Entscheidungen aufgegriffen, welche auf dem Weg zur Implementierung gefällt wurden. Das darauffolgende Kapitel Implementierung widmet sich den Lösungen, der im Kapitel Entwurf vorgestellten Problemen. Außerdem werden im Entwurf nicht-antizipierte Probleme und deren Lösungen behandelt. Eine Betrachtung der erreichten Ergebnisse findet im Kapitel Leistungsbewertung statt. In Schlussfolgerungen und Ausblicke werden zum Abgabzeitpunkt unerreichte Ziele sowie Ideen zur Verbesserung angesprochen. Das letzte Kapitel beinhaltet eine stichpunktartige Zusammenfassung der geleisteten Arbeiten.

1.1 Motivation

Der Anwendungsbereich von Karten ist groß. Exakte Karten sind aus der heutigen Gesellschaft kaum mehr wegzudenken. Dank genauer Karten ist es heutzutage möglich, Waren und Menschen von einer Hälfte der Erde auf die andere zu transportieren, komplexe Bauvorhaben zu meistern und die Welt massiv umzugestalten. Heutzutage unterstützen Orthophotos Menschen auf allen gängigen Online-Kartendiensten. Sie vermitteln den Eindruck der Umgebung genauer als eine traditionelle Karte. Sie können auch automatisch auf Karteninformationen hin durchsucht werden [1].

War es noch vor einigen Jahrzehnten undenkbar, dass Privatpersonen die Investitionsschwelle für Luftaufnahmen schultern können, ist es heutzutage, dank dem Einzug von Digitalkameras und Multicoptern, preiswerter und einfacher denn je. Um den Materialaufwand so minimal wie möglich zu halten und den Einsatz und die Kosten von Spezialkomponenten zu vermeiden, aber auch da es schlichtweg manchmal einfach nicht möglich ist eine genaue Position der Kamera zu

messen (z.B. bei historischen Aufnahmen, Fehlfunktion der GPS Hardware), widmet sich diese Arbeit der Erstellung von Orthophotos ausschließlich anhand von Bildmaterial. Hierfür wurden sogenannte Multicopter und handelsübliche Digitalkameras verwendet. Durch den Einsatz von Computer-Vision-Algorithmen sollen aus Bilddaten Aufsichten der Szenen produziert werden können.

1.2 Das Projekt

Diese Arbeit wurde im Rahmen des Projektes Archäocopter¹ geschrieben. Das Projekt möchte Informationstechnologien für die Archäologie nutzbar machen. Im Zuge des Projektes werden geeignete Hard- und Softwarewerkzeuge entwickelt, um die Qualität der Dokumentation von Ausgrabungen zu verbessern und den Zeitaufwand zu verringern. Liegen diese Informationen digital vor, eröffnen sich etliche Möglichkeiten der weiteren Nutzung:

- nachträgliche virtuelle Begehungen der Ausgrabungsstätte
- dauerhafte und verlustfreie Archivierung und Vervielfältigung
- teilen der Modelle über das Internet mit Menschen in aller Welt
- Zugänglichkeit der Daten für Simulationen um Rückschlüsse auf das damalige Leben zu ziehen (z.B. Sonnenverlauf in einem Tempel zu verschiedenen Jahreszeiten)

Doch nicht nur die gesammelten Daten, sondern auch die archäologische Arbeit soll verändert werden. Das Verfahren der Dokumentation mithilfe des Archäocopters ist in Abbildung 1.1 gezeigt.

1.3 Die archäologische Arbeit

Nicht alle archäologischen Ausgrabungsstätten werden gezielt gesucht, erlangen Berühmtheit und werden für die Öffentlichkeit zugänglich konserviert. Die Realität sind meistens sogenannte Not- oder Rettungsgrabungen. Hierbei liegt die Stätte auf einem Grundstück, auf dem Baumaßnahmen geplant sind. Häufig sind es sogar die Bauarbeiter, welche bei den schon begonnenen

¹<http://www.archaeocopter.de>

1 Motivation und Einleitung

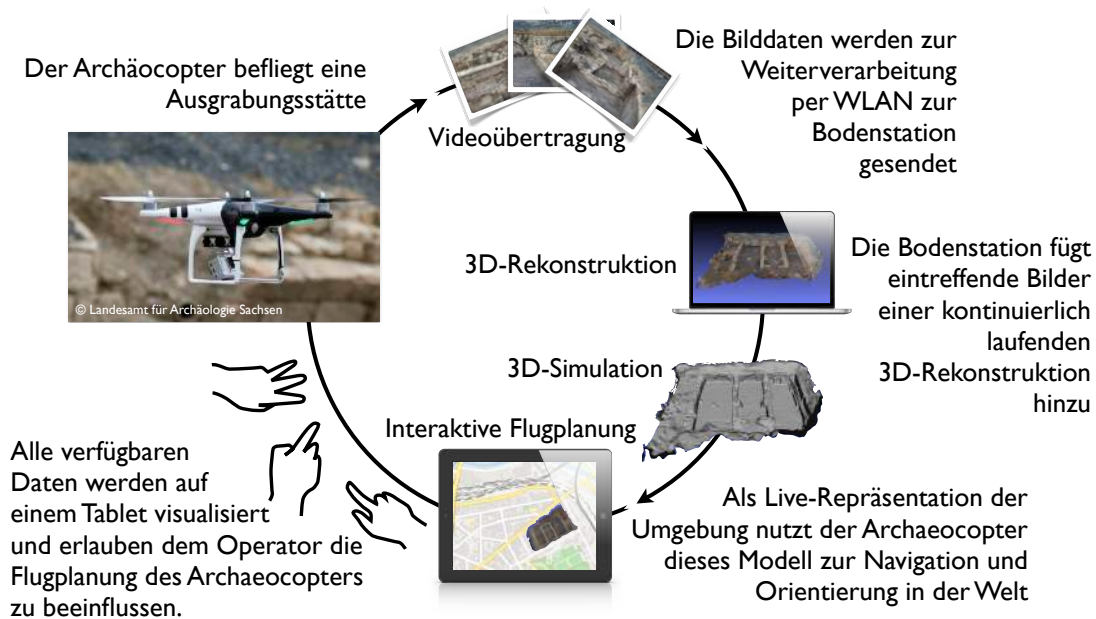


Abbildung 1.1: Wie sich die Zukunft der Dokumentation einer Ausgrabungsstätte innerhalb des Projekt Archäocopters vorgestellt wird

Arbeiten die Überreste entdecken. Unter diesen Umständen soll nun mit kleinstmöglichem Zeitaufwand bestmöglich dokumentiert werden. Die Dokumentation einer archäologischen Ausgrabung sollte akribisch genau erfolgen, denn das Vorhanden- bzw. Nichtvorhandensein von Details der Ausgrabung können für zukünftige Funde relevant sein. Meistens verfällt sogar die Gelegenheit eine Ausgrabung ein zweites Mal durchzuführen, da schon ein Gebäude über der Ausgrabungsstätte gebaut oder diese sogar demontiert wurde. Fotos reichen zur Dokumentation bei weitem nicht aus. Eine gute Dokumentation enthält deswegen noch viele zusätzliche Informationen, wie zum Beispiel Abgleiche der dokumentierten Sedimente und Gegenstände mit einer genormten Farbtabelle sowie gezeichnete Profile und Plana in Maßstäben von 1:10 oder 1:20². Die Qualität dieser Zeichnungen ist vom Können und der Zeit des Archäologen abhängig. Auch eine geocodierte Karte und die Positionen der Funde innerhalb der Ausgrabungsstätte dürfen nicht fehlen.

Obwohl frühere Häuser und Strukturen unter Erdschichten begraben aus der Luft nicht erkennbar sind, hinterlassen sie aus der Luft erkennbare Spuren. Reste der Bebauung im Boden beeinflussen die Wasser- und Nährstoffversorgung von Pflanzen. An der Art der Pflanzen, ihrem Wuchs-, Blüh- und Fruchtungsverhalten können so Rückschlüsse auf die unterliegende Bebau-

²<http://www.praehistorische-archaeologie.de/wissen/grabungen/ausgrabung/arbeitsgaenge/>

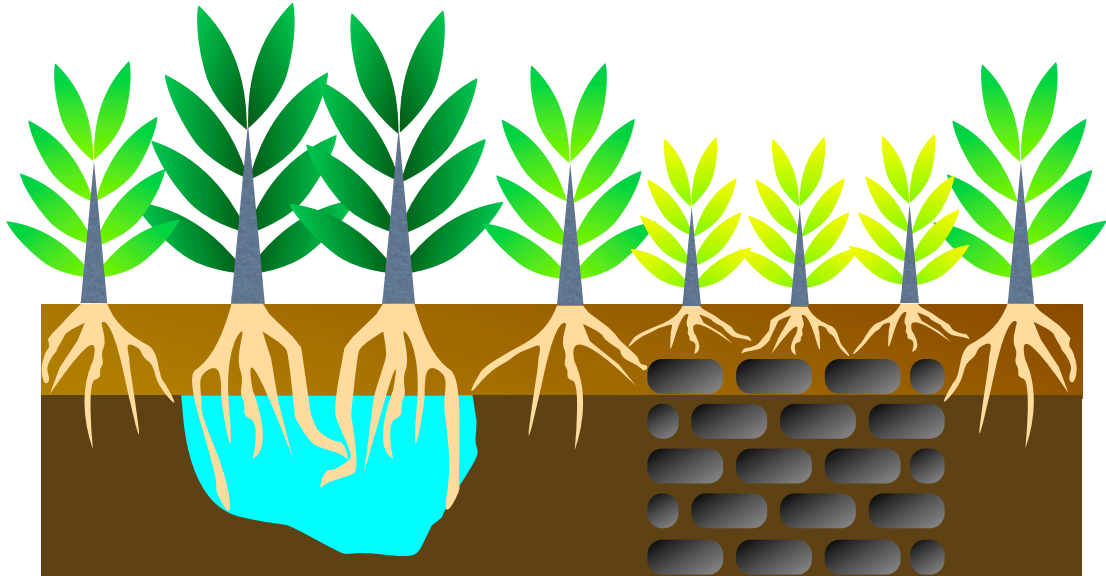


Abbildung 1.2: Ein Querschnitt durchs Erdreich. Unter der Erde liegende Reste und Spuren von Bebauung beeinflussen das Wachstum der darüberliegenden Vegetation. Aufgefüllte Gräben (links) können die Wasserversorgung verbessern, Mauerwerk oder Fundamente (rechts) hemmen das Wachstum.

ung gezogen werden (siehe Abbildung 1.2). Diese Technik ist so populär geworden, dass man ihr ein eigenes Gebiet innerhalb der Archäologie, die Luftbildarchäologie, gewidmet hat und die Landesämter eigene Flugzeuge dafür bereithalten. Da die Unterschiede im Äußeren der Pflanzen mitunter nur sehr subtil und deshalb besonders beim Wechsel der Jahreszeiten oder einer Folge von Trockentagen nur für kurze Zeit zu erkennen sind, ist es essentiell schnell zu reagieren, um den besten Zeitpunkt nicht zu verpassen³. Der Archäocopter ist dafür ideal. Mit seiner geringen Größe passt die Ausrüstung in einen Rucksack (siehe Abbildung (2.1)) und kann damit den Archäologen bis an die entlegensten Orte begleiten.

³<http://www.archaeologie.sachsen.de/532.htm>

2 Grundlagen und Stand der Technik

Das folgende Kapitel widmet sich der Klärung der Systemvoraussetzungen sowie der Erklärung der zum Verständnis der Arbeit nötigen Grundlagen. Spätere Kapitel verweisen auf die einzelnen Abschnitte.

2.1 Systemvoraussetzungen

In dieser Arbeit werden Luftbilder von Szenen verwendet. Es hierbei prinzipiell nicht von Belang, wie die Aufnahmen zustande kamen. Sie können sowohl von einer an einer Angel montierten Kamera, als auch von Fluggeräten stammen. Die verwendeten Bilder reichen in der Auflösung von 1080p bis zu 20 Megapixeln. Obwohl die Bilder der verwendeten Datensätze jeweils mit derselben Kamera in derselben Auflösung aufgenommen wurden, funktioniert das vorgestellte Verfahren prinzipiell auch mit kombinierten Datensätzen aus verschiedenen Kameras und Auflösungen. Die einzig unumstößliche Voraussetzung ist, dass die Bilder genügend überlappende Bildbereiche besitzen und sich die abgebildete Szene nicht zu stark ändern darf. Dies ist ein gar nicht so trivialer Punkt. Gebiete können sich schneller optisch ändern als gedacht. Schon innerhalb eines Tages können Gerüste an Häusern aufgebaut, Rapsfelder erblühen oder Schnee fallen und damit die optische Zuordnung für den Computer erschweren.

2.2 Physik und Hardware

In der Computer-Vision ist es essentiell einen Überblick über die Entstehung der Bilder, die verarbeitet werden sollen, zu besitzen. Deshalb erklärt dieser Abschnitt die für die Arbeit wichtigsten Grundlagen von Kameras, sowie wie der verwendeten Hardware.



Abbildung 2.1: Links: Der Archäocopter-Koffer fasst die gesamte nötige Ausrüstung. Rechts: Der Archäocopter im Flug bei einer Vorführung auf der langen Nacht der Wissenschaften in Dresden.

2.2.1 Multicopter

Multicopter (siehe Abbildung 2.1) sind unbemannte Luftfahrzeuge, welche über mehrere Propeller angetrieben werden. Da sie ihren Auftrieb nicht über Tragflächen gewinnen, sind sie nicht wie Flugzeuge auf eine ständige Vorwärtsbewegung angewiesen. Deshalb benötigen Multicopter auch keine klassischen Start- und Landebahnen. Multicopter können eine feste Position in der Luft halten und in jede Raumrichtung beschleunigen. Ihr Flugverhalten ähnelt dem eines Kolibris. Bausätze oder fertige Multicopter können zum Beispiel im Internet über zahlreiche Seiten bestellt werden. Sie sind aufgrund der geringen Größe und des Elektroantrieb (es gibt natürlich auch Modelle mit Treibstoffmotoren) in geschlossenen Räumen einsatzfähig¹.

Seit dem Aufkommen von Multicoptern ist die Investitionsschranke für Luftaufnahmen nur wenige hundert Euro hoch. Gepaart mit Satellitennavigation bieten sie eine attraktive und kostengünstige Alternative zu Aufklärungsflugzeugen. Multicopter können Aufnahmen in Bodennähe tätigen und damit Strukturen besser aufgelöst fotografieren als höher fliegende Fluggeräte. Durch die niedrige Flughöhe werden dadurch keine Verdeckungen durch Wolken aufgezeichnet. Die Antriebe eines Multicopters erzeugen Abwinde, die gering genug sind, dass die mitunter empfindlichen Ausgrabungsstätten beim Überflug nicht beschädigt werden. Jedoch sind sie auch leistungsfähig genug um den Multicopter in höhere Gefilde zu transportieren². Abstürze sind aufgrund der geringen Größe und Masse eines Quadrocopters weniger fatal und da der Multicopter ferngesteuert wird, werden in der Regel keine Menschenleben gefährdet.

Es gibt jedoch auch Nachteile. Die geringe Motorisierung kann starken Winden nicht entgegen

¹<http://www.flyingmachinearena.org/>

²<http://www.technik-consulting.eu/index.php?cat=Analyse&page=Quadrocopter>

wirken und ermöglicht es nur kleine und leichte Ausrüstung zu transportieren. Die Antriebe eines Multicopters versetzen ihn in Schwingung. Werden diese Schwinungen auf die Kamera übertragen, verursachen sie durch das nicht simultane Auslesen der Detektoren auf dem Sensor sogenannte Rolling-Shutter Effekte. Deshalb sollten Kameras auf Multicoptern grundsätzlich mit einer physikalischen Dämpfung montiert werden.

2.2.2 Der Weg des Lichts

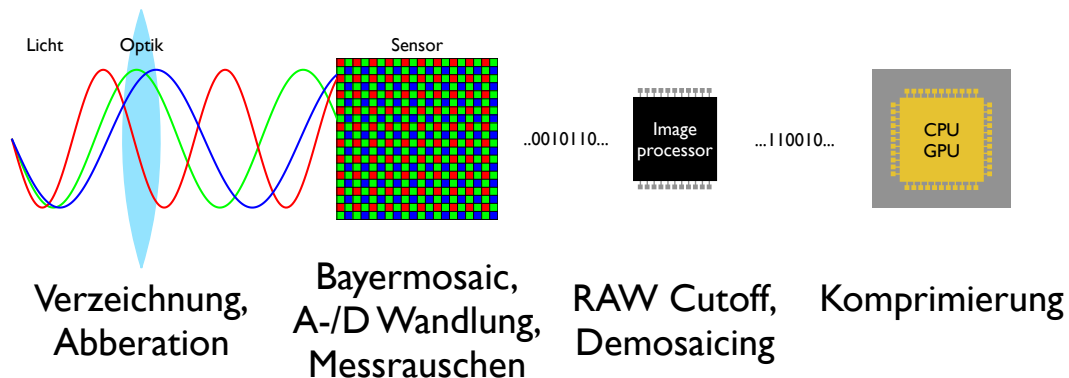


Abbildung 2.2: Die Fehlerquellen auf dem Weg des Lichts in den Computer

Eine Kamera misst die von einer Szene ausgestrahlten Lichtintensitäten, genannt Irradianzen. Auf dem Weg in eine für den Computer verständliche Form passiert das Licht mehrere Stationen, welche die Qualität der endgültigen Information auf verschiedene Weise beeinflussen. In Abbildung 2.2 ist dieser Weg dargestellt. Als erstes durchläuft das Licht die Optik, welche geometrische und farbliche Verzerrungen erzeugen kann. Nach der Optik, trifft das Licht auf den Sensor. Die Oberfläche dieser Sensoren ist wiederum mit vielen kleinen Detektoren besetzt, welche ihr elektrisches Verhalten je nach Stärke des einfallenden Lichts ändern. Im Sensor wird dieser Wert gemessen und von einem Analog-/Digitalwandler in einen Digitalwert umgewandelt. Hierbei treten Messfehler und Artefaktbildung durch die Anordnung der Detektoren (Bayer-Muster) auf der Sensoroberfläche auf [2, Abschnitt 2.1].

Manche Kameras können schon diesen unangetasteten Wert speichern. Sie werden in herstellerspezifischen RAW-Formaten abgelegt. Diese Daten enthalten oft etwas mehr Information als normale 8-Bit Pixeldaten, denn zum einen arbeiten die Sensoren häufig intern mit einer höheren Anzahl von Bits, als die im RGB-Format üblichen 8-Bit pro Kanal und zum anderen werden die Messwerte nicht durch die Impulsantwortfunktion (*Camera Response Function*) beeinflusst. Hierbei wird der Farbraum nichtlinear verändert: Dunkle Messwerte werden nachbelichtet und

helle abgewedelt, um den Gesamteindruck für menschliche Betrachter zu verbessern, da die Detektoren die Lichtintensität linear aufzeichnen, unsere Wahrnehmung jedoch logarithmisch funktioniert [3].

