



Hochschule Neubrandenburg
University of Applied Sciences

Diplomarbeit

„Automatisierung einer digitalen
Orthophotoproduktion“

eingereicht von:

Matthias Mann

Gutachter:

Prof. Dr.-Ing. Andreas Wehrenpfennig

Dr. Holger Eichstädt

urn:nbn:de:gbv:519-thesis2008-0118-6

02. März 2009

Eidesstattliche Erklärung

Ich erkläre an Eides Statt, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken habe ich als solche kenntlich gemacht.

02.03.2009

Vorwort

An dieser Stelle möchte ich mich bei den Menschen bedanken, die mir das Studium und die Ausarbeitung dieser Diplomarbeit ermöglicht haben.

Großer Dank gilt meiner Familie für die moralische Unterstützung und die Motivation zur Vollendung dieser Arbeit.

Ebenso danke ich allen Professoren und Mitarbeitern der Hochschule Neubrandenburg für die fachliche Ausbildung. Besonderer Dank gilt meinem Betreuer Herrn Prof. Dr. Wehrenpfennig und Frau Schuldt, die immer ihre Unterstützung angeboten haben.

Den Mitarbeitern der IFMS und dem Geschäftsführer Dr. Eichstädt danke ich für die Zusammenarbeit und die Hilfe bei der Erstellung dieser Arbeit.

Inhaltsverzeichnis

Abkürzungsverzeichnis	8
1. Einleitung / Überblick.....	11
1.1. Definitionen.....	12
1.2 Arbeitsumgebung und Technische Grundlagen im Unternehmen	18
2. Problemstellung/Aufgabenstellung:	22
3. Allgemeine technische Methoden zur Umsetzung.....	23
3.1 Client – Server – Modell	23
3.2 CORBA (Common Object Request Broker Architecture)	23
3.3 Programmiersprache C++	24
3.4 Stapelverarbeitung	24
3.5 ERDAS Makrosprache EML	25
3.6 Arcforest Coresystem	26
3.7 Datenbank SQL Server.....	27
4. Realisierung	28
4.1 Ist– Situation	28
4.2 Vorüberlegungen.....	32
4.3 Lösungsansatz	33
4.3.1 Untersuchung zum Aufbau eines Bildblockes	34
4.4 Konzeption	38
4.4.1 Anwendungsfälle.....	38
4.4.2 Benötigte Module.....	39
4.4.3 Bildblöcke erstellen.....	40
4.4.3.1 Vorlage erstellen	40
4.4.3.2 Modul „BlockFileCreator“	42
4.4.4 Modul „DTMExtractor“	43
4.4.5 Modul „OrthoResampler“	44
4.5 Umsetzung	45
4.5.1 Datenbankeinträge.....	45
4.5.2 Modul „BlockFileCreator“	48
4.5.2.1 Modul „BlockFileCreator“ aufrufen	50

4.5.3 Modul „DTMExtractor“	51
4.5.3.1 Modul „DTMExtractor“ aufrufen	54
4.5.4 Modul „OrthoResampler“	55
4.5.4.1 Modul „OrthoResampler“ aufrufen.....	56
5. Vergleich der Ergebnisse.....	57
5.1 <i>Qualitativer Vergleich</i>	57
5.2 <i>Quantitativer Vergleich</i>	58
6. Zusammenfassung und Ausblick.....	61
Anhang	62
Quellenverzeichnis	66

Abbildungsverzeichnis

Abbildung 1.1: transversaler Schnittzylinder.....	13
Abbildung 1.2: schematische Darstellung einer Luftbildentzerrung	14
Abbildung 1.3: DTM (mit Höhenlinien).....	15
Abbildung 1.4: Grundstruktur Tiff.....	16
Abbildung 1.5: IFD Aufbau	16
Abbildung 1.6: mögliche Anordnungen.....	17
Abbildung 1.7: Bildflug	18
Abbildung 1.8: UltraCam - D.....	19
Abbildung 3.1: Client-Server-Prinzip	23
Abbildung 4.1: LPS Bildblock.....	30
Abbildung 4.2: LPS Ortho-Resampling.....	31
Abbildung 4.3: identische Einträge in der hfa-Struktur zweier Stereobildblöcke	35
Abbildung 4.4: identische Einträge in der hfa-Struktur zweier Stereobildblöcke	36
Abbildung 4.5: Anwendungsfälle	38
Abbildung 4.6: Ablauf der Module.....	39
Abbildung 4.7: Ansicht eines Stereobildblockes im LPS - GUI.....	40
Abbildung 4.8: Aktivitäten zur Bildblockerstellung.....	42
Abbildung 4.9: Aktivitäten zur DTM-Berechnung.....	43
Abbildung 4.10: Aktivitäten zum OrthoResampling	44
Abbildung 4.11: Auszug aus der EHFA Struktur	49
Abbildung 4.12: Anwendungsaufruf.....	50
Abbildung 4.13: Definieren von Variablen.....	51
Abbildung 4.14: Anwendungsaufruf mit Variablen.....	51
Abbildung 4.15: Auszug aus dtm-relevanten Objektknoten	52
Abbildung 4.16: Befehlssyntax beim resampleprocess	55
Abbildung 4.17: Syntax des tempdem-file.....	55
Abbildung 4.18: Befehlssyntax der OrthoResampler.bat	56
Abbildung 4.19: Befehlssyntax der OrthoResampler.bat	56
Abbildung 5.1: Vergleich - Zeitaufwand	58
Abbildung 5.2: herkömmlich Methode.....	59
Abbildung 5.3: automatisierte Methode.....	59

Tabellenverzeichnis

Tabelle 4.1: Eingabeparameter.....	37
Tabelle 4.2: Auszug aus der Tabelle „ps_process_list“	46
Tabelle 4.3: Auszug aus der Tabelle „ps_process_child_map“	46
Tabelle 4.4: Auszug aus der Tabelle „ps_photo_process_list“	46
Tabelle 4.5: Tabelle „status_list“	47
Tabelle 4.6: Auszug aus der Tabelle „ps_photo_process_child_map“	47

Abkürzungsverzeichnis:

Abkürzung	–	Bedeutung
2D	–	Zweidimensional
3D	–	Dreidimensional
ALK	–	Automatisierte Liegenschaftskarte
ASCII	–	American-Standard-Code-of-Information_Interchange
ATE	–	Automatic Terrain Extraction
CCD	–	Charge-coupled Device (Einsatz in Bildsensoren)
CORBA	–	Common Object Request Broker Architecture
CPU	–	Central Processing Unit
DGM	–	Digitales Geländemodell
DHM	–	Digitales Höhenmodell
DOP	–	Digitales OrthoPhoto
DSM	–	Digitales Situationsmodell
DTM	–	Digital Terrain Model
EML	–	ERDAS Macro Language
GIS	–	Geographisches Informationssystem
GPS	–	Global Positioning System
GSD	–	Ground Sample Distance (Pixel Size)
GUI	–	Graphical User Interface
HFA	–	Hierarchical File Acces
IDL	–	Interface Definition Language
IFD	–	Image File Directory