Es existieren viele verschiedene Farbmodelle. In dieser Arbeit wird der RGB-Farbraum benutzt. Er besteht aus einem Würfel, dessen Koordinaten Achsen den Rot, Grün und Blauanteil angeben. (r, g, b) mit $r, g, b \in [0, 1]$. Der in Abbildung 2.3 dargestellte RGB-Farbraum ist kein Vektorraum im mathematischen Sinne. Da durch Addition oder Vektorprodukt Bereiche außerhalb des Würfels erreicht werden können, fehlt ihm die Abgeschlossenheit. Deshalb werden in so einem Fall die Werte einfach beim Maximalwert abgeschnitten, was den Nachteil von Informationsverlust hat [4].

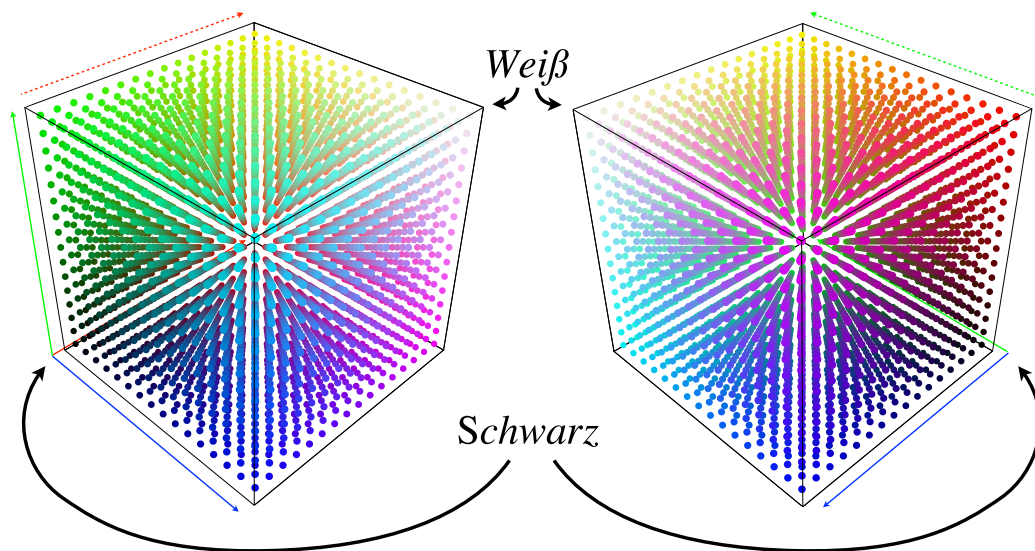


Abbildung 2.3: 3D-Histogramme des RGB-Farbwürfels. Außerhalb des Würfels sind Farben undefiniert. Die Diagonale von Schwarz nach Weiß enthält alle Graustufen.

2.2.3 Verzeichnung

Manche Bauteile eines optischen Systems brechen/beugen Lichtstrahlen und beeinflussen damit die Belichtungsposition auf der Sensoroberfläche in nichtlinearer Weise. Dieser Umstand wird Verzeichnung genannt. Die Objektive normaler Kameras verzeichnen stetig und am stärksten radial [2, Seite 334]. Bei dieser Art der Verzeichnung werden Punkte auf Kreisbögen um ein Zentrum auf Punkte auf Kreisbögen des Zentrums mit unterschiedlichen Radien abgebildet.

Die Überföhrungsfunktion dieser Radien kann durch die unendliche Summe des Produktes der Koeffizienten K_n und des Radius r

$$r_{undist} = 1 + \sum_{n=0}^{\infty} K_n r^{2n} \quad (2.1)$$

angenähert werden [5]. In der Praxis reichen bereits geringe n aus, um ein ausreichend genaues Ergebnis zu erreichen³.

Eine weitere Verzeichnung ist die Tangentialverzeichnung. Sie wird durch die nicht exakt parallele Stellung des Sensors zur Optik verursacht. Punkte auf der Tangente eines Radius werden auf Punkte der Tangente eines anderen Radius abgebildet. Die Überföhrungsfunktion der Tangenten kann durch die beiden Koeffizienten P_1 und P_2 charakterisiert werden [5]. Die Gleichung aus Formel (2.1) wird für x und y getrennt erweitert auf:

$$\begin{aligned} x_{undist} &= x + (P_1(r^2 + 2x^2) + 2P_2xy) \sum_{n=1}^{\infty} K_n r^{2n} \\ y_{undist} &= y + (2P_1xy + P_2(r^2 + 2y^2)) \sum_{n=1}^{\infty} K_n r^{2n} \end{aligned} \quad (2.2)$$

Um die Auswirkungen und Unterschiede von Radial- und Tangentialverzeichnung zu verdeutlichen, werden in Abbildung 2.4 einige Beispiele gezeigt.

2.3 Geometrie

Wer 3D-Rekonstruktion betreiben möchte, kommt um Geometrie nicht umher. Die Forschung in der Computergrafik hat in den letzten Jahren etliche mathematische Werkzeuge und Hardware entwickelt, um immer realistischere Bilder zu erzeugen. Obwohl das Problem ein 3D-Modell aus 2D-Bilddaten zu erzeugen das zur Computergrafik inverse Problem ist, besitzt die Mathematik der Computergrafik einen großen Überlappungsbereich mit der Computer-Vision.

2.3.1 Das Lochkameramodell

Die geometrischen Gegebenheiten einer Kamera werden durch das sogenannte Lochkameramodell beschrieben. Dieses Kameramodell zeichnet sich durch seine Verzerrungsfreiheit aus. Das

³http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

2 Grundlagen und Stand der Technik

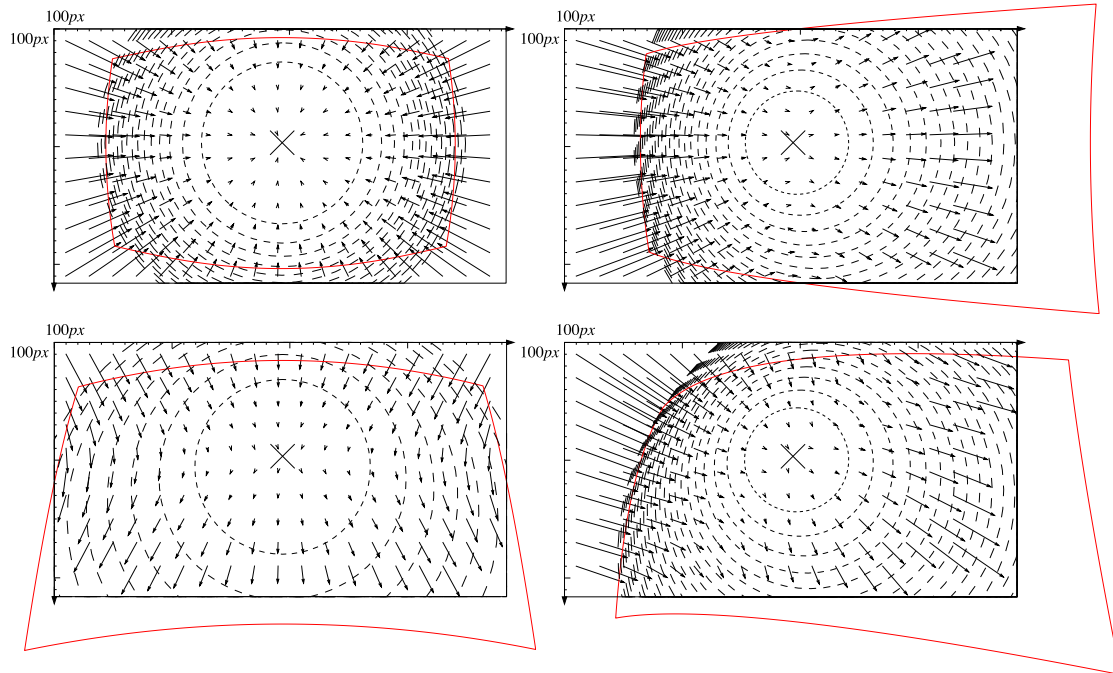


Abbildung 2.4: Verschiedene von Verzeichnungen erzeugte Vektorfelder. Der rote Rahmen zeigt den Bildbereich der Normalperspektive, die gestrichelten Linien Äquipotentiallinien im Vektorfeld der Verzeichnung. In Lesereihenfolge: Verzeichnung einer Gopro Hero 3 Kamera, Y-Tangentialverzeichnung, X-Tangentialverzeichnung, XY-Tangentialverzeichnung

bedeutet, dass Punkte auf Geraden in der Szene auch nach der Projektion wieder auf einer Geraden liegen. Die von einem Objekt ausgesandten und/oder reflektierten Lichtstrahlen passieren die Optik im punktförmigen Kamerazentrum und treffen danach auf die Sensorebene. An diesem Punkt wird die Irradianz gemessen. Die Strahlen werden im Lochkameramodell weder gebrochen noch gebeugt. Die Berechnungen, um eine Szene in ein Bild zu projizieren, lassen sich elegant mittels homogener Koordinaten durchführen [2, Seite 50]. Es ist wichtig dabei zu beachten, dass jede Kamera ein eigenes Koordinatensystem besitzt und die Punkte in ein einheitliches Weltkoordinatensystem transformiert werden, wenn sie zueinander in Beziehung gesetzt werden sollen (siehe Abbildung (2.5)).

2.3.2 Homogene Koordinaten

Im Lochkameramodell schneiden sich alle Sichtstrahlen im Kamerazentrum. Jeder Punkt auf einem dieser Strahlen, wird auf den gleichen Punkt in der projektiven Ebene abgebildet [6, Seite 2]. Wird diese in den \mathbb{R}^3 eingebettet, liegt diese konventionsgemäß parallel zur XY-Ebene und

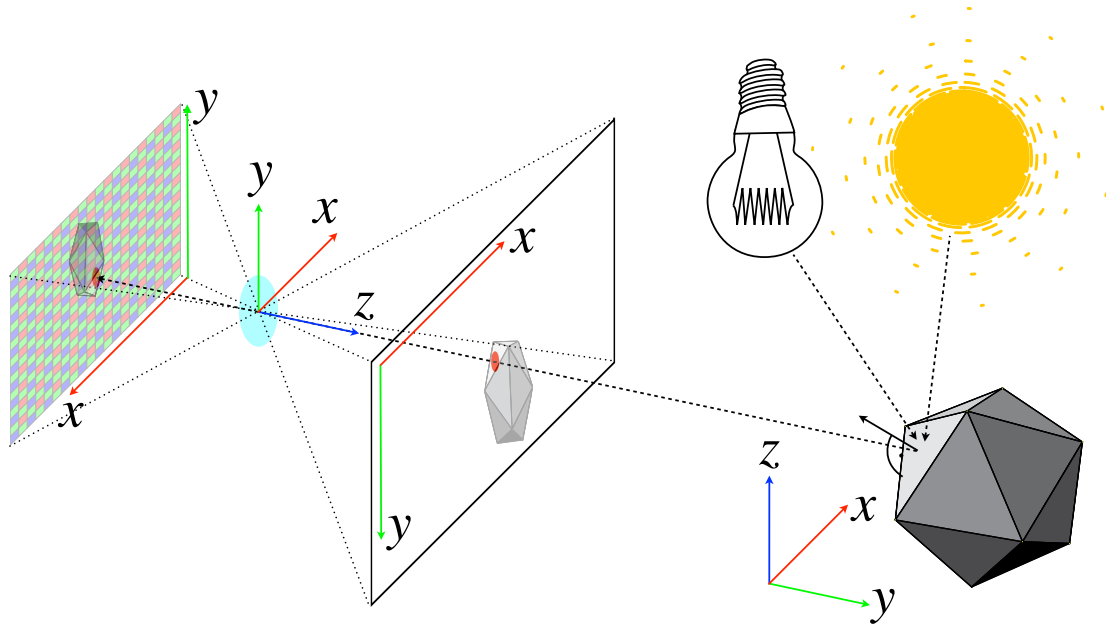


Abbildung 2.5: Schematische Darstellung der Koordinatensysteme, von links nach rechts: Sensor-, Kamera-, Bild- und Weltkoordinaten

enthält den Punkt $(0,0,1)$. Dieser Fakt lässt sich mathematisch einfach herstellen. Eine Gerade in einem Vektorraum lässt sich als $g(\omega) = \vec{q} + \omega \vec{x}$ schreiben mit dem Stützvektor \vec{q} und dem Richtungsvektor \vec{x} . Legen wir das Kamerazentrum in den Nullpunkt, kann die Menge aller Geraden durch das Weglassen des Stützvektors \vec{q} reduziert werden. Bei näherer Betrachtung der Abbildungen von Punkten im \mathbb{R}^3 auf die Projektive Ebene \mathbb{P}^2 fällt auf, dass ω von z abhängt. Eine Gerade durch den Nullpunkt und den Punkt (x, y, z) schneidet die projektive Ebene in $(x/z, y/z, 1)$, da

$$\omega \begin{pmatrix} x \\ y \\ z \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} - 1 = 0$$

$$\omega = \frac{1}{z}$$

Zur Umwandlung von Koordinaten aus dem \mathbb{R}^3 in den \mathbb{P}^2 kann also einfach durch z geteilt und die resultierende 1 ignoriert werden.

Dies ist die Zentralprojektion (siehe Abbildung (2.6a)), welche bei der Fotografie auftritt.

Um eine Karte abzubilden, ist die Orthogonalprojektion (siehe Abbildung (2.6b)) das Mittel der

Wahl. Hierbei ist es so, als besäßen die Objekte orthogonal zur Projektionsebene weder Höhe noch Tiefe. Es wird einfach das Lot zur Projektionsebene aufgetragen.

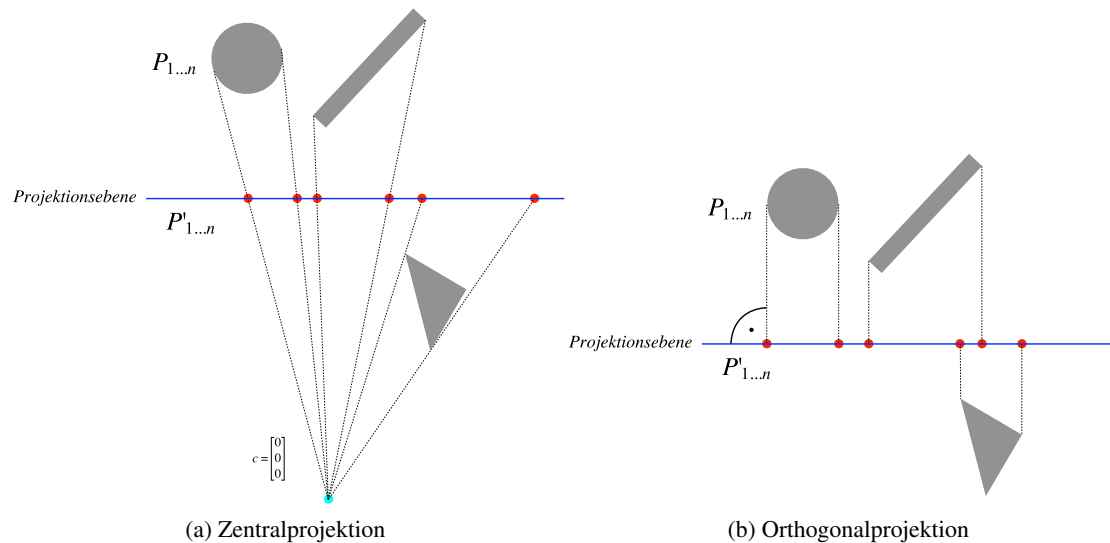


Abbildung 2.6: Ein Schnitt durch eine Szene: Die Sichtstrahlen (gestrichelte Linien) schneiden die Projektionsebene (blau) und bei der Zentralprojektion das Kamerazentrum (cyan). Die Schnittpunkte (rot) markieren die Abbildung nach der Projektion.

Doch auch eine homogene Erweiterung der dreidimensionalen Koordinaten ist lohnenswert. Skalierung, Scherung und Rotation um den Nullpunkt sind im \mathbb{R}^3 per 3x3-Matrix abbildbar. Die Translation geschieht durch Addition eines Positionsvektors. Eine Rotation um einen Ankerpunkt muss in euklidischen Koordinaten durch drei Operationen abgebildet werden und zwar durch sequentiell Ausführung von Translation in den Ankerpunkt, Rotation und Translation zurück. Im homogen erweiterten \mathbb{R}^3 lassen sich sämtliche Transformationen mit einer 4x4 Matrix beschreiben [2, Seite 40]. Diese Vereinheitlichung besitzt mehrere Vorteile:

- Transformationen können durch Multiplikation der Matrizen zusammengefasst werden
- Es kann ein einheitliches Rechenwerk für sämtliche Transformationen benutzt werden, welches hochoptimiert werden kann

Die durch solche Matrizen beschriebenen Transformationen werden affine Transformationen genannt [2, Seite 36].