IFMS	–	International Forestry Modelling Systems-GmbH & Co. KG
INS	–	Inertiales Navigationssystem
LPS	–	Leica Photogrammetry Suite
MB	–	MegaByte
MS	–	Microsoft
MSU	–	Mobile Storage Unit
OMG	–	Object Management Group
OPC	–	Office Processing Center
PC	–	Personal Computer
PDA	–	Personal Digital Assistant
RTS	–	Real Time Systems
SAN	–	Storage Area Network
SQL	–	Structured Query Language
TIFF	–	Tagged Image File Format
TIN	–	Triangulated Irregular Network
VBA	–	Visual Basic for Application

Kurzfassung:

Die grundlegende Idee dieser Diplomarbeit ist die Erleichterung der Arbeit im Unternehmen IFMS. Das Ziel ist die Entwicklung eines Werkzeugs mit dem die Abarbeitung photogrammetrischer Aufgaben vereinfacht wird. Hauptsächlich sollen die zahlreichen händischen Arbeitsschritte minimiert werden um dadurch lange Arbeitszeiten zu verkürzen.

Vorgehensweise:

Um die gestellte Aufgabe zu lösen werden zuerst die grundlegenden Begriffe und Vorgänge beschrieben. Das bedeutet, dass Schlagwörter definiert und wichtige Handlungsabläufe betrachtet werden. Dann wird die Ist-Situation des Unternehmens vorgestellt. Anschließend wird geschildert wie die geforderte Soll-Situation erreicht wurde. Hier wird auf die Ideen und deren Umsetzung detailliert eingegangen. Ein Vorher – Nachher - Vergleich zeigt dann die Unterschiede auf.

1. Einleitung / Überblick

„Von alters her versuchen die Menschen ihre Umgebung mittels Karten zu beschreiben und dies für die eigene Orientierung zu nutzen. Was in der Vergangenheit Steinzeichnungen, Papyrusrollen und später Papierkarten leisteten, haben heute digitale Karten und Geoinformationen übernommen.

Durch die Möglichkeiten der modernen Informationstechnologien können unabhängig vom Ort, an dem man sich gerade befindet, die notwendigen räumlichen Informationen per Computer, PDA oder Handy zur Verfügung gestellt werden. Ob Routenplaner, Navigationssystem im Auto, Wetterkarten im Internet, satellitengestütztes Mautsystem oder touristischer Handy-Reiseführer: Überall werden heute digitale raumbezogene Informationen verwendet.

Der gesellschaftliche Stellenwert von Geoinformationen wächst ständig und ist kaum noch aus unserem Alltags- und Berufsleben wegzudenken. So bilden Geoinformationen in 80 Prozent aller Geschäftsabläufe in unserem Berufsleben die Grundlagen für Entscheidungen.

Die Vielfalt von Geoinformationen erstreckt sich dabei von der einfachen Koordinate und Adresse über unterschiedlichste digitale Karten, 3D-Stadt- und Geländemodelle bis hin zu Luft- und Satellitenbildern.“ [LTA]

Dieser Auszug aus dem LandesTechnologieAnzeiger 1/2008 beschreibt sehr treffend die momentane Marktsituation. Mit der Gewinnung und Weiterverarbeitung von Geodaten befasst sich die IFMS. Um am hart umkämpften, internationalen Geoinformationsmarkt bestehen zu können, wird versucht, die Kunden möglichst zeitnah mit Daten zu versorgen.

Zur Produktpalette gehören neben senkrecht aufgenommenen Luftbildern: Höhenmodelle, entzerrte, georeferenzierte Orthophotos und Orthophotomosaike. In diesem Kapitel sollen bezüglich der Produktionskette relevante, grundlegende Begriffe erläutert werden.

1.1. Definitionen

Kartenprojektion

Eine Kartenprojektion ist eine von vielen verschiedenen Methoden in der Kartographie, mit der man die gekrümmte Oberfläche der (dreidimensionalen) Erde auf die flache (zweidimensionale) Karte überträgt. Dieser Prozess der Modellbildung geschieht mit Hilfe von Abbildungsvorschriften, die man mathematisch ausdrücken kann. Kartenprojektionen bilden eine spezielle Gattung von Koordinaten-Referenz-Systemen und stellen dabei eine Konversionsmethode von einem mathematischen Erdmodell in die Ebene dar. Aufgrund der Krümmung der Erde in alle Richtungen ist das nur mit Einschränkungen möglich. Durch den Abbildungsprozess bei der Projektion kommt es immer zu Verfälschungen der Konformität, Entfernung, Richtung oder Größe. Es sind über 200 verschiedene Projektionsmethoden bekannt.

Sehr häufig findet die Universal Transverse Mercator-Projektion Anwendung. Dies ist eine konforme (winkeltreue) Abbildung. Geometrisch lässt sie sich als transversale Mercatorprojektion interpretieren, d.h. ein Zylinder wird senkrecht (transversal) über die Erde gestülpt. Der Umfang dieses Zylinders ist kleiner als der Erdumfang längs des Längengrades. Es handelt sich also um einen Schnittzylinder. Dadurch kann im Unterschied zur Gauß-Krüger-Projektion, mit einem Berührungszylinder, die Streifenbreite größer gewählt werden.

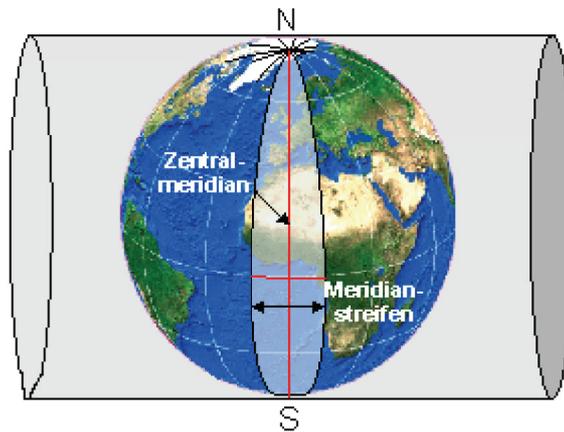


Abbildung 1.1: transversaler Schnittzylinder[olanis]

Längs der beiden Schnittmeridiane ist die Abbildung längentreu. Mit zunehmendem Abstand davon wächst der Fehler an. Um die ganze Erde abbilden zu können, muss dieser Zylinder also systematisch um je 6 Grad weiterbewegt werden. Es entstehen 60 Streifen zu je 6 Grad. Auf dieses Zonensystem wird nun ein rechtwinkliges Koordinatensystem gelegt, dessen Abszisse der Mittelmeridian (Hochwert) und dessen Ordinate durch den Äquator (Rechtswert) gebildet werden. Für Positionen in der nördlichen Hemisphäre ist der Ursprung festgelegt als "false easting" von 500,000 und als "false northing" von 0. Für Positionen in der südlichen Hemisphäre ist der Ursprung festgelegt als "false easting" von 500,000 und als "false northing" von 10,000,000.