2.3.3 Epipolargeometrie und Posenschätzung

Die Epipolargeometrie beschäftigt sich mit den projektiven Abbildungen von korrespondierenden Bildpunkten in mehreren Kamerabildern. In zwei Bildern von unterschiedlichen optischen Zentren aus aufgenommen, werden alle Punkte eines Sichtstrahls in einem Bild auf eine Gerade im anderen projiziert. Dieses wird die Epipolarbedingung genannt und mathematisch $p_1^T F p_2 = 0$

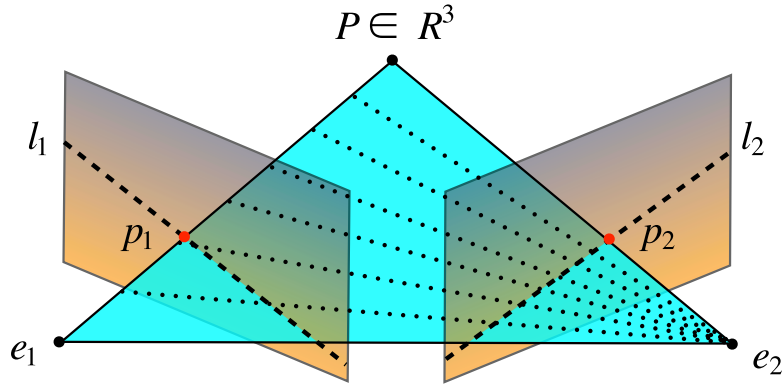


Abbildung 2.7: Bildliche Darstellung wie die Epipolarlinien entstehen

ausgedrückt [7]. F wird die Fundamentalmatrix genannt und beschreibt die Transformation von Punkten in einem Bild zu Linien im anderen. Abbildung 2.7 verbildlicht ein Beispiel. Der 3D-Punkt P wird auf die 2D-Punkte p_1 und p_2 projiziert. Durch den Punkt-Gerade-Dualismus des projektiven Raumes, ist das Ergebnis von $F p_2 = l_1$ (in Hessescher Normalform) und damit $p_1^T l_1 = d$ der Abstand vom Punkt p_1 zur Gerade l_1 . Deshalb kann aus der Epipolarbedingung Gleichung (2.3) hergeleitet werden, um den Fehler einer gefundenen Fundamentalmatrix zu beurteilen:

$$|p_1^T F p_2| = d \quad (2.3)$$

Mit $F^T e' = 0$ können aus der Fundamentalmatrix die Projektionsmatrizen der Kameras extrahiert werden:

$$P_1 = \left(I \mid 0 \right), \quad P_2 = \left([e']_{\times} F \mid e' \right) \quad (2.4)$$

mit $[e']_{\times} F$ als dem Kreuzprodukt von e' mit dem Spaltenraum von F [6, Seite 256]. Aufgrund der Nichtlinearität der involvierten Gleichungen können schon kleine Änderungen der Werte große Auswirkungen auf die rekonstruierten Daten haben (siehe Abbildung 2.8).

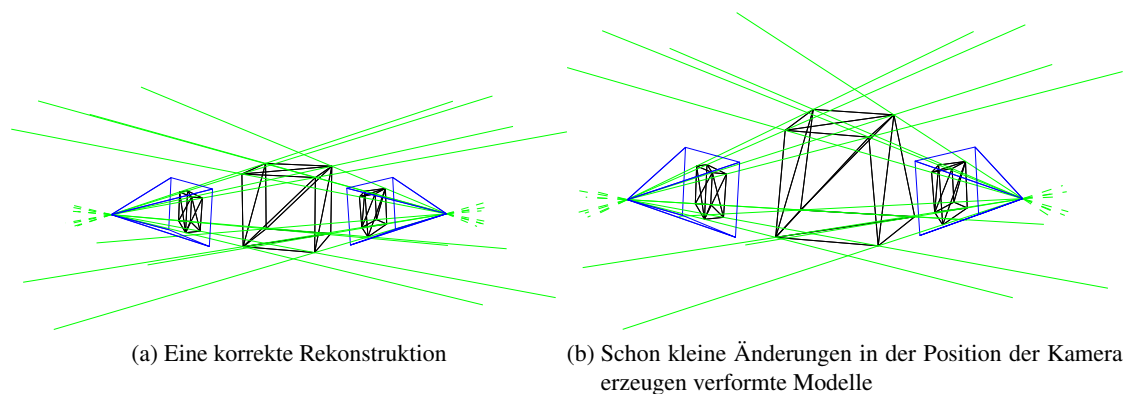


Abbildung 2.8: Ein Zwei-Kamera-Setup. Die Sichtstrahlen sind grün, die Kamerapyramiden in blau eingezeichnet

2.4 Computer-Vision und Algorithmen

Die Computer-Vision beschäftigt sich damit, die Fähigkeiten der visuellen Systeme von Lebewesen auf den Computer zu übertragen und nach Möglichkeit noch zu übertreffen. Trotz der Gemeinsamkeiten in der Funktion von Kameras und biologischen Sehorganen, ist deren Arbeitsweise grundverschieden. Digitale Kameras liefern zum Beispiel eine Menge von Farbwerten in einer festen Anordnung, während bei Katzen nachgewiesen werden konnte, dass ihre Retina Differenzen von Gaußschen Funktionen misst [8]. Ein Verfahren welches in vielen Merkmalstechniken zum Einsatz kommt [9, 10]. Viele der Algorithmen sind nicht trivial und benötigen weitere Erklärung, weshalb in diesem Abschnitt die für die Arbeit wichtigen kurz vorgestellt werden.

2.4.1 Merkmale

Merkmale befassen sich mit dem Problem, korrespondierende Bereiche einer Szene in Bildern derselben zu finden. Die Verfahren bestehen aus drei Phasen, Detektion, Bestimmung der Orientierung und Deskriptorextraktion. In der ersten Phase werden Bildbereiche auf ihre Eignung als Messwert hin untersucht. Sie entscheidet über die Position und Größe eines Merkmals im Bild. Die Festlegung einer Orientierung des Bildbereiches ist optional, da die Rotationsvarianz unerwünschtes Verhalten verursachen kann. Sogenannte *upright* Versionen von Merkmalsmechanismen performen bei der Wiedererkennung in der Regel besser [11]. In der Deskriptorextraktion werden aus den Bildbereichen an den gefundenen Positionen charakteristische Daten berechnet, Deskriptoren genannt, die der Computer miteinander vergleichen kann [2, Seite 207].

Während zu allen drei Schritten verschiedene Verfahren existieren [12, 13, 14, 15, 16, 17, 10, 18], hat sich besonders das *Scale Invariant Feature Transform*-Verfahren (SIFT) durchsetzen können [9]. Hierbei baut der Detektor eine Pyramide aus den Ableitungen der Intensitäten des Bildes in absteigender Auflösung auf. In dieser sogenannten DoG (Difference of Gaussian) Pyramide werden die lokalen Maxima der Intensitäten mit einem Ellipsoid gefittet, um die optimale Größe und Position des Features zu bestimmen (siehe Abbildung 2.9). Der Orientierungsschritt summiert die Gradienten in einem 60° Winkelstück nach Abstand vom Mittelpunkt gewichtet auf und wählt das Stück aus, in welchem der Betrag des Summenvektors am größten ist. In die Berechnung fließen auch alle anderen Winkelstücke ein, deren Beträge sich vom maximalen um weniger als 20% unterscheiden. Der Deskriptor von SIFT ist eine Sammlung von Histogrammen der gewichteten Gradienten aus den um den Mittelpunkt des Features orientierten Bildteilen. Die Histogramme werden um die Invarianz gegenüber Beleuchtungsänderungen zu steigern normalisiert und als Vektor gespeichert. Die Deskriptoren können nun miteinander verglichen werden um ähnliche Strukturen zu finden (siehe Abbildung 2.10).

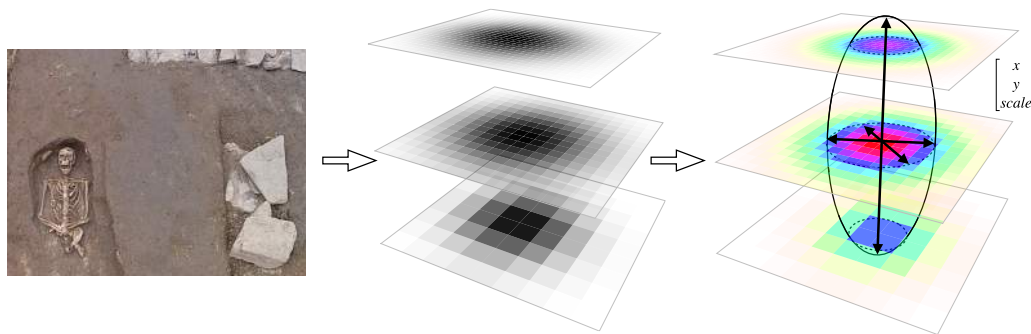


Abbildung 2.9: Die Detektorpipeline von SIFT. Zur Bestimmung der Merkmalspositionen werden aus einem Bild verschiedene Auflösungen produziert und voneinander abgezogen. In diesem sogenannten *scale space* werden die Maxima mit einem Ellipsoid gefittet um die Optimale Position zu finden (Foto © Archaeocopter).

2.4.2 Random Sample Consensus

Merkmalsdeskriptoren arbeiten leider nicht perfekt. Da der SIFT-Detektor (siehe Abschnitt 2.4.1) nur simple Strukturen wie Ecken und Blobs erfasst und der Deskriptor ähnliche Bereiche auch stabil gegenüber Beleuchtungsänderungen und Transformationen zuordnen können soll, enthält das Ergebnis einer Zuordnung etliche falsche Zuordnungen. Diese treten sogar bei perfekten synthetischen Daten auf und lassen sich nicht durch Schwellenwertbildung eliminieren. Fließen diese Ausreißer in die späteren Berechnungen mit ein, beeinflussen sie das Ergebnis negativ.

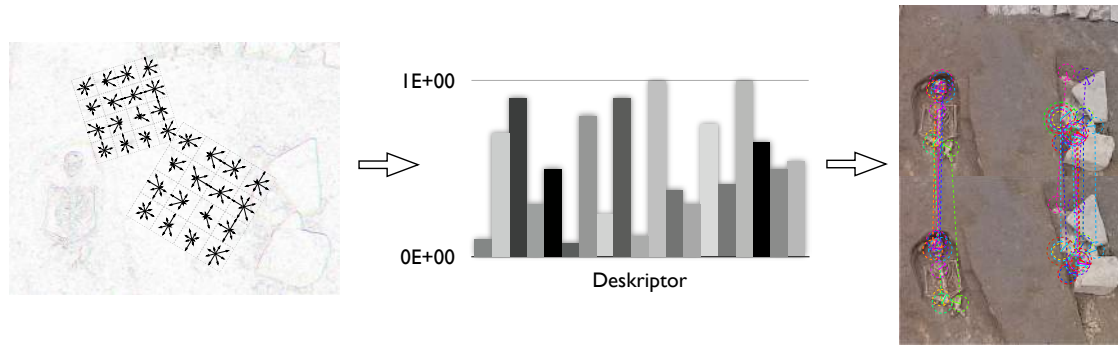


Abbildung 2.10: Der SIFT Deskriptor: An den gefundenen Merkmalspositionen werden Histogramme der Gradienten aufsummiert und die Werte in einen Vektor geschrieben. Dieser ermöglicht Bildausschnitte schnell und transformationsinvariant zu vergleichen (Foto © Archaeocopter).

Random Sample Consensus Set [19] schafft hier Abhilfe, indem es die maximale Konsensmenge findet. Der Algorithmus 2.1 beschreibt das Vorgehen von RANSAC für ein Modell, welches sich aus k Messwerten einer Menge P berechnen lässt.

Algorithm 2.1 Der RANSAC Algorithmus (Pseudocode)

```

N = Anzahl der Messwerte zur Berechnung der freien
    Modellparameter
Wähle zufällig N Messwerte aus P
Berechne die freien Modellparameter anhand dieser Messwerte
M = Messwerte, welche mit dem Modell innerhalb der
    Fehlertoleranz übereinstimmen
Wenn |M| > S
    Verbessere die Modellparameter durch eine Methode der
        Summe der kleinsten Quadrate über M und beende
Wenn nach T Versuchen immer noch keine Konsensmenge größer S
    gefunden wurde
    liefere eine Lösung von der größten gefundenen
        Konsensmenge
ansonsten
    beginne von vorn
    
```

Die Konsensmenge *brute force* zu finden ist bei den komplexen Modellen und der Anzahl der Messwerte in der Computer Vision ein teures Unterfangen: Ein Modell mit k freien Parametern müsste alle k -Teilmengen eines n -elementigen Matchings ausprobieren, um die maximale Konsensmenge zu finden, gegeben durch den Binomialkoeffizienten $\binom{n}{k}$. Die freien Modellparameter einer Fundamentalmatrix betragen $k=7$ und schon in VGA Auflösung aufgenommene Bilder

erzeugen $n=1000$ Keypoints, wofür der Binomialkoeffizient astronomische Summen produziert.

Aufgrund der zufälligen Natur von RANSAC gibt es keine Garantie dafür, dass die gefundene Menge auch wirklich die größte Konsensmenge ist. Jedoch kann durch Anpassung der Anzahl der Versuche T und der Anzahl S für einen beliebigen Anteil an Ausreißern eine beliebig dicht gegen 1 gehende Wahrscheinlichkeit der Korrektheit der Lösung erreicht werden. Mit steigender Wahrscheinlichkeit, steigt auch der Rechenaufwand [6, Seite 119]. Eine zu hohe Wahrscheinlichkeit kann dazu führen, dass RANSAC sogar mehr Zeit in Anspruch nimmt, als das *brute force*-Durchprobieren der Menge.

Angewendet auf die Posenschätzung wird eine Fundamentalmatrix aus zufälligen Korrespondenzen berechnet, alle Punkte aus dem Quellbild in Linien des Zielbildes transformiert und der Abstand der zugehörigen Punkte zu den Linien im Zielbild berechnet.

2.4.3 Bündelblockausgleich (Bundle Adjustment)



Abbildung 2.11: Beispiel eines sich aufschaukelnden Fehlers der Deltatransformation. Die Rekonstruktion startete in der linken unteren Ecke. Schon nach 24 Bildern ist die Größe des Bildes auf weniger als ein Viertel der ursprünglichen Größe geschrumpft, obwohl die Flughöhe nicht entsprechend stieg (Quelle: Institut Gewässerökologie und Binnenfischerei, Deutsches Zentrum für Luft- und Raumfahrt e.V./Optische Informationssysteme, HTW Dresden/Geoinformation, FU Berlin/ Institut für Weltraumwissenschaften).

Durch die Projektion auf eine Bildfläche gehen die Tiefeninformationen verloren. Wurden mithilfe von Merkmalen korrespondierende Punkte auf Bildern gefunden, lassen sich die Fundamen-

talmatrizen schätzen und 3D-Punkte triangulieren. Die Fundamentalmatrix gibt jedoch immer nur das Positions- und Orientierungsdelta von einem zum nächsten Bild an. Der Fehler einer Transformation wird auf diesem Weg an jede neue Kameraposition propagiert und erzeugt unerwünschte Artefakte bei der 3D-Rekonstruktion. Das Modell verformt sich mit jeder weiteren hinzugefügten Kamera immer heftiger (siehe Abbildung 2.11). Um dieses Problem zu lösen, muss die Szene global optimiert werden. Diese Verfahren werden Bündelblockausgleiche genannt und kommen ursprünglich aus der Photogrammetrie [20]. Obwohl die benötigten mathematischen Werkzeuge vom Bündelblockausgleich schon seit längerem bekannt sind, erreichte das Verfahren erst mit dem Aufkommen der Computer breitere Anwendung, da die anfallenden, riesigen Datenmengen nur schwierig ohne Computer zu lösen sind.

Es wird eine Fehlerfunktion definiert, welche den quadratischen Fehler aller Daten aufsummiert. Auf Basis dieser Fehlerfunktion wird ein Gleichungssystem aufgestellt, dass per nichtlinearer Optimierung numerisch minimiert wird [20]. Die Verfahren unterscheiden sich darin, wie die Fehlerfunktion über die Daten aufgestellt wird und dem genutzten Algorithmus zur Lösung des Gleichungssystems. In den meisten Fällen wird der Reprojektionsfehler genommen. Es existieren aber auch Bündelblockausgleich genannte Verfahren, die andere Effekte ausnutzen wie z.B. Zhang et Al., welche den Fehler über projektionsinvariante Verhältnisse aufsummieren [21]. Alle in einer Kamera sichtbaren 3D-Punkte P_j werden durch die Projektionsfunktion F_i in ein virtuelles Foto projiziert und der Abstand zur korrespondierenden Feature Position p_{ij} quadratisch aufsummiert (siehe Formel (2.5)).

$$\sum_{i=0}^n \sum_{j=0}^m (F_i(P_j) - p_{ij})^2 \quad (2.5)$$

So können Kameraorientierungen sowie Kamera- und 3D-Positionen gleichzeitig optimiert werden. Die Projektionsfunktion darf beliebig komplex werden. Es gibt zum Beispiel Versionen, welche auch die Radialverzerrung optimieren [20]. Generell verhält sich das Gleichungssystem jedoch instabiler, wenn mehr Parameter optimiert werden. Die numerischen Verfahren um das Gleichungssystem zu minimieren arbeiten mit linearer Näherung und Gradientenabstieg. Dadurch kann es passieren, dass der Lösungsalgorithmus in ein lokales Minimum konvergiert. Dies ist abhängig von der gefundenen Startlösung und wie glatt die Fehlerfunktion sich verhält [6, Seite 14]. Für einen einfachen Bündelblockausgleich, welcher nur Kamerapositionen und -orientierungen optimiert, wächst der Parametervektor um 6 Freiheitsgrade für jede neue Kamera (3 für Position, 3 für Rotation) und um 3 Freiheitsgrade für jeden 3D-Punkt. Da in 3D-Szenen nicht jeder Punkt auch in jeder Kamera abgebildet wird (z.B. durch Verdeckung oder

Bewegung der Kamera) ist das Gleichungssystem jedoch nur dünnbesetzt. Solche dünnbesetzten Gleichungssysteme lassen sich durch geschicktes Umordnen in weniger komplexe, voneinander unabhängig lösbare Gleichungssysteme zerlegen. So erreichen einige Systeme für beschränkte Parametergrößen sogar Echtzeitperformance [22, 23].

2.4.4 Multiview Stereo

Multiview Stereo fasst Techniken zusammen, welche aus mehr als zwei Bildern 3D-Informationen extrahieren. Khan et al. [24] legen Ebenen in den Raum, auf welche die Bilder projiziert werden. Diese werden π -Ebenen genannt. Ist die Höhe der π -Ebene gleich der Höhe des Objektes überlagern sich die projizierten Bildbereiche an der Stelle des Objektes. Die Varianz der Farbwerte wird als Maß für die Übereinstimmung herangezogen. In Abbildung (2.12) ist das Verfahren bildlich verdeutlicht.

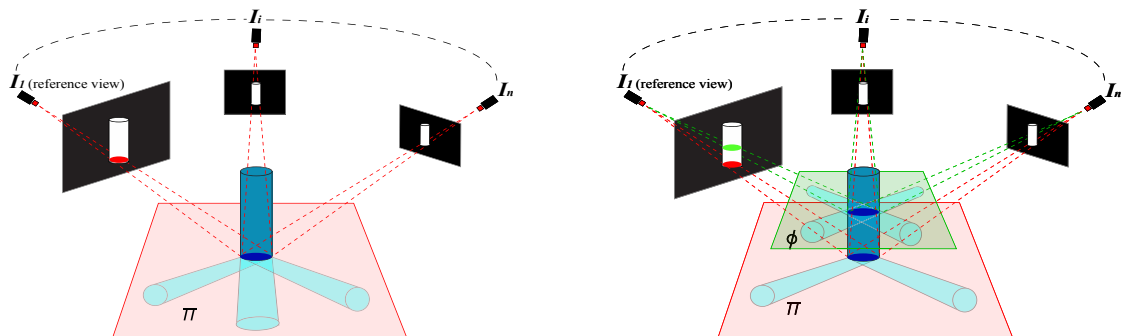


Abbildung 2.12: Multiview Stereo über Korrespondenzanalyse: Auf der π -Ebene der Höhe des Objektes überlagern sich auch die Projektionen der Kameraansichten (Quelle: [24]).