Photogrammetrie

Die Photogrammetrie umfasst Methoden zur Gewinnung und Verarbeitung von Informationen beliebiger Objekte. Aus einem oder mehreren (Luft-)Bildern wird indirekt die Form und Lage des Objektes durch Bildmessungen erfasst. Ziel einer photogrammetrischen Messung ist die exakte dreidimensionale geometrische Rekonstruktion des Objektes, wobei das Objekt in digitaler (Koordinaten, abgeleitete geometrische Elemente) oder grafischer Form (Bilder, Pläne, Karten) modelliert wird. [Luhmann]

Orthophoto

Bei einer Luftbildaufnahme entstehen Verzerrungen einer fotografischen Zentralprojektion sowie Verzerrungen durch Höhenunterschiede des Geländes. Ein Orthophoto (griech. orthós richtig, gerade, aufrecht) ist ein senkrecht zur Erdoberfläche aufgenommenes, perspektivisch entzerrtes und damit maßstabsgenaues Luftbild, das durch Transformation auf ein orthogonales Koordinatensystem projiziert ist. Zur Entzerrung wird ein digitales Geländemodell genutzt. [wiki08]

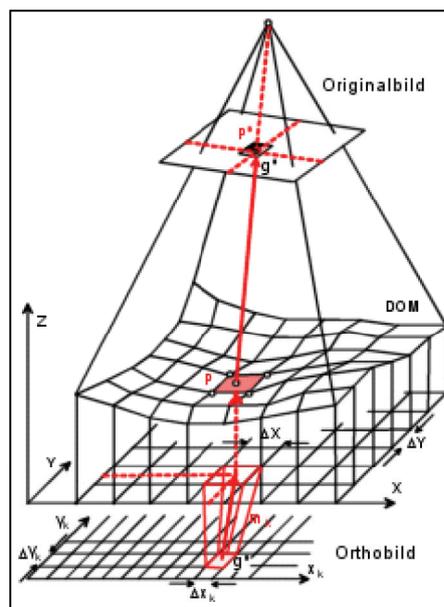


Abbildung 1.2: schematische Darstellung einer Luftbildentzerrung[hro]

Digitales Geländemodell (DGM)

(engl.: Digital Terrain Model - DTM)

Ein DGM bezeichnet die digitale Darstellung der Geländeoberfläche durch räumliche Koordinatentripel einer Menge von Flächenpunkten. Oft wird das Format TIN genutzt, um die Höheninformation darzustellen. Die Abkürzung TIN steht für **T**riangulated **I**rrregular **N**etwork. Dabei werden die Höhenpunkte so vermischt, dass ein Netz von unregelmäßigen Dreiecken erzeugt wird. Ein DGM ist ein Datenbestand zur höhenmäßigen Beschreibung des Geländes. Aus dem DGM lassen sich z.B. Höhenlinienkarten ableiten, Volumina und Neigungen berechnen. Das Digitale Geländemodell (DGM) umfasst inhaltlich sowohl das Digitale Höhenmodell (DHM) als auch das Digitale Situationsmodell (DSM).
[hro]

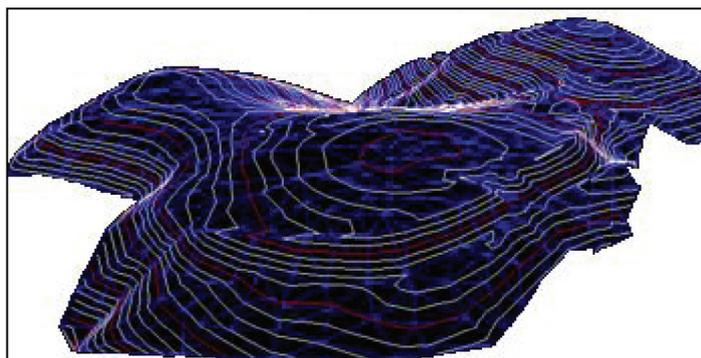


Abbildung 1.3: DTM (mit Höhenlinien)

Dateiformat – TIFF

Zur Speicherung von hochauflösenden Bilddaten hat sich das Tagged Image File Format (TIFF oder auch TIF) etabliert. Es wurde 1986 von der Firma Aldus (später Adobe) entwickelt. TIFF ermöglicht die verlustfreie Speicherung von großen Datenmengen. Beliebig oft können Tiff-Dateien geladen und gespeichert werden, ohne dass ein Qualitätsverlust zu verzeichnen ist. Zudem ist das Format nicht an eine einzelne, bestimmte Software gebunden, sondern eine Vielzahl von Programmen kann zur Bearbeitung genutzt werden.

Der Grundaufbau sieht folgendermaßen aus:

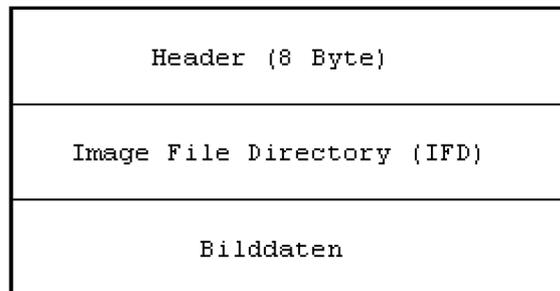


Abbildung 1.4: Grundstruktur Tiff

Der Header belegt immer die ersten acht Bytes der Datei. Er enthält Informationen für die Speicherart der Datei-Bytes (II für Intel little-endian, MM für Motorola big-endian) sowie die Versionsnummer und die Adresse des ersten Image File Directories (IFD). Ein Image File Directory (IFD) besteht aus den drei folgenden Bereichen:

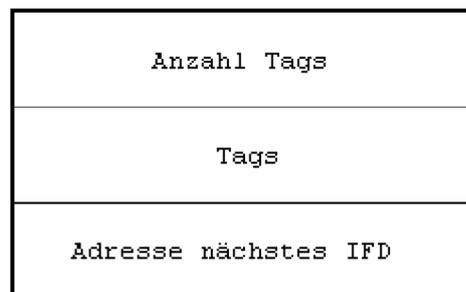


Abbildung 1.5: IFD Aufbau

Jedes Tag enthält Informationen über das zugehörige Bild. Dies können z.B. Bildbreite, Bildhöhe, Komprimierungsart sowie X- und Y-Offset sein. Die Anzahl der Tags ist variabel. Es gibt über 90 verschiedene Tags, die auftreten können. Daher ist die Länge des IFD nicht fest. Außerdem kann ein IFD überall (nach dem Header) in einer TIFF-Datei vorkommen. Enthält die Datei mehrere Bilder, so sind mehrere Anordnungen von IFD und Bilddaten möglich:

Header	Header	Header
IFD_1	IFD_1	Bilddaten_1
IFD_2	Bilddaten_1	Bilddaten_2
...	IFD_2	...
IFD_n	Bilddaten_2	Bilddaten_n
Bilddaten_1	...	IFD_1
Bilddaten_2	...	IFD_2
...	IFD_n	...
Bilddaten_n	Bilddaten_n	IFD_n

Abbildung 1.6: mögliche Anordnungen

Dateiformat – GeoTIFF

Das Format GeoTIFF ist eine Erweiterung des TIFF-Formats für georeferenzierte Daten. Um die zusätzlichen Informationen in das TIFF einzubetten werden sechs weitere Tags genutzt. Die so genannten Geokeys sind dabei in einem Geokey Directory Tag (Tag# 34735) gespeichert. Damit können für entzerrte Luftbilder die Koordinaten zur Georeferenzierung und die verwendete Kartenprojektion gespeichert werden.

1.2 Arbeitsumgebung und Technische Grundlagen im Unternehmen

Luftbilderzeugung

Die IFMS GmbH & Co. KG ist ein Ingenieur-Dienstleistungsunternehmen mit Sitz im Kompetenzzentrum für Flugzeugbau und Entwicklung in Pasewalk. Der Haupttätigkeitsbereich liegt neben Consulting-Dienstleistungen in der digitalen Photogrammetrie. Eines der Hauptabsatzprodukte des Unternehmens ist das digitale Orthophoto (DOP). Zur Datenakquisition werden Bildflüge mit einer digitalen Kamera durchgeführt. Die gewonnenen Luftbilder werden zu Geländemodellen und schließlich Orthophotos weiterverarbeitet.

Bildflüge

Eine Messkamera ist in einem Flugzeug installiert und macht von einem bestimmten Zielgebiet Aufnahmen. Heutzutage werden die Aufnahmen rechnergesteuert ausgelöst. Dazu wird im Vorfeld ein genauer Flugplan erstellt. In ihm sind die Koordinaten aller Aufnahmeorte gespeichert. In parallelen Flugstreifen wird so die gewünschte Fläche erfasst.

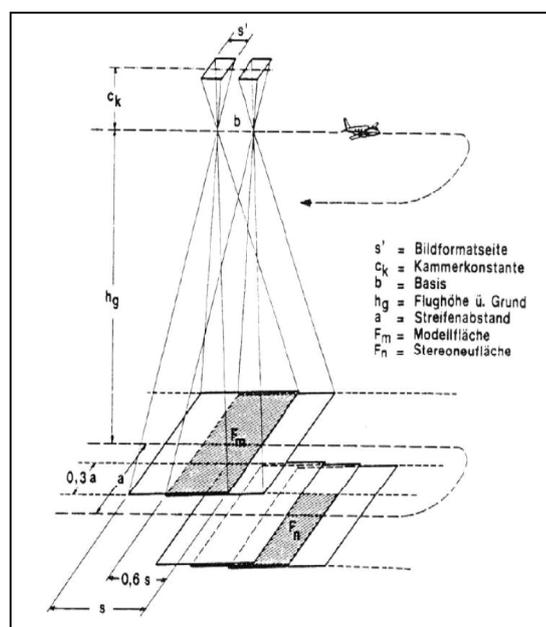


Abbildung 1.7: Bildflug [Kraus1]

Beim Flug muss eine wichtige theoretische Grundvoraussetzung eingehalten werden. Um die Fotografien photogrammetrisch auswerten zu können, muss jedes Objekt auf mindestens zwei verschiedenen Bildern abgebildet sein. Um dies zu gewährleisten werden die Aufnahmen mit einer Überlappung von wenigstens 50% erzeugt. Ist diese Voraussetzung erfüllt, kann der räumliche Vorwärtsschnitt angewandt werden, um die 3D-Koordinaten der Objektpunkte zu berechnen. Diese Koordinaten sind die Basisdaten für das DTM (Abschnitt 3.1), welches zur Entzerrung des Luftbildes genutzt wird, um daraus ein Orthophoto zu generieren.

Digitale Sensoreinheit:

Die IFMS nutzt zur Bilddatengewinnung einen digitalen Sensor (UltraCam D) der Firma Vexcel. Die UltraCam ist eine Matrixkamera und verwendet neun Flächenarrays auf vier Fokalebene. Die Arrays werden so ausgelöst, dass lediglich ein Projektionszentrum einer virtuellen Fokalebene entsteht. [LEBERL, 2003] Zusätzlich zu den neun Sensoren, die das panchromatische Bild erzeugen, werden vier CCD's eingesetzt, um den multispektralen Bereich abzubilden. Wichtige Informationen einer digitalen Kamera sind die Lage des Bildhauptpunktes in der Bildebene und die Kamerkonstante; sie werden als *innere Orientierung* bezeichnet.



Abbildung 1.8: UltraCam - D

GPS / INS:

Während des Bildfluges wird zusätzlich die *äußere Orientierung* bestimmt. Dazu gehören die drei Koordinaten XYZ (Lage) der jeweiligen Aufnahmeorte, die mittels Global Positioning System im Objektkoordinatensystem aufgenommen werden. Weiterhin sind es drei Drehwinkel, die durch ein Inertialsystem aufgezeichnet werden. Sie beschreiben die räumliche Stellung der Aufnahmerichtung im Augenblick der Belichtung des Luftbildes [Pfeiffer]

OPC - Bildbearbeitung:

Aus den Bildern der Digitalkamera können nicht direkt Orthophotos erzeugt werden. Es sind mehrere Schritte notwendig um, die Informationen verwertbar zu machen. Dazu werden die digitalen Daten, von der im Flugzeug montierten Kamera, auf eine mobile Speichereinheit überspielt. Im Büro wird dann der Download von der MSU (MobileStorageUnit) auf das dortige Speichersystem vollzogen. Diese Rohdaten werden als lv10 – Daten bezeichnet. Sie sind das Ausgangsmaterial für die OPC- Software. Dieses spezielle Programm wird von dem Kamerahersteller Vexcel mitgeliefert. Damit wird aus den dreizehn einzelnen Bildern, die die dreizehn Sensoren der Kamera liefern, ein Gesamtbild berechnet. Der erste Schritt ist das Berechnen der lv12-Bilder. Das sind zum Einen ein hochaufgelöstes panchromatisches Bild, das aus neun Sensoren besteht, und zum Anderen ein niedrigaufgelöstes Bild, das aus den vier multispektralen Sensoren entsteht. Aus den lv12 Bildern werden im nächsten Prozess lv13-Daten. Dazu wird zuerst eine radiometrische Anpassung durchgeführt, sodass die Objekte im Bild den Objekten in der Realität farblich gleichen. Dann folgt das Pan-sharpening, bei dem die niedrigaufgelösten Farbkanäle und der hochaufgelöste schwarz-weiß – Kanal zu hochaufgelösten Multispektralbildern fusionieren.

GPS Postprocessing:

So wie die Bilddaten der Kamera, müssen auch die Daten des GPS nachbearbeitet werden.

Die Angaben über die Lage werden durch differentielle Berechnung verfeinert. Zur Verbesserung der Positionsbestimmung ist es möglich einen Referenzempfänger zu verwenden, der sich an einer geodätisch genau bekannten Position befindet. Anhand der Abweichung zwischen der realen Position und der berechneten GPS-Positionierung des Referenzempfängers lässt sich ein Korrekturwert zur Lagekorrektur einer zeitgleich durchgeführten Feldpositionsbestimmung festlegen.