2.4.5 Stitching

Beim Stitching wird ein großes Bild aus vielen kleinen zusammengesetzt. In der Theorie ist es einfach: Liegen die optischen Zentren der aufgenommenen Bilder in einem Punkt, verhalten sich die Sichtstrahlen in beiden Bildern gleich. Die überlappenden Bildbereiche können dazu genutzt werden, um eine Transformation von einem Bild zum nächsten zu berechnen. Diese kann auf die neuen Bildbereiche angewendet werden. Es können keine Artefakte durch Verdeckungen oder Änderungen des Kamerazentrums entstehen (siehe Abbildung (2.13)).

In der Praxis sieht die Sache etwas anders aus. Optiken entsprechen den mathematischen Modellen nur näherungsweise und es ist, selbst mit professioneller Ausrüstung, nahezu unmöglich das

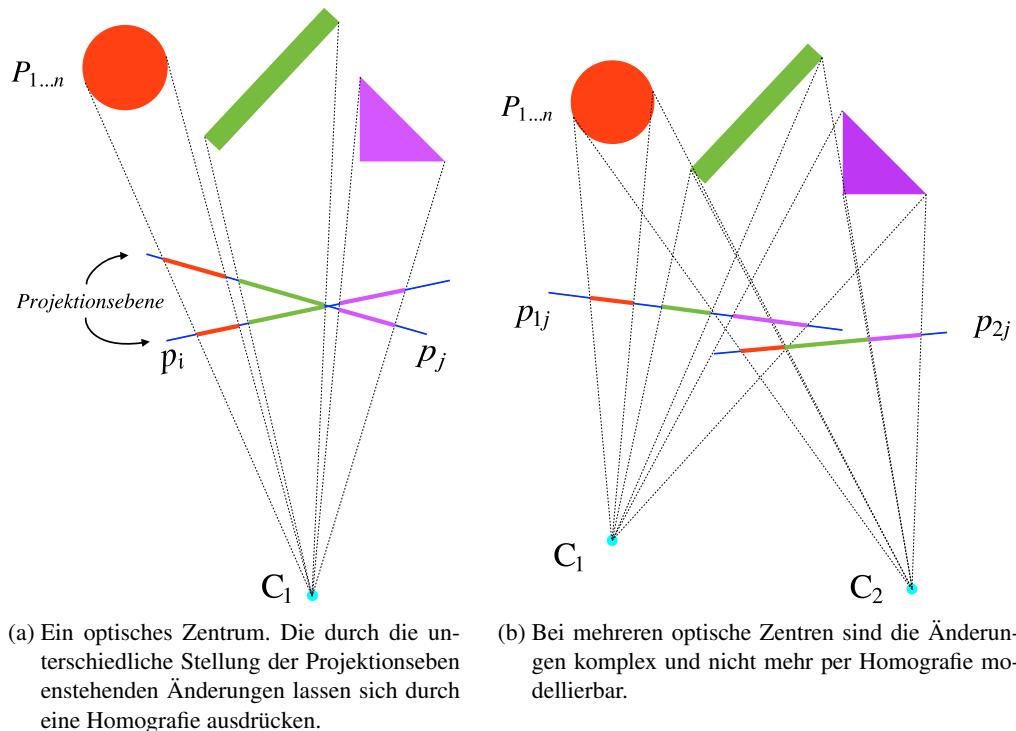


Abbildung 2.13: Durch unterschiedliche Kamerazentren ändern sich die Bildinhalte in komplexer Weise und verursachen Artefakte beim Stitching von Panoramas.

Kamerazentrum perfekt in einem Punkt zu halten. Deshalb bieten verschiedene Firmen sogar motorisierte Roboteraufsätze für Stative an⁴.

2.4.6 Farbtransfer

Auf verschiedenen Aufnahmen einer Szene kann dasselbe Objekt unterschiedliche Farbwerte erzeugen. Diese werden durch Änderung der Beleuchtungssituation, Blende, Empfindlichkeit, Belichtungszeit, Weißabgleich, Sensor oder Position verursacht. Merkmalstechniken verwenden zahlreiche Mittel, um möglichst beleuchtungsunabhängig zu sein. So können diese Punkte zwar zugeordnet werden, nach dem Zusammensetzen zu einem Panorama sind diese Farbunterschiede jedoch störend. Tian et al. [25] verwenden Matrixtransformationen um korrespondierende Farbwerte einander anzupassen. In der Veröffentlichung wird eine affin erweiterte, lineare Matrix empfohlen. Die Matrixtransformation eines Farbwerts p zu einem anderen Farbwert q durch die Matrix M sieht wie folgt aus:

⁴<http://gigapan.com/cms/shop/epic-pro>

$$pM = q$$

$$\text{mit } p = \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix}, q = \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix} \text{ und } M = \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix}$$

Damit wird eine quadratische Fehlerfunktion definiert:

$$e = \sum ((pM - q) \cdot (pM - q)) \quad (2.6)$$

Die Formel aus (2.6) kann durch partielle Ableitungen aller Komponenten und Ausfaktorisierung der Unbekannten in Gleichungssystem (2.7) überführt werden.

$$\begin{pmatrix} & & & 0 & 0 & 0 & 0 & 0 & 0 \\ & S & & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & & & & 0 & 0 & 0 \\ 0 & 0 & 0 & & S & & 0 & 0 & 0 \\ 0 & 0 & 0 & & & & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & & S & \\ 0 & 0 & 0 & 0 & 0 & 0 & & & \end{pmatrix} \times \begin{pmatrix} a_{00} \\ a_{01} \\ a_{02} \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{20} \\ a_{21} \\ a_{22} \end{pmatrix} = \sum \begin{pmatrix} r_2 r_1 \\ r_2 g_1 \\ r_2 b_1 \\ g_2 r_1 \\ g_2 g_1 \\ g_2 b_1 \\ b_2 r_1 \\ b_2 g_1 \\ b_2 b_1 \end{pmatrix} \quad (2.7)$$

$$\text{mit } S = \sum \begin{pmatrix} r_1^2 & r_1 g_1 & r_1 b_1 \\ r_1 g_1 & g_1^2 & g_1 b_1 \\ r_1 b_1 & g_1 b_1 & b_1^2 \end{pmatrix}$$

Das Gleichungssystem kann in drei linear voneinander unabhängige Gleichungssysteme geringerer Komplexität zerlegt werden.

2.5 Verwendete APIs und Software

Die im Zuge der Arbeit erstellten Programme wurden in Java und C geschrieben. Java bietet mit Eclipse eine auf den meisten Plattformen verfügbare kostenlose Entwicklungsumgebung und die Möglichkeit native Bibliotheken aufzurufen. Um plattformunabhängig zu bleiben, fiel die Entscheidung bei der Wahl der verwendeten Grafikbibliothek auf OpenGL. OpenGL (Open Graphics Library)⁵ abstrahiert die Grafikkarte und bietet so eine für alle Grafikkarten einheitliche Schnittstelle an. Grafikkarten besitzen für die Verarbeitung von 3D-Geometrie und Pixeldaten spezialisierte Rechenwerke, weshalb sie Aufgaben in diesem Bereich performanter erledigen können, als normale Prozessoren. OpenGL lässt sich über LWJGL (Lightweight Java Game Library)⁶ aus Java heraus ansprechen. LWJGL ist ein automatisch erzeugtes Binding an OpenGL, weshalb der Java-Code sehr dem C-Code ähnelt und sich auch ähnlich verhält.

Um die EXIF-Daten von Bildern zu lesen und zu schreiben wurde die Apache Commons Imaging⁷ Bibliothek verwendet.

2.5.1 VisualSFM

VisualSFM⁸ ist eine Software zur 3D-Rekonstruktion aus Bildern. Sie bietet einen GUI- als auch einen Kommandozeilenmodus. VisualSFM ist als *closed source binary* für Linux, Mac OS und Windows erhältlich. VisualSFM besitzt ein eigenes auf SIFT (siehe Abschnitt 2.4.1) basierendes Merkmal- und Zuordnungs-Modul, bietet jedoch auch an, dass eigene Merkmalpositionen und/oder -paarungen an das Programm übergeben werden können. Dies ist per API oder per Datei möglich. Die Software ist vielfältig über eine Textdatei konfigurierbar.

Aus den hier vorher genannten Schritten ergibt sich eine mögliche 3D-Rekonstruktionspipeline. In Abbildung 2.14 ist der von VisualSFM verwendete Arbeitsfluss dargestellt. Nach jeder neu hinzugefügten Kamera wird ein Bündelblockausgleich (siehe 2.4.3) durchlaufen, um den aufakkumulierten Fehler so gering wie möglich zu halten, damit die Lösung häufiger in das globale Optimum konvergiert [26].

⁵<http://www.opengl.org/>

⁶<http://lwjgl.org>

⁷<http://commons.apache.org/proper/commons-imaging/>

⁸<http://ccwu.me/vsfm/>

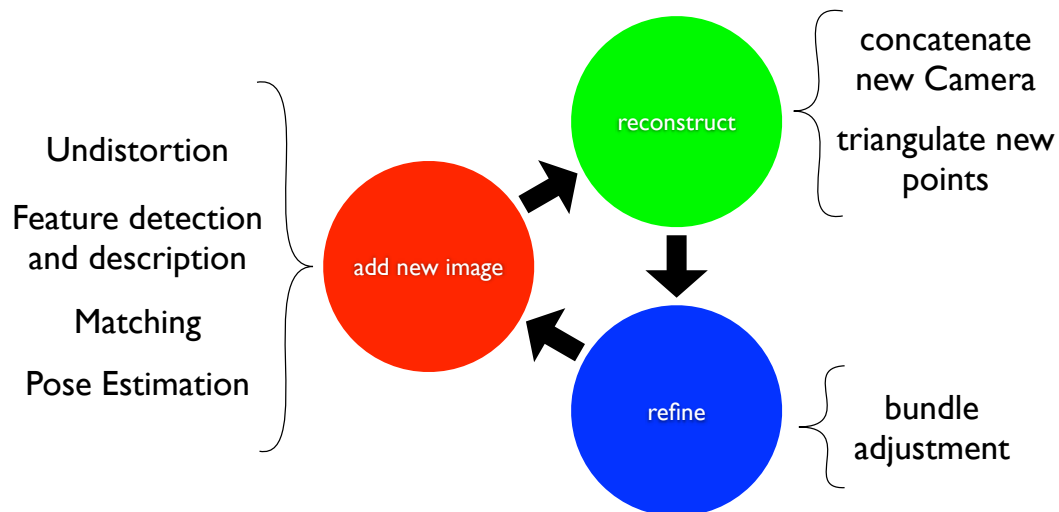


Abbildung 2.14: Zustandsdiagramm der 3D-Rekonstruktion von VisualSFM. Um die Genauigkeit der Rekonstruktion zu steigern, wird nach jedem *reconstruct* Schritt ein *refine* durchlaufen. Einstiegspunkt ist *add new image*. Hinter den geschweiften Klammern stehen die dem Zustand zugrunde liegenden Einzelschritte.

2.5.2 Meshlab

Meshlab ist eine Open Source Software zur Geometrieverarbeitung. Wie VisualSFM bietet auch Meshlab einen GUI und einen Kommandozeilenmodus. In dieser Arbeit wurde Meshlab benutzt, um Punktwolken zu bereinigen, die Poissonrekonstruktion durchzuführen und Texturen auf Oberflächen zu projizieren.

2.6 Das Problem aus Sicht der Kartographen

Aufsichten, Schnitte und Karten haben gemein, dass sie alle orthogonal projiziert werden. Deshalb können sie nicht durch herkömmliche, zentralprojizierende Kameras erstellt werden. Bisherige Verfahren verwenden ein DSM (Digital Surface Model) oder DTM (Digital Terrain Model), um die Fotos orthogonal zu rektifizieren [27, 28, 29, 30, 31, 32]. Ein DSM enthält Höhendaten vom Ist-Zustand der Umgebung und kann durch Scans erstellt werden. Das DTM enthält nur die Terraindaten ohne Bebauung und optional ohne Vegetation. Die nicht erwünschten Objekte müssen durch geeignete Messverfahren oder Nachbearbeitung entfernt werden. Die hier genannten Veröffentlichungen gewinnen ihre 3D-Daten aus entweder per digitalem Stereoauswertegerät oder per LiDAR Scanner. Ein Stereoauswertegerät kann Höhenlinien aus Stereobildern mittels

halbautomatischem Prozess erstellen [33, Kapitel 7]. LiDAR Scanner bestrahlen ihre Umgebung mit Laserlicht und können aus dem reflektierten Licht Rückschlüsse über die 3D-Struktur der Szene ziehen.

3D-Flächenmodelle vereinfachen das Problem darauf, die Kamerabilder korrekt auf das Modell zu projizieren. Bei der Implementation von Skarlatos et al. [34] werden die Fotos auf ein 3D-Flächenmodell der Umgebung projiziert und die Orthogonalprojektion pixelweise so ausmaskiert, dass nur noch die Dächer übrig bleiben. Diese Teile werden ins Originalbild eingezeichnet. Das heißt, dass Gebäude in ihrer korrekten Größe und Position erkennbar sind, aber von Gebäuden verdeckte Bereiche verdeckt bleiben. Dieser Effekt wird *double mapping* genannt (siehe Abbildung 2.15).



Abbildung 2.15: Der Double Mapping Effekt auf einem rektifizierten Foto. Die korrekte Größe des Hauses kann abgelesen werden, Teile der Umgebung werden jedoch verdeckt (Quelle [27]).

Deshalb schlagen die Autoren vor, Löcher in der per LiDAR gewonnenen Punktwolke zu schließen und dann die Fotos auf das Modell zu projizieren, um anschließend orthogonal zu rektifizieren. So bleiben verdeckte Bereiche unbemalt. Die Details des Algorithmus werden nicht genannt. Kuzmin et al. [27] schlagen vor, ein DBM (Digital Building Modell), welches keine Terraindaten enthält, einmal orthogonal zu projizieren und die Differenzmenge mit der Projektion des DBM aus Sicht der Kamera zu bilden. Diese Projektion wird ohne Rasterisierung ausgeführt, um die Polygondaten zu erhalten. So können die verdeckten Bereiche berechnet und dann später aus anderen Ansichten hergestellt werden. Als Vorteil wird angegeben, dass der Speicheraufwand im Vergleich zu Z-Buffer basierten Verfahren geringer ist. Diese Verfahren rendern das DSM aus Sicht der Kamera mit derselben Auflösung wie den fotografierten Kartenausschnitt und markieren einen Pixel beim Überschreiben mit einem Farbwert mit geringerer Tiefe als verdeckt. So werden alle verdeckten Pixel berechnet [28]. Die verdeckten Bereiche können später wie bei Skarlatos et al. aus anderen Ansichten aufgefüllt werden. Alle Verfahren gehen davon

2 Grundlagen und Stand der Technik

aus, dass kalibrierte Stereokameras verwendet werden, sowie exakte Positionen der Kameras und dichte 3D-Flächenmodelle der Umgebung verfügbar sind.

Schickler und Thorpe [35] schreiben, dass das 3D-Flächenmodell der Umgebung zu erstellen der teuerste Teil des Prozesses sei. Nach der Orthogonalprojektion, werden die Einzelbilder zu einem Panorama zusammengefügt. In diesem Schritt werden die vorher noch nicht gefüllten Bereiche aufgefüllt. Schickler und Thorpe generieren aus den Verdeckungs- und Überlappungsbereichen initiale Mosaikgrenzen, die von folgenden Faktoren abhängen:

1. wie dicht der Pixel im Einzelbild an einem verdeckten Bereich liegt
2. wie hoch der Abstand vom Nadir ist
3. wie steil der Sichtstrahl auf die gesehene Fläche trifft

Diese Grenzen werden nachträglich verfeinert, damit sie hohe Objekte meiden, in denen der Wechsel der Perspektive auffallen könnte. Dieser Schritt muss außerdem verschiedene Lichtsituation bei der Aufnahme kompensieren, denn zwischen zwei Bildern können sehr unterschiedliche Beleuchtungssituationen herrschen. Für diesen Schritt werden in den genannten Veröffentlichungen keine Verfahren genannt [35].

Neitzel et al. [36] gehen einen ähnlichen Weg wie diese Arbeit. Ihr Fokus liegt jedoch darauf dichte 3D-Punktwolken der überflogenen Areale zu erstellen und diese dann orthogonal zu projizieren.

3 Entwurf

In diesem Kapitel werden die eigenen Ideen und Entwurfsentscheidungen diskutiert. Es findet eine Betrachtung der Hard- und Softwareentscheidungen, welche zum endgültigen Lösungsweg geführt haben statt und zeigt diejenigen Probleme auf, welche in der Entwurfsphase erkannt wurden.

3.1 Die Ausgangssituation

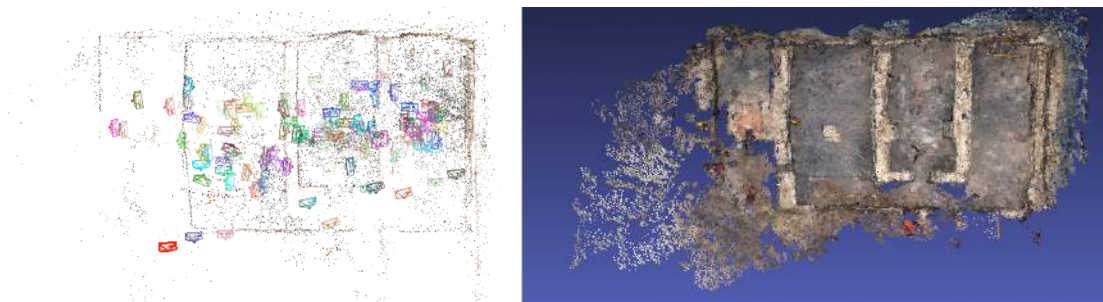


Abbildung 3.1: Eine Rekonstruktion der Ausgrabungsstätte Galeriestraße in Dresden. Links die dünnbesetzte Punktwolke mit eingezeichneten Kamerapyramiden (49669 3D-Punkte), rechts eine mit CMVS/PMVS erstellte dichtbesetzte Punktwolke (378496 3D-Punkte)

Viele Verfahren werden unter dem Begriff 3D-Rekonstruktion geführt, liefern aber grundlegend verschiedene Ergebnisse. Sind korrespondierende Punkte in einer Bildmenge per Merkmalstechnik (siehe Abschnitt 2.4.1) gefunden worden, werden in einem ersten Rekonstruktionsschritt per Posenschätzung (siehe Abschnitt 2.3.3) die Kamerapositionen und ausgewählte Punkte rekonstruiert. Dieses Modell wird die dünnbesetzte Punktwolke genannt. Wurden die Kamerapositionen in einer 3D-Szene registriert, können sie in einem zweiten Rekonstruktionsschritt zur Erstellung einer dichtbesetzten Punktwolke (siehe Abbildung 3.1) herangezogen werden. Der Arbeitsfluß von VisualSFM bindet hierfür das von Furukawa entwickelte CMVS/PMVS¹ an.