Storage:

Während der Abarbeitung von Luftbildprojekten fallen immense Summen an Daten an. Um die großen Mengen an Daten verwalten zu können, reicht kein normales Speichersystem aus. Im Unternehmen wird ein StorageAreaNetwork von Hitachi Data Systems genutzt, das insgesamt eine Kapazität von sechs Terabyte bietet. SAN ist ein Netz, das Server und Speichersystem über dedizierte Leitungen miteinander verbindet. Ein großer Vorteil des SAN ist die Virtualisierung der vorhandenen Plattensysteme. Der verteilte Massenspeicher kann virtuell wie eine einzige Festplatte behandelt werden.

2. Problemstellung/Aufgabenstellung:

Der zeitaufwändigste Abschnitt in der Prozesskette ist die jetzt anschließende Arbeit im ERDAS – LPS, um Orthophotos zu erzeugen. Mit Hilfe der grafischen Bedienoberfläche, können die gewünschten Manipulationen einfach durchgeführt werden, jedoch sind die, sich wiederholenden, zahlreichen Schritte bei großen Projekten sehr zeitaufwändig. Da es immer die gleichen monotonen Abfolgen sind, besteht zudem die Gefahr falsche Eingaben zu tätigen und so Fehler in der Produktionskette zu erzeugen.

Wie jedes Unternehmen ist auch die IFMS daran interessiert, gewinnorientiert zu arbeiten. Die Fehlerquellen und die manuellen Tätigkeiten sollen auf ein Minimum reduziert werden, um effektiver zu arbeiten.

Aus diesem Grund entstand die Aufgabe, und auch das Thema dieser Diplomarbeit: den bestehenden Workflow im Bereich Orthophotoproduktion zu optimieren. Das Ziel ist die Entwicklung eines Werkzeugs, mit dem die zahlreichen händischen Arbeitsschritte minimiert werden, um dadurch lange Arbeitszeiten zu verkürzen.

Als Vorgabe der Firma stand als wichtiges Kriterium: die Prozesse während der Orthophotoproduktion automatisiert ablaufen zu lassen.

3. Allgemeine technische Methoden zur Umsetzung

3.1 Client – Server – Modell

Das Client-Server-Modell beschreibt eine Möglichkeit, Aufgaben und Dienstleistungen innerhalb eines Netzwerkes zu verteilen. Man unterscheidet dabei zwischen Rechnern, die Dienste anbieten (Server) und Rechnern, die diese Dienste in Anspruch nehmen (Clients) - ein Rechner kann auch beide Eigenschaften besitzen. Ein Server hat die Aufgabe, Anfragen, die ihn über das Netzwerk erreichen, entgegen zu nehmen, zu bearbeiten und die Antwort an den aufrufenden Client weiterzuleiten.

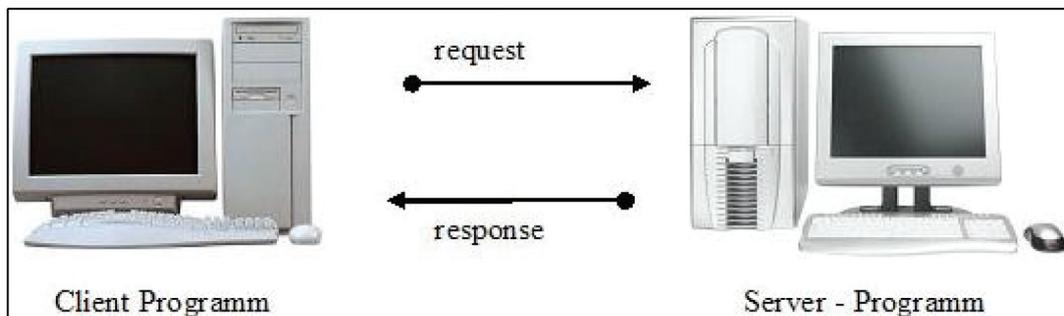


Abbildung 3.1: Client-Server-Prinzip

3.2 CORBA (Common Object Request Broker Architecture)

CORBA ist eine von der OMG 1992 verabschiedete Spezifikation zur Kommunikation zwischen verschiedenen Programmen. CORBA-konforme Implementierungen vereinfachen das Erstellen verteilter Anwendungen in heterogenen Umgebungen. CORBA ist nicht an eine bestimmte Programmiersprache gebunden. Mittels einer Interface Definition Language (IDL) erstellt man eine formale Spezifikation der Klassen und Objekte sowie sämtlicher Parameter und Datentypen. Diese Schnittstellenbeschreibung wird dann in ein Objektmodell der verwendeten Programmiersprache umgesetzt. Die breiteste Unterstützung existiert für Java und C++. Es existieren jedoch auch Implementierungen für viele weitere Sprachen. [omg]

3.3 Programmiersprache C++

C++ wurde in den 1980er Jahren von Bjarne Stroustrup basierend auf der Programmiersprache C entwickelt und ist schon lange de facto standardisiert. Das heißt, das C++-Programm kann normalerweise mit jedem beliebigen C++-Compiler übersetzt werden - unabhängig, von welchem Hersteller der Compiler stammt, und unter welchem Betriebssystem er läuft. Seit 1997/98 gibt es einen offiziellen C++ Standard. Der wichtige Teil des Standards ist die C++ Standard Library. In dieser Library sind nützliche Funktionen, Klassen und Objekte zusammengefasst und standardisiert worden, um diese Hilfsmittel allen C++-Programmierern in einheitlicher Form zur Verfügung zu stellen. Es handelt sich also bei diesem Standard um eine Art Werkzeugkasten, auf den jeder C++-Programmierer in seinen Programmen zugreifen kann. In diesem Werkzeugkasten liegen Werkzeuge, die alle mit Hilfe von C++ entwickelt wurden. [highscore08]

3.4 Stapelverarbeitung

Stapelverarbeitung, oder auch Batchverarbeitung, bezeichnet die sequentielle, nicht-interaktive Bearbeitung von Aufgaben. Stapelverarbeitung im Sinne von Listenabarbeitung wird häufig für betriebssystemnahe Dienste und Aufgaben verwendet. Für solche Listen ist vor allem durch MS-DOS der Begriff Batchdatei oder Batchfile bekannt geworden. Batchdateien enthalten eine Reihe von Befehlen, die die Betriebssystemfunktionen nutzen; diese werden nach dem Starten der Datei Schritt für Schritt abgearbeitet. Mit Batchdateien, die auch als Batchprogramme oder Skripts bezeichnet werden, können sich ständig wiederholende Tasks vereinfacht werden. Eine Batchdatei ist eine nicht formatierte Textdatei, die einen oder mehrere Befehle enthält und die Dateinamenerweiterung .bat oder .cmd hat. Nachdem der Dateiname an der Eingabeaufforderung eingegeben ist, führt Cmd.exe die Befehle in der Reihenfolge aus, in der sie in der Datei stehen. [wiki08]

3.5 ERDAS Makrosprache EML

Bei dem Begriff Makrosprache handelt es sich um eine einfache Form einer Programmiersprache, wie sie in vielen Anwendungsprogrammen zu finden ist. Über einfache Aufrufe werden bereits umfangreiche Funktionalitäten angeboten und über diese bereitgestellten Funktionen können noch mächtigere Funktionen (Makros) definiert werden.