¹<http://www.di.ens.fr/cmvs>

Obwohl das Ergebnis von CMVS/PMVS dichtbesetzt genannt wird, ist es, wie die dünnbesetzte Punktwolke, nur eine Punktwolke. Allerdings ist die Punktdichte höher und jedem Punkt ist eine Normale zugeordnet, welche über die sogenannte *photometric discrepancy function* berechnet wird. Hierfür nimmt Furukawa an, dass alle Oberflächen sich nach dem Lambertschen Gesetz verhalten und erwähnt, dass dieses Modell bei Glanzlichtern oder Verdeckungen versagt [37]. Die dichtbesetzten Punktwolken mit Normalen eignen sich für eine Oberflächenrekonstruktion besser, als die dünnbesetzten. Furukawa schlägt die Poissonoberflächenrekonstruktion vor [38]. Danach können die Texturen auf die einzelnen Polygone des Modells projiziert werden. Abbildung 3.2 zeigt ein Diagramm des Informationsflusses der benutzten 3D-Rekonstruktion.

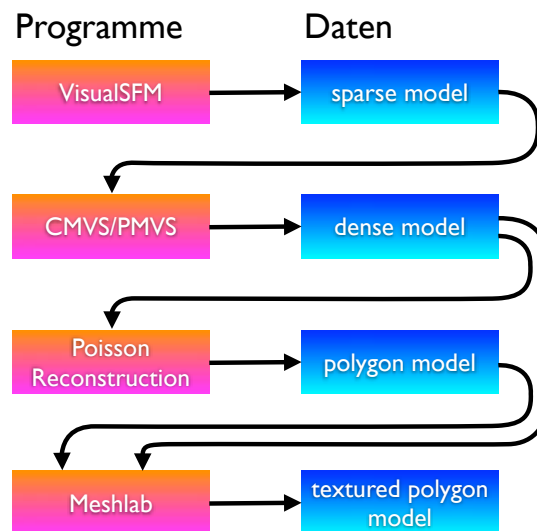


Abbildung 3.2: Der Informationsfluss und die Abhängigkeiten der Programme auf dem Weg von Bild-daten zum fertig texturierten 3D-Flächenmodell.

3.2 Hardwarelimitierungen

Empfindliche und sperrige Hardware wie LiDAR-Scanner oder hochempfindliche GPS-Module, welche in den erwähnten Arbeiten benutzt werden, kommen für das Archaeocopter-Projekt nicht in Frage, da die Copter nur kleine Ausrüstungsteile tragen können und Robustheit eines der Hauptziele des Archaeocopterprojektes ist. Obwohl für manche Multicoptermodelle Nutzlasten von mehreren Kilos angegeben werden, verkürzt jedes zusätzliche Gramm Gewicht die Flugzeit pro Akkuladung.

RGB-D Sensoren wie Microsoft Kinect sind untauglich bei starken Sonnenlichteinfall, was sie für den Außeneinsatz disqualifiziert [39].

3.3 Softwarebeschränkungen

In der Arbeitsgruppe wird die Software VisualSFM benutzt um 3D-Modelle aus Bildern zu erstellen. Zu Beginn der Arbeit war VisualSFM als ausführbare Datei nur für Windows erhältlich. Unter Linux und Mac OS musste die Software aus vorkompilierten Teilen zusammen mit als Open Source erhältlichen Teilen gelinked werden. Die ersten Versuche von 3D-Rekonstruktionen lieferten ernüchternde Resultate. Die Modelle brauchten Stunden bis Tage zur Vollendung, waren deformiert und mit Wolken von fehlerhaften Punkten gefüllt. Es kristallisierte sich heraus, dass es nicht von Vorteil ist, jedes Einzelbild eines Videos in die 3D-Rekonstruktion einfließen zu lassen: Die Rechenzeit steigt stark an mit der Anzahl der Bilder. Doch nach einer vorherigen Verzeichnungskorrektur (siehe Abschnitt 2.2.3), Histogrammausgleich, sowie sorgfältiger Keyframe-Auswahl verbesserten sich die Ergebnisse. An neuen Schwierigkeiten mangelte es trotzdem nicht. Die Datenmenge nimmt allein schon wegen der schiereren Größe der überflogenen Fläche horrende Ausmaße an. So viele Daten zu einem 3D-Modell zu verrechnen dauert trotz sorgfältiger Keyframe-Auswahl seine Zeit. Bei einem Datensatz von ca. 3100 Bildern benötigte ein Rechner mit aktueller Ausstattung 29477 Minuten was ca. 20,5 Tagen entspricht. Um die Rechenzeit bei der Rekonstruktion von großen Landstrichen zu verkürzen, bietet VisualSFM die als experimentell markierte Option an, die Zuordnung abhängig von GPS Positionen einzuschränken. So kann eingestellt werden, dass jedes Bild nur mit Bildern innerhalb eines bestimmten Radius kombiniert wird. Hierfür müssen die Daten in den korrekten EXIF-Feldern der Bilddateien stehen. Die Berechnung beim Helsungendatensatz verkürzte sich durch die Nutzung von GPS-Information bei einem ähnlichen Ergebnis auf 7,5 Tage. Der Zuordnungsradius wurde hierfür auf 100 Meter gesetzt.

3.4 Die Projektion

Nachdem das Problem ein 3D-Modell aus Bildern zu berechnen gelöst war, blieb die Frage der Kartenprojektion. Eine Orthogonalprojektion aus einem 3D-Flächenmodell zu erstellen, ist rechnerisch und technisch trivial. Gängige 3D-Grafikbibliotheken bringen die Orthogonalprojektion von Hause aus mit². Doch die zu dem Zeitpunkt der Arbeit in der Arbeitsgruppe verwendeten

²<https://www.opengl.org/sdk/docs/man2/xhtml/glOrtho.xml>

Werkzeuge (Meshlab) konnten nur in eine Textur projizieren, was die Pixelauflösung für ein Modell automatisch auf die von der Grafikkarte verarbeitbare maximale Texturauflösung reduzierte. Die für das Projekt verfügbare OpenGL Hardware unterstützte eine maximale Texturauflösung von 16384x16384 Pixeln. Die Kartenfläche auf eine passende Größe zu verkleinern ist keine Option. Eine Auflösung von 100 Pixeln pro Meter würde ab einer Kantenlänge 160m unterschritten. Das ist zu wenig für die Anwendung in der Kartographie.

Mit der Kartengenerierung kann an verschiedenen Punkten in der 3D-Rekonstruktionspipeline begonnen werden. Es ist fraglich, ob ein dichtbesetztes Modell besser für eine Kartendarstellung ist. Da viele Flächen im endgültigen Orthophoto eventuell gar nicht zu sehen sind (siehe Abbildung 3.3), werden dabei viele im Orthophoto unsichtbare Flächen berechnet. Desweiteren sind Techniken, die auf der dichten Rekonstruktion aufsetzen, schlecht als online Variante implementierbar. Es gibt bereits Systeme, welche dünnbesetzte 3D-Rekonstruktionen aus Bilddaten in Echtzeit anfertigen können [40], deshalb fiel die Entscheidung an der dünnbesetzten Punktwolke anzusetzen.

3.5 Gigapixelgroße Bilder

Der letzte unbekannte Schritt ist das Rendern riesiger Bilder. Java benutzt intern ein einzelnes Integer Array um die Pixeldaten zu speichern. Da in Java nur 32-Bit Pointer möglich sind, ist die maximale Pixelanzahl der Java eigenen Objekte und Methoden auf Integer.MAX_VALUE (es sind ein paar weniger, da einige Pointer reserviert sind) beschränkt. Das bedeutet, dass keine Bilder größer als 2.15 Gigapixel geschrieben werden können. Es existieren Bibliotheken mit denen größere Bilder geschrieben werden können, indem die Pixeldaten in zweidimensionalen Arrays gespeichert werden³. Hiermit trat ein neues Problem auf: Viele Bildbetrachtungsprogramme können nicht mit so großen Bildern umgehen. So brauchte die Vorschau von Mac OS mehrere Minuten um ein Bild zu laden oder die Zoomstufe zu ändern. Diese Nachteile machten das Arbeiten mit den großen Bildern sehr zeitaufwändig und unpraktikabel, weshalb der Ansatz das Orthophoto in eine Datei zu schreiben verworfen wurde. Stattdessen machte es mehr Sinn das Bild in kleine Kacheln zu unterteilen und ein spezielles Anzeigeprogramm zu schreiben, welches die Kacheln in der korrekten Anordnung darstellt.

³<https://code.google.com/p/pngj/>

3.6 Farben

Die ersten Panoramen litten unter der deutlichen Verschiedenfarbigkeit der Einzelbilder. Das Thema ist Gegenstand aktueller Forschung [41]. Jedes Verfahren besitzt seine Nachteile. Da es noch keine allumfassende Lösung hierfür gibt, liegt mitunter daran, dass die Änderung des Messwertes auf dem Sensor durch etliche Faktoren komplex beeinflusst wird: Wie ist der Winkel zu allen Lichtquellen? Wo werden die Lichtquellen verdeckt? Wie ist die Farbe des zugrunde liegenden Materials und in welcher Farbe strahlt das Licht überhaupt? Menschen verstehen das komplexe Zusammenspiel von Licht und Farben intuitiv. Fährt ein Auto unter einem Baum entlang und wird durch die Blätter beschattet, weiss jedes Kind, dass der Lack an allen Stellen dieselbe Farbe besitzt. Für einen Computer ist die Situation unklarer. Selbst globale Änderungen der Lichtstimmung durch Wolken oder Wechsel des Sonnenstandes sind für einen Computer nicht trivial. Merkmalstechniken nutzen zahlreiche Mittel wie die Normalisierung der Intensitäten des Bildes oder das Verwenden von Verhältnissen um möglichst unabhängig von Beleuchtungsänderungen zu sein [10, 17]. So eine Bearbeitung mindert jedoch den visuellen Eindruck enorm. Deshalb muss eine Technik geschaffen werden, welche die Farben von zugeordneten Merkmalen ineinander überführt. Die ersten Versuche mit im Studium behandelten diskreten Histogrammausgleichen über die kumulative Verteilungsfunktion verliefen ungenügend.

Die kumulative Verteilungsfunktion ist das Integral des Histogramms. Aus ihr kann eine Inversfunktion berechnet werden, mit der ein Bild auf seinen optimalen Kontrast gespreizt werden kann. Dieser Ansatz versagte leider, da die Histogramme der Bilder zu unterschiedlich sind und nur gegen lokale Optima konvergieren (siehe Abbildung 3.4). Der nächste Ansatz benutzte die Helligkeitsübertragungsfunktion über Teilbereiche der Bilder. Dafür wird der Farbbereich des Bildes so gespreizt, dass die kumulative Verteilungsfunktion möglichst ähnlich wird [4]. Dieser Ansatz verminderte die Dynamik der Bilder nach mehrfacher Anwendung so stark, dass die Bilder sehr unansehnlich wurden (siehe Abbildung 3.5). Obwohl nicht-modellgetriebene Ansätze laut Xu et al. bessere Ergebnisse versprechen [42] liefen die Versuche mit dem modellgetriebenen Ansatz von Tian et al. besser. Dieser lieferte stabilere Ergebnisse und auch nach mehrmaliger Anwendung litt die Dynamik nur gering. Xu et al. behandeln in ihrer Übersicht jedoch auch nur den Farbausgleich zwischen Bildpaaren. Der Ausgleich zwischen größeren Mengen wird nicht betrachtet.

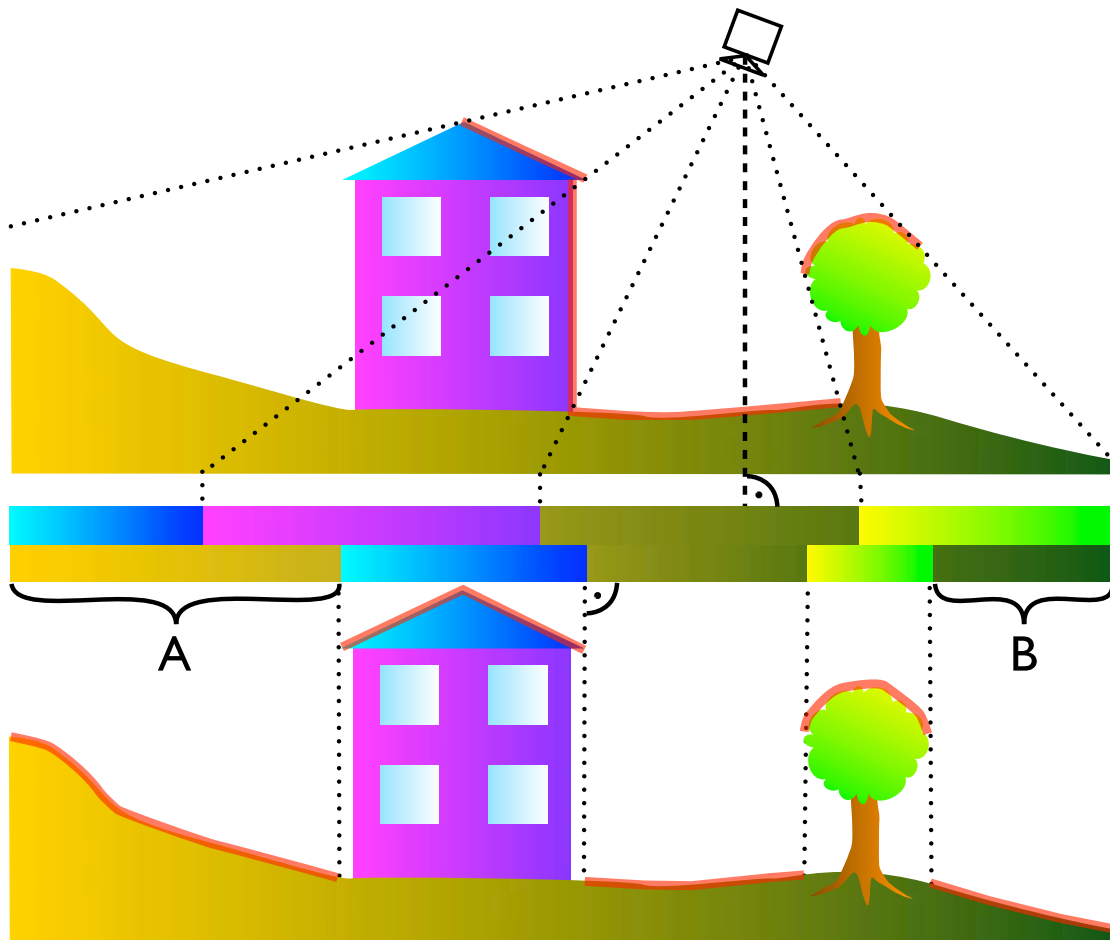


Abbildung 3.3: Verdeckungen in der Zentral- und Orthogonalprojektion im Vergleich. Die roten Linien in der Szene markieren die in der Projektion sichtbaren Oberflächen, die schwarzen punktierten Linien exemplarische Sichtstrahlen. Die Zentralprojektion (oben) erzeugt andere Verdeckungen (Mittelstreifen) als die Orthogonalprojektion (unten). Dadurch treten nach einer Rektifizierung in Bereichen A und B *double mapping*-Effekte auf.



Abbildung 3.4: Drei überlappende Bilder, zeitlich kurz nacheinander aufgenommen und per kumulative Verteilungsfunktion gespreiztem Farbraum. Die lokalen Optima produzieren verschiedene Farben für dieselben Objekte (Quelle: HTW Dresden/Geoinformation, GEO-METRIK-Ingenieurgesellschaft mbH Magdeburg).

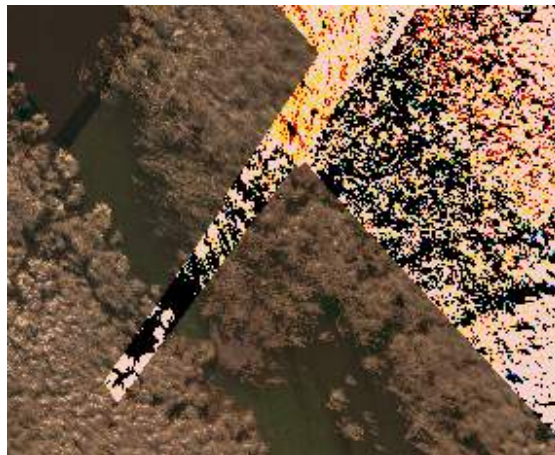


Abbildung 3.5: Ausschnitt eines Orthophotos nach mehrmaligem diskreten Histogrammausgleich. Die Quellbilder oben rechts waren im Vergleich zu den Bildern links unten stark unterbelichtet. Obwohl die Summen der Farben nach dem Histogrammausgleich ähnlich sind, ist der visuelle Eindruck unbefriedigend (Quelle: Institut Gewässerökologie und Binnenfischerei, Deutsches Zentrum für Luft-und Raumfahrt e.V./Optische Informationssysteme, HTW Dresden/Geoinformation, FU Berlin/ Institut für Weltraumwissenschaften).

4 Implementierung

Die aufgabenbezogenen Entscheidungen werden in diesem Kapitel diskutiert. Entscheidungen, welche nicht vom Problem beeinflusst wurden, werden nicht betrachtet.

4.1 Die Kartenebene

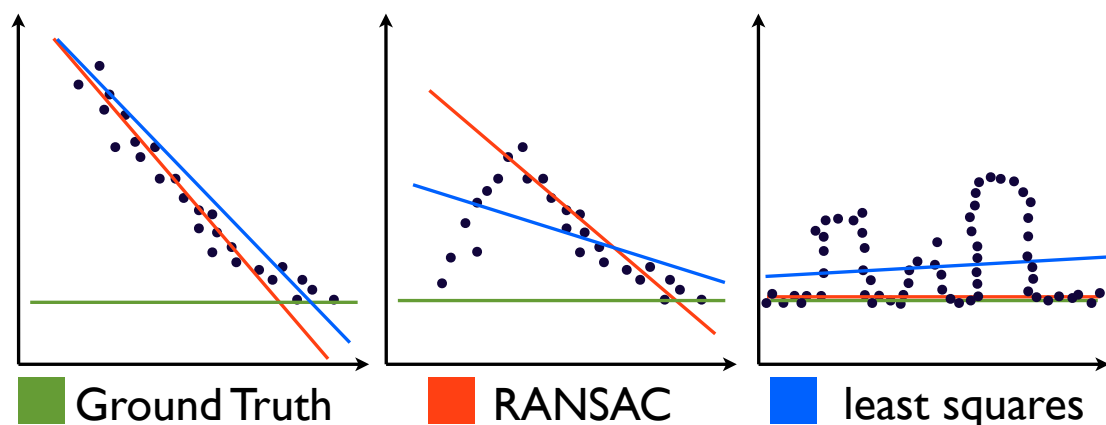


Abbildung 4.1: Schnitt durch eine 3D-Geometrie. Wird nur ein Steilhang (links) oder Berg (mitte) rekonstruiert versagt die algorithmische Bestimmung der Projektionsebene. Einzig bei Gegenden, welche plan mit Aufbauten sind, kann RANSAC eine geeignete Ebene finden.

Wenn Bilder auf eine Ebene projiziert werden sollen, muss erst einmal eine Ebene definiert werden. Ist die Kartenebene unbekannt, wie im Falle der in der Arbeit verwendeten Daten, kann eine Kartenebene aus den von VisualSFM rekonstruierten Punkten geschätzt werden. Dazu existieren verschiedene Möglichkeiten. Je flacher und ausgedehnter das gescannte Areal ist, umso besser eignet sich die Ebene, welche den quadratischen Abstand aller Punkte zur Ebene minimiert. Doch dieser Ansatz versagt bei schrägen Wänden (Berge, Gruben) oder Gebieten mit vielen hohen Objekten (siehe Abbildung 4.1). Bei einer urbanen Umgebung kann ein RANSAC-Ansatz

4 Implementierung

bessere Ergebnisse liefern, da die Gebäude durch ihre unterschiedliche Höhe als Ausreißer erkennbar bleiben. Bei sehr schwierigen Punktwolken bleibt immer noch die Möglichkeit einer manuellen Definition einer Projektionsebene. Der Fehler einer Ausgleichsebene im \mathbb{R}^3 lässt sich in homogenen Koordinaten wie folgt definieren mit \vec{p} als Ebene durch die Punktemenge \vec{v}_i .

$$f = \sum (\vec{p} \bullet \vec{v}_i)^2 \quad \text{mit} \quad \vec{p} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \quad \text{und} \quad \vec{v}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix}$$

Damit kann das Gleichungssystem 4.1 hergeleitet und gelöst werden um die Ausgleichsebene zu finden.

$$\sum \begin{pmatrix} x_i^2 & x_i y_i & x_i z_i & -x_i \\ x_i y_i & y_i^2 & y_i z_i & -y_i \\ x_i z_i & y_i z_i & z_i^2 & -z_i \\ -x_i & -y_i & -z_i & 1 \end{pmatrix} \times \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{2} \end{pmatrix} \quad (4.1)$$

4.2 GPS Impfung der Bilddaten

VisualSFM bietet die experimentelle Option an, den Suchradius per GPS einzuschränken. Diese Möglichkeit ist sehr interessant, da hiermit insbesondere bei sehr ausgedehnten Arealen die Rechenzeit stark verkürzt werden kann. Die von der Gewässerforschung der HTW Dresden gelieferten Daten enthielten keine GPS-Informationen. Es existierte allerdings eine GPS-Spur mit Zeitstempeln im *Keyhole Markup Language*-Format, welches von Google Earth verwendet wird. Über die Zeitstempel konnten so den Bildern ungefähre GPS Positionen zugeordnet werden. Dann wurden die GPS-Daten in EXIF kompatible Minuten und Sekunden Darstellung umgerechnet und mit der *Apache Commons Imaging*-Bibliothek in die zugehörigen EXIF Felder der Bilddatei geschrieben.

4.3 NVM Datei

VisualSFM speichert die Rekonstruktion in NVM Dateien. Die Datei enthält ASCII codierten Text und eine oder mehrere Punktwolken sowie die Kamerapositionen und Orientierungen in

4 Implementierung

der jeweiligen Szene. Das Format ist einfach und eine Spezifikation auf der Webseite von VisualSFM erhältlich¹. Die Punkte sind in euklidischen Koordinaten, die Orientierungen als Rodriguez Vektor gespeichert. Manchmal trennt VisualSFM die Fließkommawerte mit Komma, andere Male werden Punkte zur Trennung der Nachkommastelle verwendet. Die Ursachen dieses Verhaltens konnten nicht gefunden werden. Eine Vermutung ist, dass ein einer Stelle die Spracheinstellung des Systems verwendet wird und an anderen nicht, weshalb auf einem Rechner mit deutscher Spracheinstellung dann ab und zu mit Kommata getrennt wird. Ein anderes Problem ist die relative Referenzierung der Bilddateien. Die relative Ordnerstruktur darf sich deshalb nicht ändern, sonst werden Daten nicht gefunden. Aus den in der von VisualSFM erzeugten Ausgabedatei enthaltenen Positionen der Punkte lässt sich eine Kartenebene schätzen auf welche die Fotos aus dem Kamerapositionen heraus projiziert werden können.

4.4 Programmierte Module

Der Arbeitsfluss von 3D-Punktwolke zu Orthophoto wurde in zwei Programmen abgebildet. NVMProject erledigt alle geometrischen Berechnungen während das Programm Chopper den Rendervorgang übernimmt. Abbildung 4.2 verdeutlicht den Arbeitsfluss.

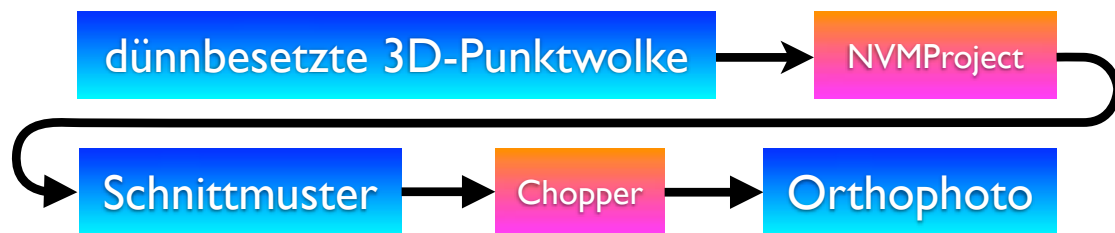


Abbildung 4.2: Die Pipeline vom dünnbesetzten Modell bis zum Orthophoto mit NVMProject und Chopper

4.4.1 NVMProject

Das Programm NVMProject liest eine von VisualSFM geschriebene NVM Datei ein, berechnet die Kartenebene und Projektionen der Kameras auf dieselbige (siehe Abschnitt 4.1) und schreibt die Daten in eine Datei. Eine Beispielsansicht von NVMProject ist in Abbildung 4.3 zu

¹<http://ccwu.me/vsfm/doc.html#nvm>

4 Implementierung

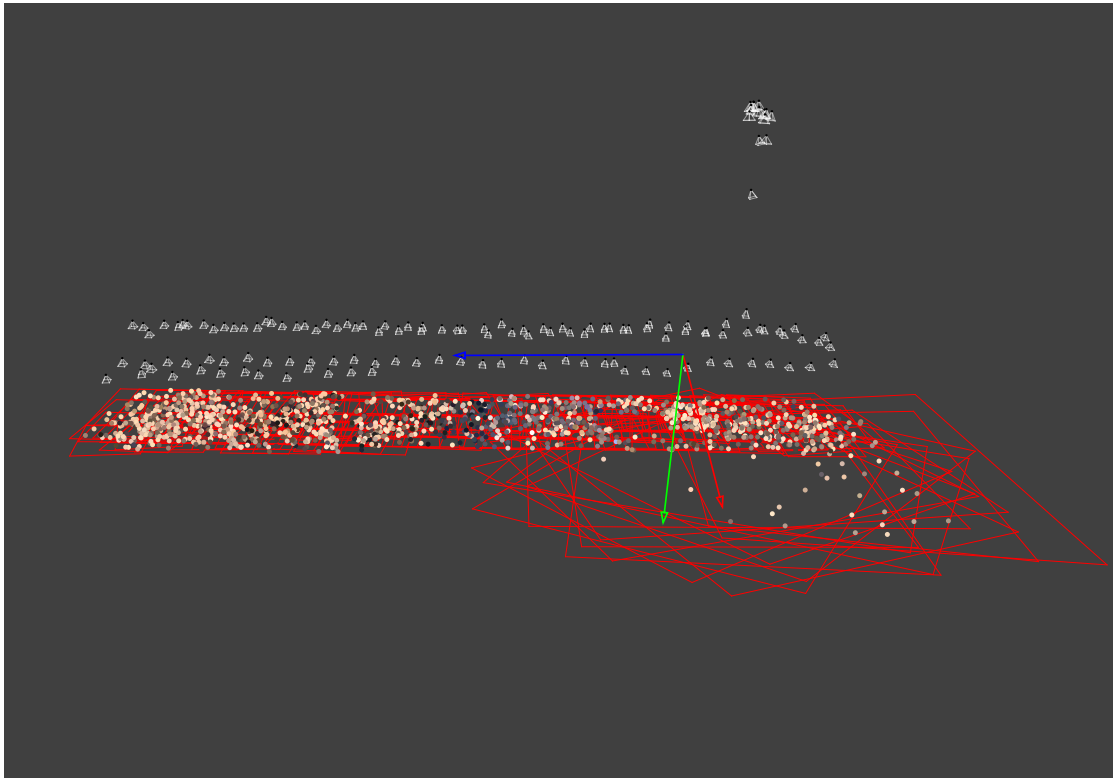


Abbildung 4.3: Die grafische Ausgabe von NVMProject. Die kleinen schwarzen Punkte mit den weißen Pyramiden sind die rekonstruierten Kamerapositionen, die großen farbigen Punkte rekonstruierte 3D-Punkte. Die roten Rahmen stellen die auf die Kartenebene projizierten Kamerabilder dar. Roter, grüner und blauer Pfeil markieren die X-,Y- und Z-Achse des Weltkoordinatensystems.

sehen. Um die Daten im 2D-Fall besser handhaben zu können, wird das Weltkoordinatensystem so transformiert, dass die Kartenebene in der XY-Ebene und die größte Hauptkomponente parallel zur X-Achse sowie die zweitgrößte parallel zur Y-Achse liegt. Da die Bildkoordinaten keine negativen Werte zulassen, wird die Punktemenge so verschoben, dass keinerlei negative Koordinaten mehr vorkommen. Im nächsten Schritt werden die Projektionen der Kamerabilder auf die Ebene gesucht. Mithilfe der Brennweite und Kameraposition können die vier Ecken der Kamerapyramide berechnet werden (siehe Abbildung 4.4). Aus den Eckpunkten und dem Kamerazentrum werden Geradengleichungen aufgesetzt, welche dann mit der Projektionsebene geschnitten werden. Die resultierenden Punkte reichen aus, um eine homographische Zuordnung vom Kamerabild auf die Ebene zu finden. Danach können diese Daten in eine Datei geschrieben werden.

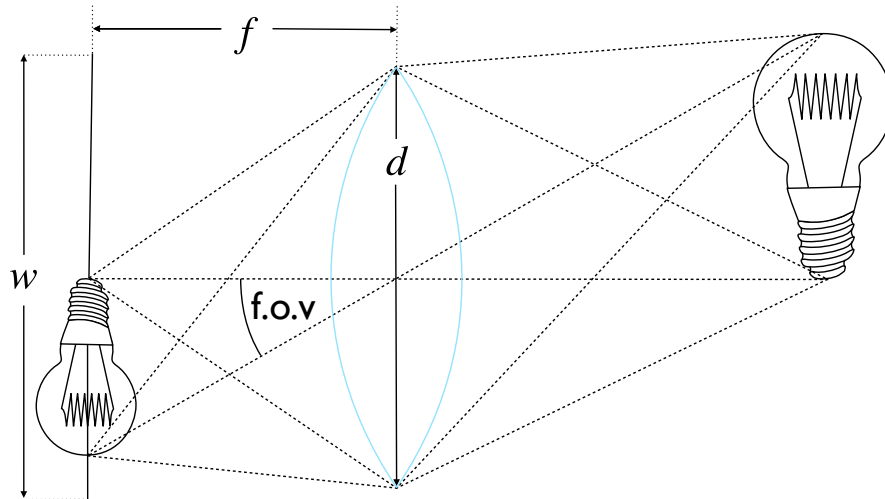


Abbildung 4.4: Die Geometrie eines Objektivs. Über die Brennweite f , Breite w und Öffnungswinkel (f.o.v.) oder Höhe (nicht eingezeichnet) kann die Kamerapyramide berechnet werden.

4.4.2 Chopper

Um die Wahl zwischen den Renderern zu ermöglichen, wurde der Rendervorgang in ein eigenes Programm ausgegliedert. Die erste Version renderte per Software und malte die Bilder sortiert nach dem Abstand der Kamerapositionen zur Kartenebene. Hierbei wurden hohe Objekte nahe des Bildrandes stark verzerrt, da die Kameras häufig nicht parallel zum Nadir zeigten und der Projektionsfehler mit steigendem Winkel zum Nadir zunimmt [28]. In [Abbildung 4.5](#) ist gezeigt

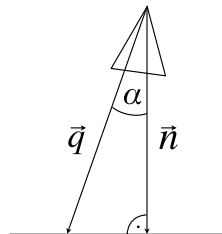


Abbildung 4.5: Der Fehler steigt mit dem Winkel α zum Nadir \vec{n} . Dabei ist die Kameraorientierung unbedeutend.

wie die Formel zustande kommt. Es ist der Winkelabstand vom Lot der Kamera zur Ebene:

$$\alpha = \cos^{-1} \left(\frac{\vec{n} \cdot \vec{q}}{|\vec{n}| \cdot |\vec{q}|} \right)$$

[Abbildung 4.6](#) zeigt den Fehler eines berechneten Orthophotos als Graustufenskala ausgegeben

4 Implementierung

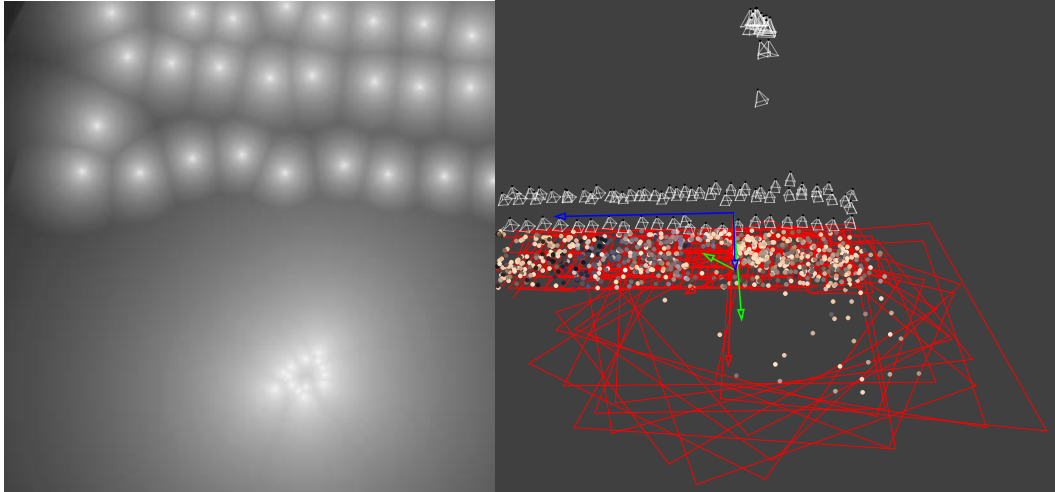


Abbildung 4.6: Links: Ein Kartenausschnitt auf dem der Winkelabstand vom Nadir als Graustufenbild aufgetragen wurde. Je dunkler ein Bereich, desto größer ist der Winkel α . In den weißen Punkten zeigt das Lot zur Projektionsebene genau ins Kamerazentrum. Die ringförmige Struktur im unteren Bereich entsteht, da die Drohne in großer Höhe sich einmal im Kreis fotografierte. Rechts flog der Multicopter systematischer im Tiefflug ab. Rechts: 3D-Ansicht des korrespondierenden Modellausschnitts.

an und ein nach diesem Verfahren gerendeter Kartenausschnitt ist in [Abbildung 4.7](#) zu sehen. Trotz der großen Verbesserung durch die Mischung nach dem Nadirfehler wirkten die Karten verrauscht und unruhig. Dies liegt am Verhältnis der Bildpixel zu den Kartenpixeln. Überdeckt ein Kartenpixel mehrere Bildpixel, müsste er eigentlich mit der Summe der Farben des überdeckenden Bereichs beschrieben werden. Anisotropische Filterung [\[2\]](#) (siehe [Abbildung 4.8](#)) mindert dieses Problem. Es wird dabei eine Palette von geringeren Auflösungen der Textur erstellt und die Farbe des Pixels in der passenden Auflösung nachgesehen. Anisotropische Filterung und die Möglichkeit den Ausschluss der Kartenbereiche per Z-Buffer zu erledigen, führten zur Entscheidung einen Renderer auf Basis von OpenGL zu implementieren.

Die Z-Buffer von Grafikkarten sind darauf ausgelegt, den pixelweisen Ausschluss für die 3D-Darstellung von durchdringenden Geometrien zu berechnen. Wird jedes Kamerabild mit dem Nadirfehler orthogonal zur Projektionsebene extrudiert (siehe [Abbildung 4.9](#)), kann ein Z-Buffer genutzt werden, um den Ausschluss zu berechnen. Die Hinwendung zu einem OpenGL-Renderer zwang zu einigen Änderungen im Design, denn Grafikkarten können keine gigapixelgroßen Dateien in Gänze in ihren Speicher laden. Deshalb müssen so große Bilder in grafikkartengerechte Häppchen unterteilt werden. Doch dadurch tat sich auch eine elegante Möglichkeit der Parallelverarbeitung beim Rendern auf. Unterschiedliche Kacheln müssen bei Schreibzugriff nicht gesperrt werden und auf alle anderen Daten wird nur lesend zugegriffen. Auch beim Laden des

4 Implementierung



Abbildung 4.7: Vorher/Nachher-Vergleich der Mischung nach dem Nadirfehler. Links: Besonders stark verzerrte Teile überdecken Bereiche mit niedrigem Fehler. Rechts: Die Mischung der Bereiche dicht am Nadir liefert ein ansprechendes Ergebnis. (Quelle: HTW Dresden/Geoinformation, GEO-METRIK-Ingenieurgesellschaft mbH Magdeburg)

Bildes ergeben sich dadurch Vorteile: Es müssen nur diejenigen Kacheln geladen werden, welche auch aktuell angezeigt werden. Dadurch kann die benötigte Speichermenge bei Ausschnitten drastisch reduziert werden. Die Zahlenauflösung des Z-Buffers ist abhängig von der Grafikkarte. Je nach Fähigkeiten kann dieser 8, 16 oder 32 Bit tief sein. Wegen der geringen Zahlenauflösung produzierte die Formel bröckelige Strukturen (gezeigt in Abbildung 4.10) an den Rändern. An Stellen mit hohem Fehler ist die Genauigkeit verschwendet, denn häufig wird der Bereich von einem mit einem geringen Fehler verdeckt. Deshalb kann die Genauigkeit im niedrigen Zahlenbereich durch nichtlineare Spreizung der Fehlerfunktion auf Kosten des hohen Fehlerbereiches verbessert werden. Hierfür wird der Fehler zwischen einem relevanten Intervall abgeschnitten und normalisiert. Ein anschließendes Ziehen der Wurzel bringt die erwünschte Spreizung (siehe Abbildung 4.11).