EML wurde von den ERDAS IMAGINE Entwicklern zur Verfügung gestellt, um die Standard-Anwenderschnittstelle zu individualisieren und um häufig wiederkehrende Aktionen in Form von Makros zu automatisieren. Es fungiert als eine Skriptsprache und zugleich als eine Schnittstellensprache. EML bietet mit dem „IMAGINE Developers‘ Toolkit“ eine Sammlung von Bibliotheken und Dokumentationen, um die ERDAS Software zu modifizieren und nutzerspezifisch zu erweitern. Das Toolkit beinhaltet bereits ein Vielzahl von C/C++ Programmierschnittstellen, die für den erfahrenen Programmierer gedacht sind.

Je nach Umfang der verwendeten Makrosprache lassen sich sogar kleinere Anwendungsprogramme entwickeln. Im Allgemeinen sind Makrosprachen fast immer der Interpreter-Sprache zuzuordnen, das bedeutet, dass zum Start der entwickelten Programme das Anwendungsprogramm selbst benötigt wird. Die wohl bekannteste Makrosprache ist Visual Basic for Application, kurz VBA.

[bull]

3.6 Arcforest Coresystem

Das in der Aufgabenstellung genannte Kriterium „Automatisierung“ hat große Bedeutung, weil bereits die Luftbildprozessierung (Abschnitt OPC) derart vereinfacht wurde. Es existiert ein sehr komplexes Programm, das die dort auftretenden Prozesse steuert. Das Coresystem wurde von IFMS in Zusammenarbeit mit dem Unternehmen RTS entwickelt.

Die Rechenschritte bei der Luftbildprozessierung werden normaler Weise per GUI oder Kommandozeile ausgeführt. Der kommandozeilenbasierte Aufruf der Funktionen wird im Coresystem genutzt. Eine Batch-Datei dient zum Aufruf der speziellen Anwendungen (im Falle lv12 zu lv13 ist das die opc.exe), und weist diesen die erforderlichen Verarbeitungsvariablen zu. Die Variablen können beispielsweise Farbwerte, Dateinamen oder Laufwerke sein.

Die Grundbausteine des Coresystem sind eine SQL-Datenbank, ein Masterservice, mehrere Slaveservices und eine grafische Nutzerschnittstelle namens Projektmonitor. Die Datenbank enthält Tabellen mit allen notwendigen Basis- und Projektdaten:

- verfügbare Rechner
- Leistungsfähigkeit der Rechner
- mögliche Arbeitsprozesse
- Beziehung zwischen den Prozessen
- Bezeichnung der Batch-Dateien für den jeweiligen Arbeitsprozess
- Parameter für Radiometrie, Flughöhe, Pixelgröße u.v.m
- Quell- und Ziellaufwerke
- IDs der Bilder, die bearbeitet werden sollen
- Status der Prozesse

Vereinfacht dargestellt ist das Coresystem ein großer Informationspool, der alle zu erledigenden Aufgaben enthält. Die Aufgaben werden durch einen Verteiler an die angebundenen Recheneinheiten zugewiesen. Als Recheneinheit zählt nicht nur der einzelne Rechner, sondern jede logische CPU.

Mit dem Projektmonitor wird ausgewählt welche Prozesse auf welche Bilder angewendet werden. Hier kann auch das Editieren der Attributwerte der Datenbank erfolgen.

Die Anwendung „Masterservice“ hat die Rolle des Verteilers. Er greift auf die Datenbankinformationen zu und gibt die einzelnen Aufgaben an die „Slave Services“ weiter. Auf den Rechnern, die als Slave fungieren ist die Anwendung „Slave Service“ installiert. Hier werden die Rechenprozesse ausgeführt. Das System basiert auf der Client-Server-Architektur, wobei der Masterservice der Client ist und die Slaves die Server darstellen.

3.7 Datenbank SQL Server

Das Core-System nutzt ein relationales Datenbanksystem. Das hierfür genutzte relative Datenbankmanagementsystem ist der Microsoft SQL-Server. Die großen Mengen an Informationen werden mit Hilfe der Datenbank effizient verwaltet. Zur Kommunikation mit dem Datenbanksystem und zum Editieren wird die, von Edgar F. Codd entwickelte, Datenbanksprache SQL verwendet.

4. Realisierung

In diesem Kapitel wird beschrieben, wie das geforderte Ziel erreicht wurde. Beginnend mit der Betrachtung der Ausgangslage, weiter mit der Analyse, bis hin zur Umsetzung.

4.1 Ist – Situation

ERDAS-LPS

Für die eigentlichen photogrammetrischen Aufgaben findet ein Softwarepaket aus dem Hause Leica-Geosystems Anwendung.

Mit dem „ERDAS Imagine“ bietet Leica-Geosystems eines der weltweit führenden Produkte in punkto Auswerte- und Interpretationssoftware an. ERDAS ist unterteilt in die Module IMAGINE Essentials, IMAGINE Advantage und IMAGINE Professional. Der Anbieter selbst beschreibt das Modul Essentials als Einstiegspaket zur Visualisierung von geographischen Daten: 2-D- und 3-D- Darstellung, Georeferenzierung, einfache Analysefunktionen, Bildverwaltung, Kartengestaltung und -ausgabe. Die nächste Ausbaustufe enthält dann zusätzlich Funktionen zur Bildverarbeitung und Raster-GIS Analyse. Orthorektifizierung von Einzelbildern und deren Mosaikieren runden das „IMAGINE Advantage“ ab. Komplettiert wird das Paket durch Werkzeuge zur flexiblen Klassifizierung, zur Modellbildung und zur Radardatenbearbeitung. Dann trägt es die Bezeichnung „Professional“. [Leica]

Wie bereits erwähnt, erstellt die IFMS Orthophotos. Das dafür notwendige Arbeitsmittel ist das Modul Advantage. Speziell für Arbeitsschritte wie die DTM-Generierung steht ein Produkt namens „Leica Photogrammetry Suite“ (LPS) zur Verfügung. Es verfügt über die Funktionen von IMAGINE Advantage und ist damit der Link zum Programmpaket ERDAS IMAGINE. Das LPS und seine Submodule bieten eine Vielfalt von Funktionen.

Die besonders relevanten Einsatzmöglichkeiten sind:

- das Messen von Passpunkten
- die automatische Generierung von Verknüpfungspunkten (TiePoints)
- das Durchführen der Triangulation
- die Definition des verwendeten Aufnahmesensors (Luft- oder Satellitenbild)
- die ATE (Automatic Terrain Extraction)
 - erlaubt das automatische Generieren eines digitalen Geländemodells aus Bildern eines photogrammetrischen Blockes
- Berechnen und Mosaikieren von Orthophotos

Im Unternehmen IFMS ist die LPS das einzige Werkzeug, das benutzt wird, um zum Endprodukt DOP zu gelangen. An dieser Stelle soll jetzt detailliert auf den Umgang mit LPS eingegangen werden, um einen Eindruck vom bestehenden Workflow zu bekommen.

Die LPS wird über eine grafische Benutzeroberfläche namens „Project Manager“ gesteuert. Das ist die zentrale Stelle, von der aus alle Einstellungen und Berechnungen vorgenommen werden. Nach dem Öffnen des Project Managers wird zuerst ein so genanntes block-file erstellt. Das Dateiformat für diesen Bildblock lautet *.blk. Zu Beginn wird gleich festgelegt, was für ein Sensor benutzt wurde und in welcher Karten-Projektion die Enddaten erzeugt werden sollen. Es folgt der Import der zu bearbeitenden Luftbilder in das block-file.

4. Realisierung

Die nächsten Einstellungen betreffen die innere Orientierung. Die Kameradaten: Brennweite, Lage des Bildhauptpunktes und Pixelgröße werden definiert. Darauf folgt die Zuordnung der äußeren Orientierung zu den Luftbildern. Die Informationen der äußeren Orientierung stehen in einer txt-Datei, diese beinhaltet die Bildnummern und die dazugehörigen Werte für X, Y, Z und Omega, Phi, Kappa. Je nach Projektgröße enthält die Datei mehrere tausend Datensätze. Es ist zwingend notwendig, dass die Werte den richtigen Bildern zugeordnet werden. Nachdem die Richtigkeit kontrolliert wurde, werden die Werte fixiert, sodass sie während weiterer Rechenprozesse nicht mehr verändert werden können. Das Setup des Bildblockes ist damit abgeschlossen.

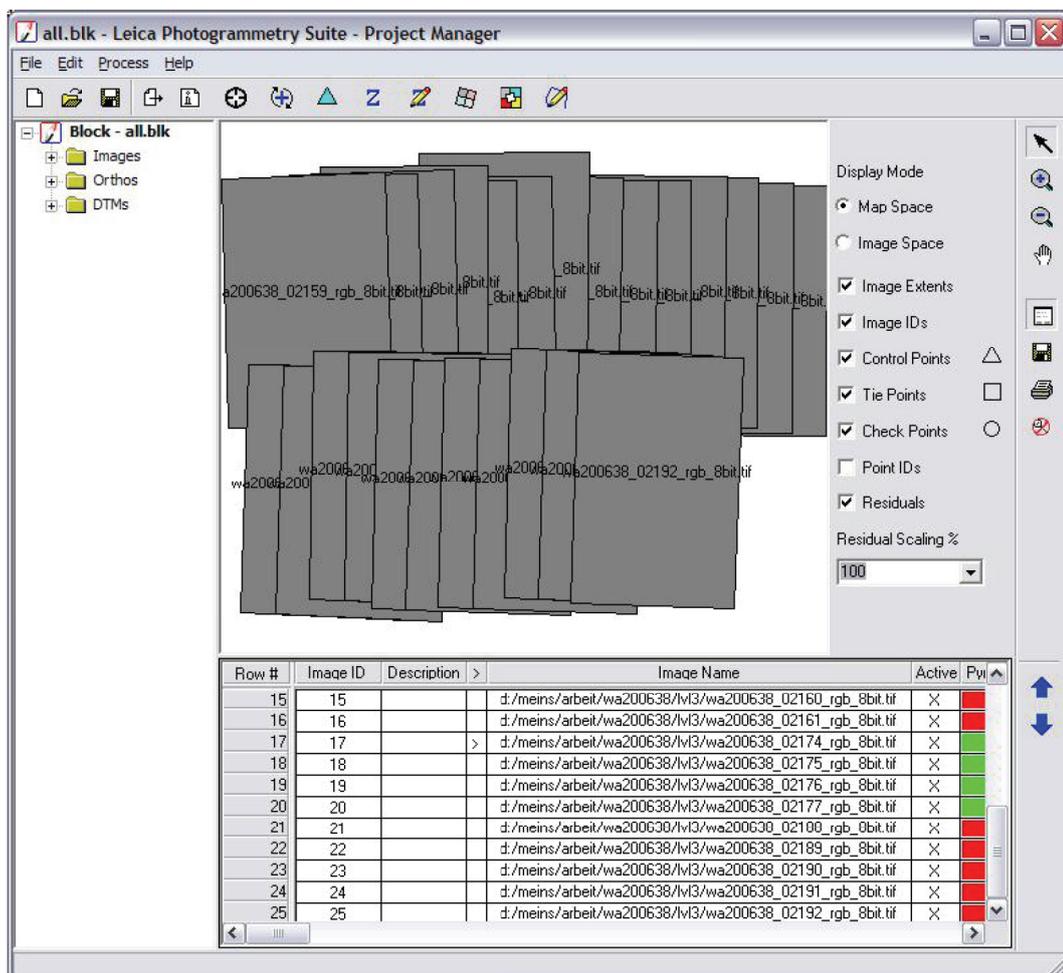


Abbildung 4.1: LPS Bildblock

Als weiterer Schritt steht die Generierung eines Höhenmodells an. Dazu wird das Untermenü „DTM Extraction“ geöffnet. Hier werden Einstellungen über das Dateiformat des DTM (z.B. TIN, ASCII, shape), die Punktdichte, den Dateinamen und den Speicherort vorgenommen. Anschließend kann die automatische Berechnung durchgeführt werden.

Sobald der Rechenprozess abgeschlossen ist, folgt der letzte Schritt im LPS – die Generierung des Orthophotos. Auch hier können und müssen diverse Konfigurationen, im dafür vorgesehenen Menü, vorgenommen werden. Dateiname, -format und Speicherort müssen bestimmt werden. Das zuvor erzeugte Höhenmodell wird benötigt, um die Luftbilder zu entzerren. Eine weitere bedeutende Angabe ist die Pixelgröße für das zu erzeugende, entzerrte Bild. Das Orthophoto ist das finale Produkt der Prozesskette. Bis hier sind mehr als 90 Mouseklicks nötig, um die Arbeit im LPS erfolgreich abzuschließen.