4 Implementierung

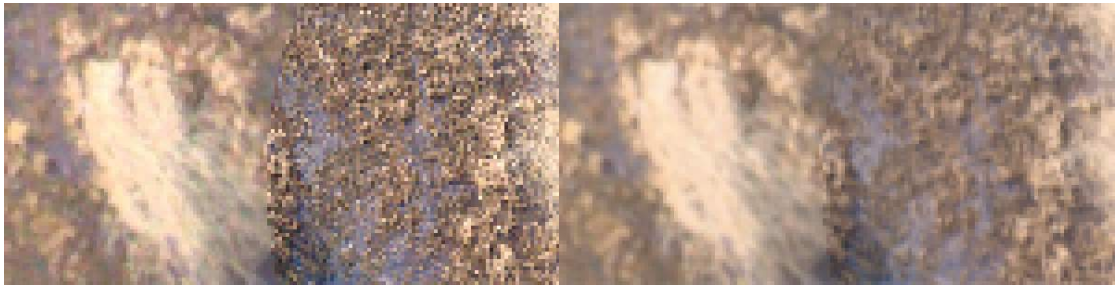


Abbildung 4.8: Die Auswirkungen von anisotroper Filterung am Detail einer Karte. Links: Der Softwarerenderer ohne Filterung lässt Bildteile verrauscht aussehen und Bildgrenzen hervortreten. Rechts: Weichere Grenzen des OpenGL Renderers dank Anisotroper Filterung (Quelle: Institut Gewässerökologie und Binnenfischerei, Deutsches Zentrum für Luft- und Raumfahrt e.V./Optische Informationssysteme, HTW Dresden/-Geoinformation, FU Berlin/ Institut für Weltraumwissenschaften).

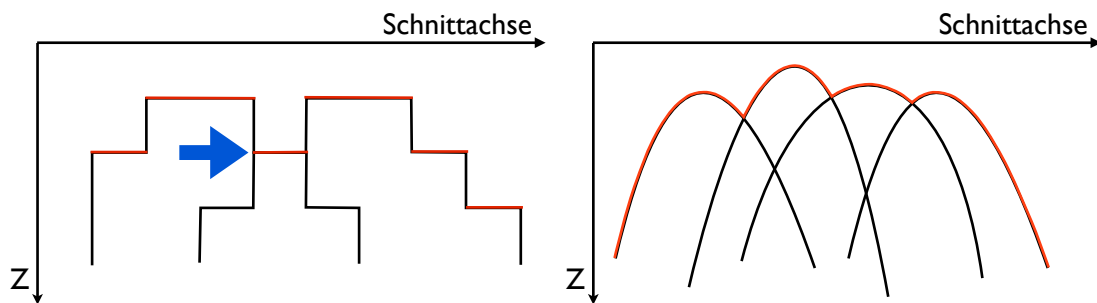


Abbildung 4.9: Ein Schnitt durch die interne Kartengeometrie nach der Extrusion mit dem Nadirfehler (Symbolbild). Die schwarzen Linien stellen die extrudierten Kameraansichten dar, die roten Linien den Bereich mit dem geringsten Fehler. Links zeigt die Ursache für die fransigen Ränder in Abbildung 4.10. Am blauen Pfeil zu sehen: Durch die geringe Zahlenauflösung herrscht Doppeldeutigkeit. Da für das Ergebnis relevante Schnittpunkte geringe Z-Tiefe besitzen, lohnt es sich diesen Zahlenbereich zu spreizen. Rechts: Ein idealer Z-Buffer mit genügend Genauigkeit.

4 Implementierung



Abbildung 4.10: Vorher/Nachher-Vergleich des Randverhaltens beim Ausschluss über Z-Buffer. Links: Bröckelige Ränder und mehrdeutige Tiefenverhältnisse aufgrund der geringen Bit-Tiefe des Z-Buffers. Rechts: Die auf die geringe Bittiefe angepasste Fehlerfunktion erzeugt klare Ränder (Quelle: Institut Gewässerökologie und Binnenfischerei, Deutsches Zentrum für Luft- und Raumfahrt e.V./Optische Informationssysteme, HTW Dresden/Geoinformation, FU Berlin/ Institut für Weltraumwissenschaften)

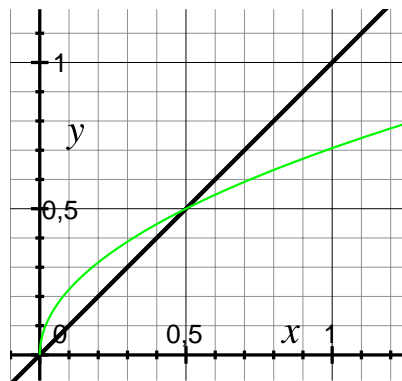


Abbildung 4.11: Die für die Spreizung des Nadirfehlers gewählte Wurzelfunktion (grün) im Vergleich zur Identität (schwarz). Werte zwischen 0 und 0,5 werden wichtiger bewertet, als Werte zwischen 0,5 und 1.

5 Leistungsbewertung

Die Leistungsbewertung ist in drei Teile gegliedert. Der erste Teil betrachtet das Verfahren an sich und die Erkenntnisse, die aus den Experimenten gezogen wurden. Da die Technik des Berechnen der Karte grundsätzlich getrennt vom Ausgleich über alle Farben ist, wird das Ergebnis in Güte der erstellten Karten getrennt betrachtet. Dies geschieht unter dem Gesichtspunkt der Genauigkeit. Der Abschnitt Visueller Eindruck konzentriert sich dagegen auf den globalen Farbausgleich der Einzelbilder.

5.1 Projektion und Geometrie

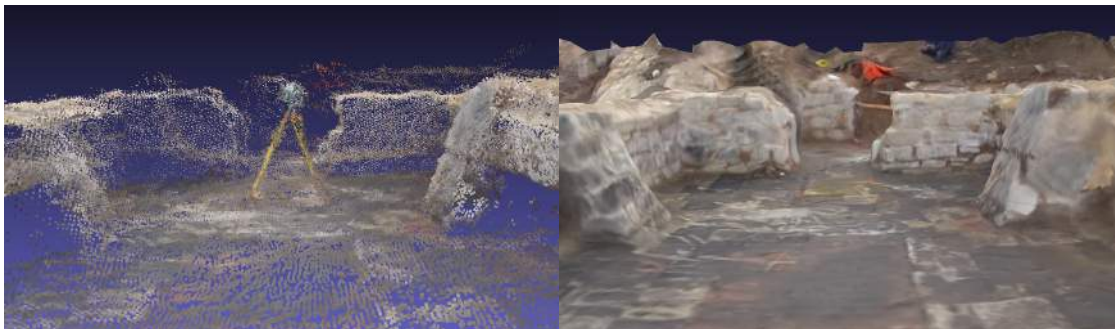


Abbildung 5.1: Links eine mit CMVS/PMVS rekonstruierte dichte Punktwolke, rechts die Poisson Rekonstruktion mit projizierten Texturen. Das links vorhandene Stativ wird nicht korrekt rekonstruiert.

Die für die reine 3D-Rekonstruktion genutzten Methoden sind mathematisch eindeutig. Tests mit synthetischen Daten führten stets zu fehlerlos rekonstruierten Punktwolken. Trotzdem sind auf den rekonstruierten Karten kleinere Artefakte zu erkennen. Die Ursache hierfür ist, dass die Quellen der Ungenauigkeiten schon vor der 3D-Rekonstruktion liegen. Diese sind die Kalibrierung (Verzeichnung) und Merkmalsdetektion (Bildrauschen, Änderungen der Merkmalsposition durch Änderungen in der Perspektive, falsche Zuordnungen). Bei Tests mit der Posenschätzung von OpenCV zeigte sich, dass schon Änderungen der Merkmalspositionen im Subpixelbereich

großen Einfluss auf das Ergebnis haben können. Die mittlere quadratische Abweichung stieg bei synthetischen Daten mit künstlich hinzugefügtem Positionsrauschen auch erwartungsgemäß an. Um die Genauigkeit und Stabilität der gemessenen Merkmalspositionen so gut wie möglich zu gewähren, verwendet der Lokalisationsschritt von SIFT (siehe Abschnitt 2.4.1) mehrere Methoden. So werden immer nur Verteilungen genügend großer Pixelmengen betrachtet und die DoG Daten mit Gaußkernen gefiltert die den Einfluss von Rauschen mindern. Der Punkt bei dem am meisten Potential für Verbesserung besteht, ist die Kamerahardware. Die Verwendung weniger stark verzeichnender Objektive würde sich wahrscheinlich positiv auf die Qualität der Rekonstruktionen auswirken.

Die von Wu für VisualSFM vorgeschlagene Rekonstruktion mit CMVS/PMVS und anschließender Poissonrekonstruktion liefert sehr grobe und stark geglättet aussehende Ergebnisse. Ein Beispiel: Ein in der Szene vorhandenes Stativ wurde gar nicht hergestellt (siehe Abbildung 5.1), da die Struktur zu filigran war. Die Poissonrekonstruktion nutzt ausschließlich die Punktdaten sowie ihre Normalen, um eine wasserdichte Oberfläche zu schätzen. So eignet sich die Poissonrekonstruktion vor allem für Daten, welche per LiDAR-Sensor gewonnen wurden. Diese Systeme können problemlos beliebig viele Punkte mit hoher Genauigkeit messen können. Bei SFM werden aber keine 3D-Punkte gemessen, sondern aus Bildern berechnet. Die Bildinformationen bei der Schätzung einer Oberfläche außen vor zu lassen ist kontraproduktiv. Eine Methode, welche die Information aus den Bilddaten mit einbezieht könnte wahrscheinlich weniger geglättet aussehende Ergebnisse liefern.

5.2 Güte der erstellten Karten

Da für die in der Arbeit verwendeten Daten keine ausreichend genauen GPS Daten verfügbar sind, ist keine verlässliche mathematische Beurteilung der Güte der Karte möglich. Das normale Vorgehen hierbei ist die mittlere quadratische Abweichung zu berechnen. Eventuell könnten die Fotos mit der aufgezeichneten Flugbahn eingepasst werden. Ein Vergleich des erstellten Orthophotos mit einem Satellitenfoto von Google Earth¹ (siehe Abbildung 5.2) brachte jedoch keine größeren Ungereimtheiten zu Tage.

¹<http://www.google.com/earth/>

5.3 Visueller Eindruck

Der Farbausgleich zwischen den Bildern liefert auf synthetischen Testdaten mit globalen Änderungen der Farbe gute Resultate (siehe Abbildung 5.3). Auch auf den Realdaten konnte das Verfahren eine Besserung der Ergebnisse erreichen, hinterließ aber trotzdem deutlich erkennbare Bildgrenzen (siehe Abbildung 5.5). Die Bilder des Helsunger Bruchs wurden an einem wolkenigen Tag aufgenommen und dementsprechend schwankte die Beleuchtungssituation stark (siehe Abbildung 3.4).

Das affine Modell der Farbraumanpassung liefert eine Transformation des Farbraums. Diese kann Translation, Rotation, Skalierung und Scherung beinhalten. Wird der Farbraum als 3D-Histogramm visualisiert (siehe Abbildung 5.4a), ist zu ersehen, dass die Farbverteilungen teilweise sehr eintönig sind und nur einen schmalen Bereich des RGB-Würfels abdecken. Die zu erkennenden zigarrenförmigen Objekte liefern in Richtung der zweiten und dritten Hauptkomponente nur sehr eng beieinanderliegende Stützpunkte für die Berechnung der Transformation, was dazu führt dass die Farbräume nach zu häufiger Anwendung des Farbausgleichs zu Linien degenerieren (siehe 5.4b).

Die Kompression auf eine Linie wird vor allem durch die erlaubte Scherung und Skalierung verursacht. Scherung und Skalierung zu verbieten könnte diese Probleme lösen, hätte dann aber auch eine schlechtere Farbanpassung untereinander zur Folge.

5 Leistungsbewertung

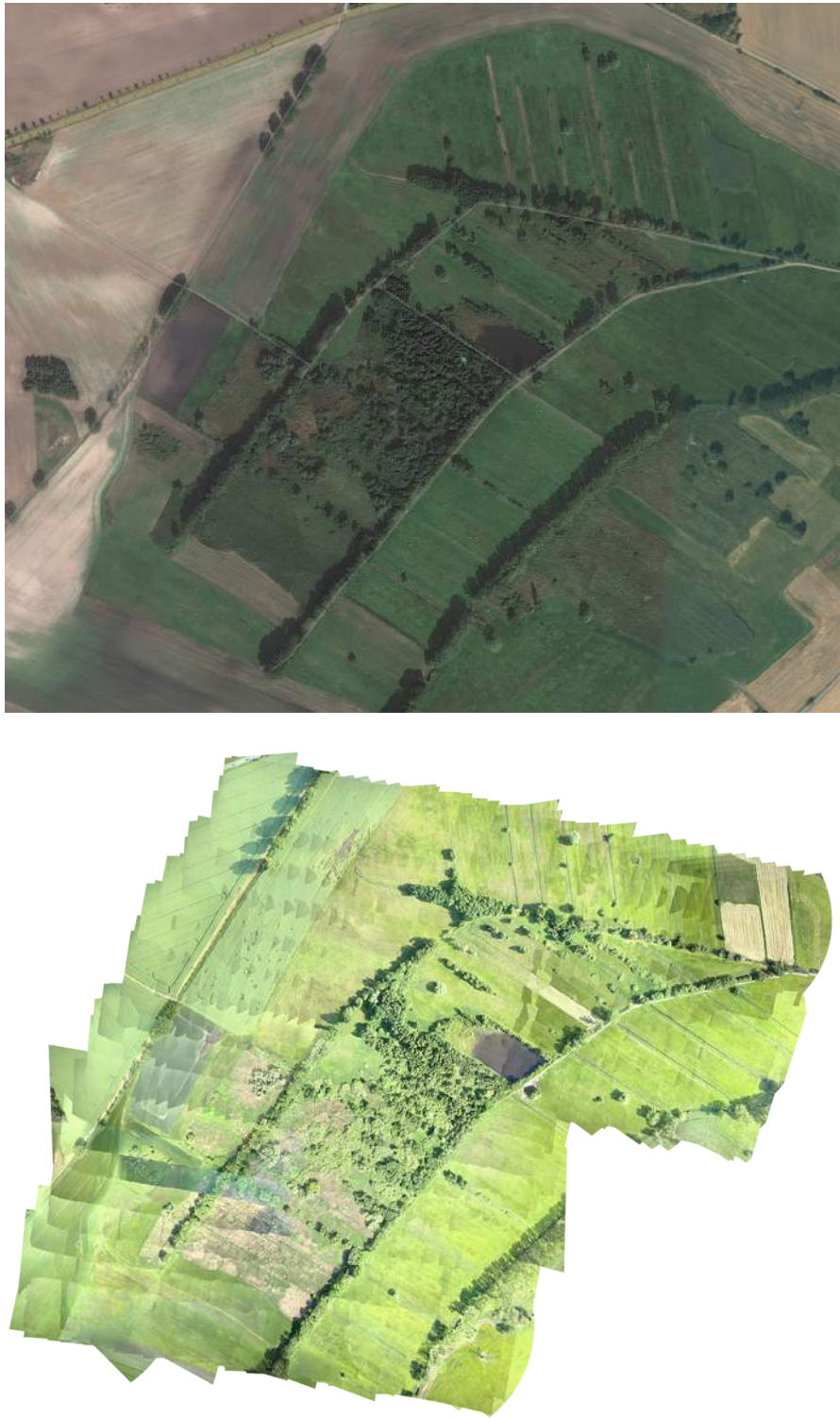


Abbildung 5.2: Der Vergleich einer Satellitenansicht (oben, Quelle Google Earth) mit einem erstellten Orthophoto des Helsenberger Bruchs (unten, Quelle: Institut Gewässerökologie und Binnenfischerei, Deutsches Zentrum für Luft- und Raumfahrt e.V./Optische Informationssysteme, HTW Dresden/Geoinformation, FU Berlin/ Institut für Weltraumwissenschaften)

5 Leistungsbewertung

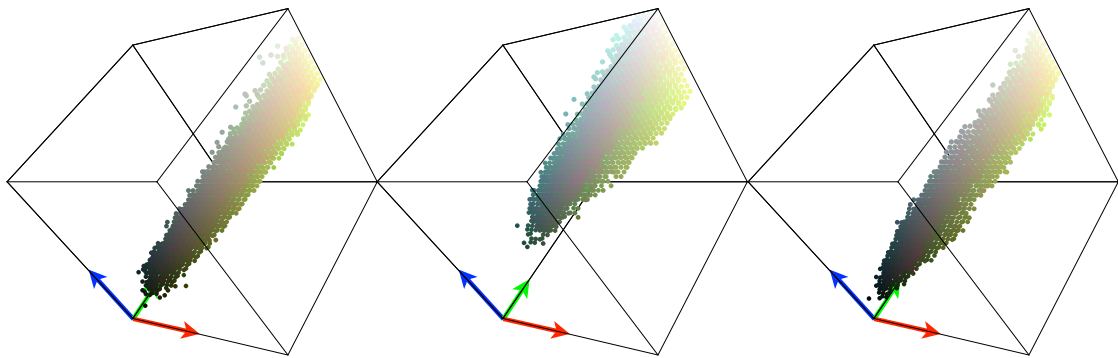


(a) Bilder vor dem Farbausgleich

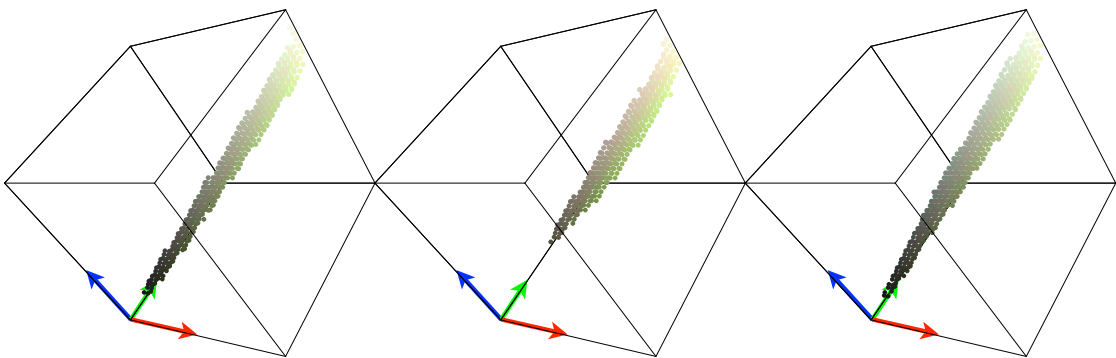


(b) nach dem Farbausgleich

Abbildung 5.3: Ergebnis des Farbausgleichs auf synthetischen Testdaten. Der Farbausgleich produziert hier gute Resultate.



(a) vor dem Farbausgleich



(b) nach mehrmaligem Farbausgleich

Abbildung 5.4: 3D-Histogramme von korrespondierenden Ausschnitten aus dem Helsingens Datensatz. Für jede vorkommende Farbe wird der zugehörige Punkt im Farbwürfel bemalt. Die Degeneration des Farbraums in Richtung einer Linie ist deutlich zu erkennen.

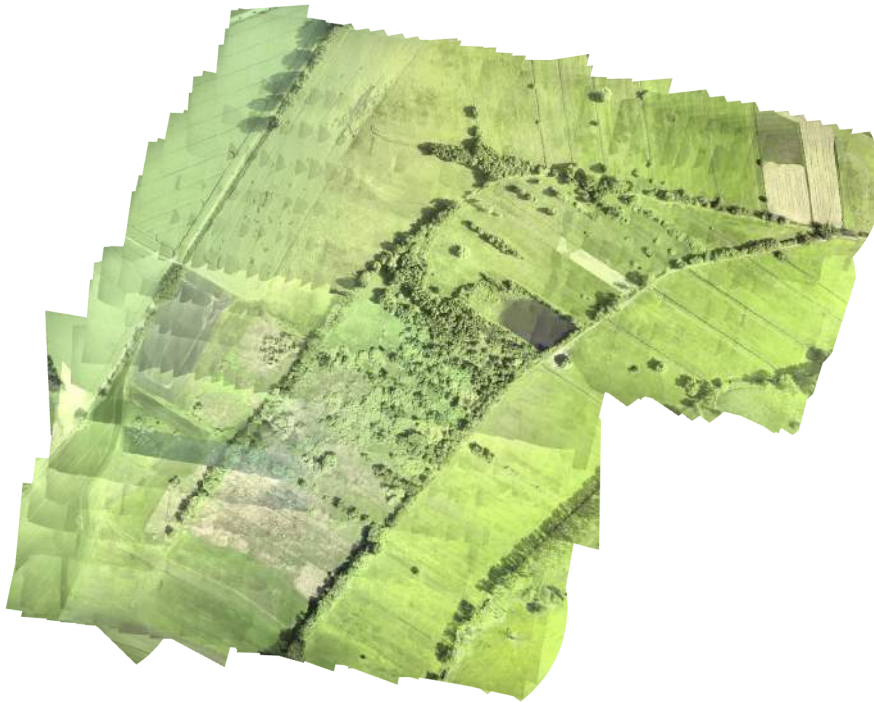


Abbildung 5.5: Der Helsingendatensatz nach 50 Iterationen des Farbausgleichs. Besonders gut sind die Auswirkungen des Ausgleichs im Nordosten zu erkennen. Die Farben sind dort erkennbar weniger unterschiedlich als in Abbildung 5.2 unten (Quelle: Institut Gewässerökologie und Binnenfischerei, Deutsches Zentrum für Luft- und Raumfahrt e.V./Optische Informationssysteme, HTW Dresden/Geoinformation, FU Berlin/ Institut für Weltraumwissenschaften).

6 Schlußfolgerungen, Fragen und Ausblicke

Das Verfahren bietet verschiedene Ansätze der Weiterentwicklung. Einige betreffen bestehende verwendete Algorithmen, andere neue Fähigkeiten wie die Registrierung der Karte in Weltkoordinaten oder das Anzapfen von anderen Informationsquellen als VisualSFM. Dieses Kapitel soll die wichtigsten davon kurz vorstellen.

6.1 Die Geometrie

Die benutzte Mischung nach dem Nadirfehler erzeugt wie andere Verfahren *double mapping*-Effekte an Stellen, deren Höhe nicht mit der Kartenebene übereinstimmt. Eine vielversprechende Möglichkeit, das *double mapping*-Problem zu beheben und geometrisch konsistente Orthophotos zu berechnen, wäre mittels Multiview-Stereo-Ansatz (siehe Abschnitt 2.4.4). Eine Mischung der π -Ebenen nach der geringsten Farbvarianz und Z-Tiefe könnte ansprechende Resultate und gleichzeitig auch die Höhenlinien der Umgebung liefern. Hier können natürlich neue Probleme auftreten und Schwierigkeiten bei nicht ausreichend texturierten Bereichen. Ein weiteres anzugehendes Problem ist die Behandlung von großen Arealen, bei denen die Erdkrümmung miteinbezogen werden muss. Die aktuelle Version geht davon aus, dass die Erde eine Scheibe ist.

6.2 Der Farbausgleich

Ein weiteres interessantes Gebiet ist der Farbausgleich über viele Bilder eines Panoramas. Bilder überschneiden sich nur teilweise und Änderungen müssen über mehrere Bilder hinweg propagiert werden (Bild A überdeckt einen Teil von Bild B, B überdeckt einen Teil von C, aber A und C besitzen keinen gemeinsamen Überdeckungsbereich). Eine globale Optimierung wie Bündelblockausgleiche (siehe Abschnitt 2.4.3) könnte auch auf die Farben angewendet werden.

Der Farbtransfer übernimmt die Rolle der Reprojektionsfunktion, der endgültige Farbwert die Rolle der 3D-Punktposition und die Merkmalsfarbe die Rolle der Merkmalsposition. Minimiert werden die Farbabstände der Merkmale zum endgültigen Farbwert. Schätzungen für den Farbwert könnten aus den Mittelwerten der Merkmalsfarben gewonnen werden. HaCohen et al. [41] verwenden neben einer neuartigen Technik namens *Non-rigid Dense Correspondence* um korrespondierende Pixelbereiche zu finden, ein Farbtransfermodell auf Basis der kumulativen Verteilungsfunktion. Statt diskrete Berechnungen zu verwenden, nähern die Autoren die Funktion durch stückweise quadratische Splines an. Dabei ist das Ziel jedoch nicht das globale Optimum zu finden, welches alle Bilder mit kleinstmöglicher Energie annähert, sondern die Bilder an eine vom Benutzer gewählten Farbstimmung anzupassen. Für dieses gewählte Bild könnte ein Bild mit besonders guter Dynamik aus der Ausgangsmenge ausgewählt werden. Durch das Modell der angenäherten kumulativen Verteilungsfunktion könnten die Farbunterschiede eventuell besser angepasst werden als mit dem affinen Farbmodell.

6.3 Geocoding

Die Karten bieten ohne Einbettung in das Weltkoordinatensystem zwar einen lokalen Bezug, insbesondere für die Luftbildarchäologie ist allerdings ein globaler Bezug hilfreich. Wie sonst sollten die auf den Luftbildaufnahmen entdeckten potentiellen Ausgrabungsstätten auf dem Boden wiedergefunden werden, wenn keine genauen Positionen bekannt sind? Deshalb ist es wichtig die Karte mit einer genauen Positionierung in Weltkoordinaten zu versehen. Dies kann anhand von am Boden vermessenen Markern geschehen oder, falls keine Marker beim Überflug fotografiert wurden, auch nachträglich anhand markanter Landschaftsmerkmale wie Straßenkreuzungen oder anderen Landschaftsmarken. Durch die große Rechenzeiteinsparung bei großen Gebieten lohnt sich der Einsatz von GPS-Technik auf der eingesetzten Flugplattform, auch wenn die Positionen zum erstellen der Karten nicht zwingend notwendig sind.

6.4 Echtzeiteinsatz

Da das implementierte Verfahren nur die Kamerapositionen sowie einige rekonstruierte 3D-Punkte benötigt, kann es an Echtzeit-Mapping Systeme wie z.B. PTAM [40] von Klein et al. angebunden werden und könnte so einem Piloten sofort beim Überflug eine Karte der Umgebung zeigen. Es könnten auch mehrere Multicopter gleichzeitig ein Gebiet überfliegen und einzelne Kacheln bearbeiten. So könnte eine Karte ständig aktuell gehalten werden, solange neues

6 Schlußfolgerungen, Fragen und Ausblicke

Bildmaterial der Umgebung geliefert wird. Eine interessante Anwendung für den Katastrophenschutz, welcher auch auf aktuelles Kartenmaterial angewiesen ist.

7 Zusammenfassung

In dieser Arbeit wurde der Weg von Bilddaten zu einem fertigen Orthophoto anhand visueller Merkmale betrachtet. Gezeigt wurde, dass es möglich ist Orthophotos ausschließlich anhand von visuellen Informationen zusammenzufügen. GPS Daten wurden nur zur Rechenzeitverkürzung verwendet.

Dies beinhaltet:

- Erarbeitung der mathematischen Grundlagen der projektiven Geometrie (siehe Abschnitte [2.3.2](#), [2.3.3](#)) und Computergrafik (siehe Abschnitt [2.3.1](#))
- Erarbeitung der algorithmischen Grundlagen auf dem Gebiet der Computer-Vision wie Merkmale (siehe Abschnitt [2.4.1](#)) und RANSAC
- Qualitative Experimente zur Posenschätzung, zum Farbausgleich über große Bildmengen, Merkmalsextraktion und -zuordnung und 3D-Rekonstruktion
- Erarbeitung eines Workflows von Fotos zum fertigen, texturierten 3D-Modell auf Basis frei verfügbarer Software
- Implementation einer mit OpenCV parameterkompatiblen Verzeichniskorrektur (siehe Abschnitt [2.2.3](#))
- Entwurf eines Farbausgleichs von beliebig vielen Bildern auf Basis von Matrixtransformationen und Merkmalen (siehe Abschnitt [2.4.6](#))
- Implementation von Importern für PLY- und NVM-Dateien
- Implementation einer Software für die Extraktion des Kartenlayouts aus einem 3D-Modell (siehe Abschnitt [4.4.1](#))
- Implementation eines Softwarerenderers für große Orthophotos

7 Zusammenfassung

- Implementation eines OpenGL-Renderers für große Orthophotos (siehe Abschnitt 4.4.2)
- Implementation einer Anzeigesoftware, welche speicherschonend mit Kacheln das Anzeigen von großen Bildern ermöglicht
- Erstellung von texturierteren 3D-Modellen für das Projekt Archäocopter
- Erstellung von Orthophotos für die Gewässerforschung der HTW Dresden und das Projekt Archäocopter

Literaturverzeichnis

- [1] Li, Yan und Ronald Briggs: *Automatic extraction of roads from high resolution aerial and satellite images with heavy noise*. World Academy of Science, Engineering and Technology, 54:416–422, 2009.
- [2] Szeliski, Richard: *Computer vision: algorithms and applications*. Springer, 2011.
- [3] Mitsunaga, Tomoo und Shree K Nayar: *Radiometric self calibration*. In: *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, Band 1. IEEE, 1999.
- [4] Reinhard, Erik, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward und Karol Myszkowski: *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010.
- [5] Brown, Duane C: *Decentering distortion of lenses*. Photometric Engineering, 32(3):444–462, 1966.
- [6] Hartley, Richard und Andrew Zisserman: *Multiple view geometry in computer vision*, Band 2. Cambridge Univ Press, 2000.
- [7] Faugeras, Olivier D und Quang Tuang Luong: *The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications*. MIT press, 2004.
- [8] Enroth-Cugell, Christina und John G Robson: *The contrast sensitivity of retinal ganglion cells of the cat*. The Journal of physiology, 187(3):517–552, 1966.
- [9] Lowe, David G: *Distinctive image features from scale-invariant keypoints*. International journal of computer vision, 60(2):91–110, 2004.
- [10] Bay, Herbert, Andreas Ess, Tinne Tuytelaars und Luc Van Gool: *Speeded-up robust features (SURF)*. Computer vision and image understanding, 110(3):346–359, 2008.

- [11] Dreuw, Philippe, Pascal Steingrube, Harald Hanselmann, Hermann Ney und G Aachen: *SURF-Face: Face Recognition Under Viewpoint Consistency Constraints*. In: *BMVC*, Seiten 1–11, 2009.
- [12] Maji, Subhransu: *A Comparison of Feature Descriptors*. University of California, Berkeley, 2006.
- [13] Mikolajczyk, Krystian, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir und Luc Van Gool: *A comparison of affine region detectors*. *International journal of computer vision*, 65(1-2):43–72, 2005.
- [14] Shi, Jianbo und Carlo Tomasi: *Good features to track*. In: *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, Seiten 593–600. IEEE, 1994.
- [15] Rosten, Edward und Tom Drummond: *Machine learning for high-speed corner detection*. In: *Computer Vision–ECCV 2006*, Seiten 430–443. Springer, 2006.
- [16] Rublee, Ethan, Vincent Rabaud, Kurt Konolige und Gary Bradski: *ORB: an efficient alternative to SIFT or SURF*. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*, Seiten 2564–2571. IEEE, 2011.
- [17] Alahi, Alexandre, Raphael Ortiz und Pierre Vandergheynst: *Freak: Fast retina keypoint*. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, Seiten 510–517. IEEE, 2012.
- [18] Leutenegger, Stefan, Margarita Chli und Roland Y Siegwart: *BRISK: Binary robust invariant scalable keypoints*. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*, Seiten 2548–2555. IEEE, 2011.
- [19] Fischler, Martin A und Robert C Bolles: *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*. *Communications of the ACM*, 24(6):381–395, 1981.
- [20] Triggs, Bill, Philip F McLauchlan, Richard I Hartley und Andrew W Fitzgibbon: *Bundle adjustment a modern synthesis*. In: *Vision algorithms: theory and practice*, Seiten 298–372. Springer, 2000.
- [21] Zhang, Jeffrey, Daniel G Aliaga, Mireille Boutin und Robert Insley: *Angle independent bundle adjustment refinement*. In: *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, Seiten 1108–1116. IEEE, 2006.

- [22] Mouragnon, E, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser und Patrick Sayd: *Generic and real-time structure from motion using local bundle adjustment*. Image and Vision Computing, 27(8):1178–1193, 2009.
- [23] Lourakis, Manolis IA und Antonis A Argyros: *SBA: A software package for generic sparse bundle adjustment*. ACM Transactions on Mathematical Software (TOMS), 36(1):2, 2009.
- [24] Khan, Saad M, Pingkun Yan und Mubarak Shah: *A homographic framework for the fusion of multi-view silhouettes*. In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Seiten 1–8. IEEE, 2007.
- [25] Tian, Gui Yun, Duke Gledhill, David Taylor und David Clarke: *Colour correction for panoramic imaging*. In: *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, Seiten 483–488. IEEE, 2002.
- [26] Wu, Changchang: *Towards linear-time incremental structure from motion*. In: *3DTV-Conference, 2013 International Conference on*, Seiten 127–134. IEEE, 2013.
- [27] Kuzmin, Yevgeniy P, Sergey A Korytnik und Orrin Long: *Polygon-based true orthophoto generation*. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 35(Part 3):529–531, 2004.
- [28] Amhar, Fahmi, Josef Jansa, Christine Ries *et al.*: *The generation of true orthophotos using a 3D building model in conjunction with a conventional DTM*. International Archives of Photogrammetry and Remote Sensing, 32:16–22, 1998.
- [29] Mayr, Werner und Christian Heipke: *A contribution to digital orthophoto generation*. International Archives of Photogrammetry and Remote Sensing, 27(B11):430–439, 1988.
- [30] Rau, Jiann Yeou, Nai Yu Chen und Liang Chien Chen: *True orthophoto generation of built-up areas using multi-view images*. Photogrammetric Engineering and Remote Sensing, 68(6):581–588, 2002.
- [31] Zebedin, Lukas, Andreas Klaus, Barbara Gruber-Geymayer und Konrad Karner: *Towards 3D map generation from digital aerial images*. ISPRS Journal of Photogrammetry and Remote Sensing, 60(6):413–427, 2006.
- [32] Kocaman, S, L Zhang, A Gruen und D Poli: *3D city modeling from high-resolution satellite images*. In: *Proceedings of ISPRS Workshop on Topographic Mapping from Space 2006, Proceedings in CD-ROM*. Citeseer, 2006.

- [33] Morgan, Dennis und Edgar Falkner: *Aerial mapping: methods and applications*. CRC Press, 2010.
- [34] Skarlatos, D: *Orthophotograph production in urban areas*. The Photogrammetric Record, 16(94):643–650, 1999.
- [35] Schickler, Wolfgang und Anthony Thorpe: *Operational procedure for automatic true orthophoto generation*. International Archives of Photogrammetry and Remote Sensing, 32:527–532, 1998.
- [36] Neitzel, Frank und J Klonowski: *Mobile 3D mapping with a low-cost UAV system*. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci, 38:1–6, 2011.
- [37] Furukawa, Yasutaka und Jean Ponce: *Accurate, dense, and robust multiview stereopsis*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32(8):1362–1376, 2010.
- [38] Kazhdan, Michael, Matthew Bolitho und Hugues Hoppe: *Poisson surface reconstruction*. In: *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.
- [39] Zhou, Qian Yi und Vladlen Koltun: *Dense scene reconstruction with points of interest*. ACM Transactions on Graphics (TOG), 32(4):112, 2013.
- [40] Klein, Georg und David Murray: *Parallel tracking and mapping for small AR workspaces*. In: *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, Seiten 225–234. IEEE, 2007.
- [41] HaCohen, Yoav, Eli Shechtman, Dan B Goldman und Dani Lischinski: *Optimizing color consistency in photo collections*. ACM Transactions on Graphics (TOG), 32(4):38, 2013.
- [42] Xu, Wei und Jane Mulligan: *Performance evaluation of color correction approaches for automatic multi-view image and video stitching*. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, Seiten 263–270. IEEE, 2010.