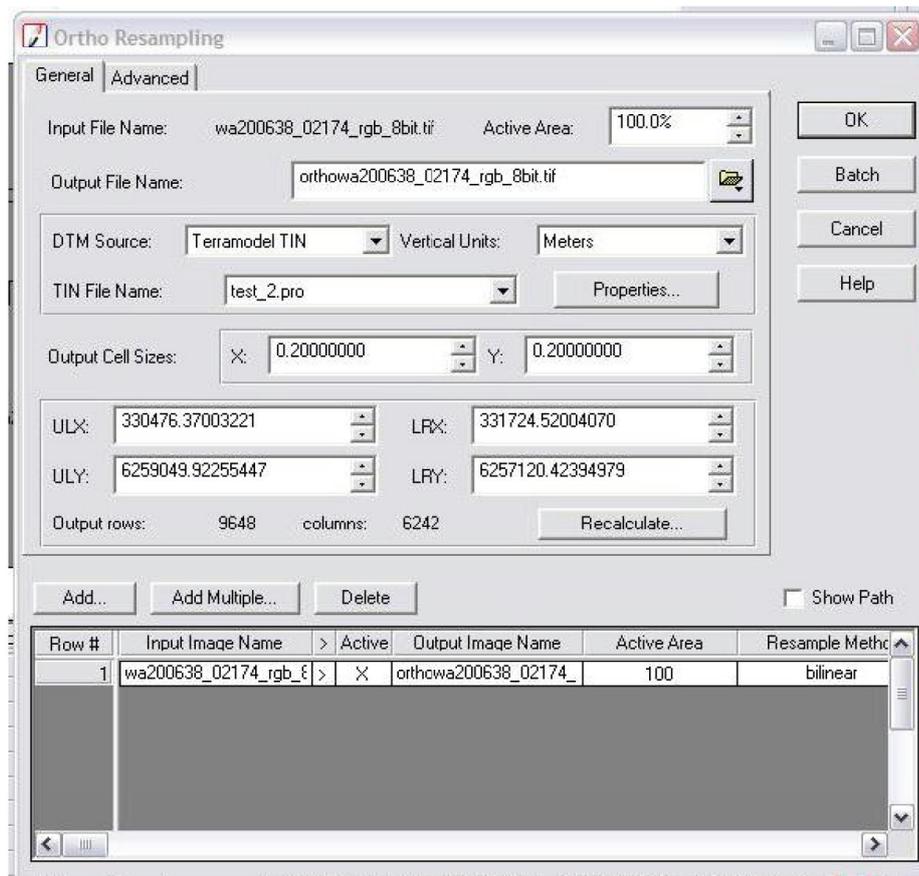


Abbildung 4.2: LPS Ortho-Resampling

4.2 Vorüberlegungen

Im Kapitel 2 wurde beschrieben, dass die Problempunkte Zeitaufwand und Fehlerrate durch einen optimierten Workflow minimiert werden sollen. Der Bearbeiter der photogrammetrischen Tätigkeiten soll ein Werkzeug zur Verfügung gestellt bekommen, welches die Vielzahl von manuellen Arbeitsschritten ausführt und dabei gleichzeitig die Möglichkeiten des Einbringens von Fehlern reduziert. Die Generierung von blk-Dateien, die Berechnung von Höhenmodellen und das Erzeugen von Orthophotos soll nicht, wie bisher, durch den menschlichen Anwender und die graphische Benutzeroberfläche erfolgen. Die Prozesse sollen so weit wie möglich automatisch ablaufen.

Die Lösung der gestellten Aufgabe soll so realisiert werden, dass die vorhandenen Ressourcen des bereits betriebsbereiten CoreSystems genutzt werden können. Aufgrund vorhandener Lizenzen soll nicht vom Programm ERDAS LPS abgewichen werden. Alternativen wie beispielsweise Inpho's *Orthomaster* wurden aus Kostengründen nicht weiter betrachtet. Die vorgegebenen Einschränkungen zeigen eindeutig auf, dass die geplanten Neuerungen nur IFMS-intern genutzt und nicht auf dem freien Markt angeboten werden. Die Rahmenbedingungen für die Orthophotoproduktion sind somit markiert.

4.3 Lösungsansatz

In der ERDAS IMAGINE Dokumentation ist dargestellt, dass über die Kommandozeile die Möglichkeit besteht, sämtliche Erdas-Befehle (eml) einzugeben und auszuführen. Auf diesem Prinzip basiert auch die Funktionsweise des Coresystems. Das ist der Ansatzpunkt für die eventuelle Lösung. So können die vorhandenen eml Funktionen für die jeweiligen Prozesse, ohne die grafische Schnittstelle, genutzt werden. Die drei notwendigen Prozesse (blk, dtm, ortho) sollen als separate Applikationen gestartet werden können. Die Schritte DTM-Generierung und OrthoResampling können nicht völlig eigenständig arbeiten. Sie benötigen als Voraussetzung eine funktionsfähige blk-Datei und die darin enthaltenen Informationen. Demzufolge muss zuerst die Grundlage erstellt werden.

Das Starten und Ausführen der Rechenprozesse soll durch das firmeneigene Coresystem gesteuert werden. Das bedeutet, dass die entwickelten Prozesse anschließend in das Coresystem integriert werden müssen. Dazu gehören neue Einträge in einigen Datenbanktabellen und zugleich das Erstellen der aufrufenden Batch-Dateien.

Wie man den vorrausgegangen Ausführungen entnehmen kann, ist der Ausgangspunkt für die Orthophotoproduktion das Vorhandensein der blk – Datei. Die manuelle Erstellung der blk – Datei wurde in Kapitel 4.1 beschrieben. Bis dato wurden die Bildblöcke immer so definiert, dass circa fünfzig Bilder zu einem Teilgebiet zusammengefasst worden sind. Mittels der DTM-Funktion erstellt der Bearbeiter davon ein Höhenmodell. Nach der Fertigstellung des Höhenmodells können sequentiell die zugehörigen fünfzig Orthophotos generiert werden.

Das Prinzip des Coresystems ist es, viele, möglichst kleine Prozesse verteilt abzuarbeiten. Dieser Grundgedanke bleibt für die neuen Rechenprozesse erhalten. Im automatisierten Workflow werden nur noch zwei Bilder pro Bildblock importiert. Die Mindestanforderung für eine stereoskopische Auswertung ist damit erfüllt.

4.3.1 Untersuchung zum Aufbau eines Bildblockes

Um die Bildblöcke später maschinell zu erzeugen wird zunächst der Aufbau untersucht, um daraus Schlussfolgerungen ziehen zu können. Bei den Untersuchungen zum Aufbau dieser blk - Dateien diente ein Tool aus der ERDAS Palette, der *HFA-Viewer*. Mit ihm ist es möglich die hierarchische Architektur von verschiedenen ERDAS – Dateiformaten zu betrachten. So auch die Anordnung vom .blk. Der Viewer zeigt die Gliederung als eine Baumstruktur. Der Baum ist aus Knotenpunkten aufgebaut, die eine Vielzahl von Datentypen enthalten. Der Inhalt dieser Knoten kann sehr detailliert betrachtet werden. Zur Analyse der Dateien wurden mehrere Bildblöcke aus einem laufenden Bildflugprojekt der IFMs untersucht. Während der Prüfung ist deutlich geworden, dass die Struktur der verschiedenen blk-Dateien sehr ähnlich ist.

4. Realisierung

Innerhalb eines Bildprojektes müssen diverse Parameter eingegeben werden. Dabei gibt es Werte, die von Block zu Block unterschiedlich sind. Genauso gibt es Werte, die für alle Bildblöcke konstant bleiben. Beispielsweise sind die Eingaben für das Koordinatensystem, die Digitalkamera und auch die Parameter der Zwischen- und Endprodukte, DTM-Punktabstand und GSD identisch. Die Einträge für diese Werte sind in der hierarchischen blk – Struktur absolut gleich.

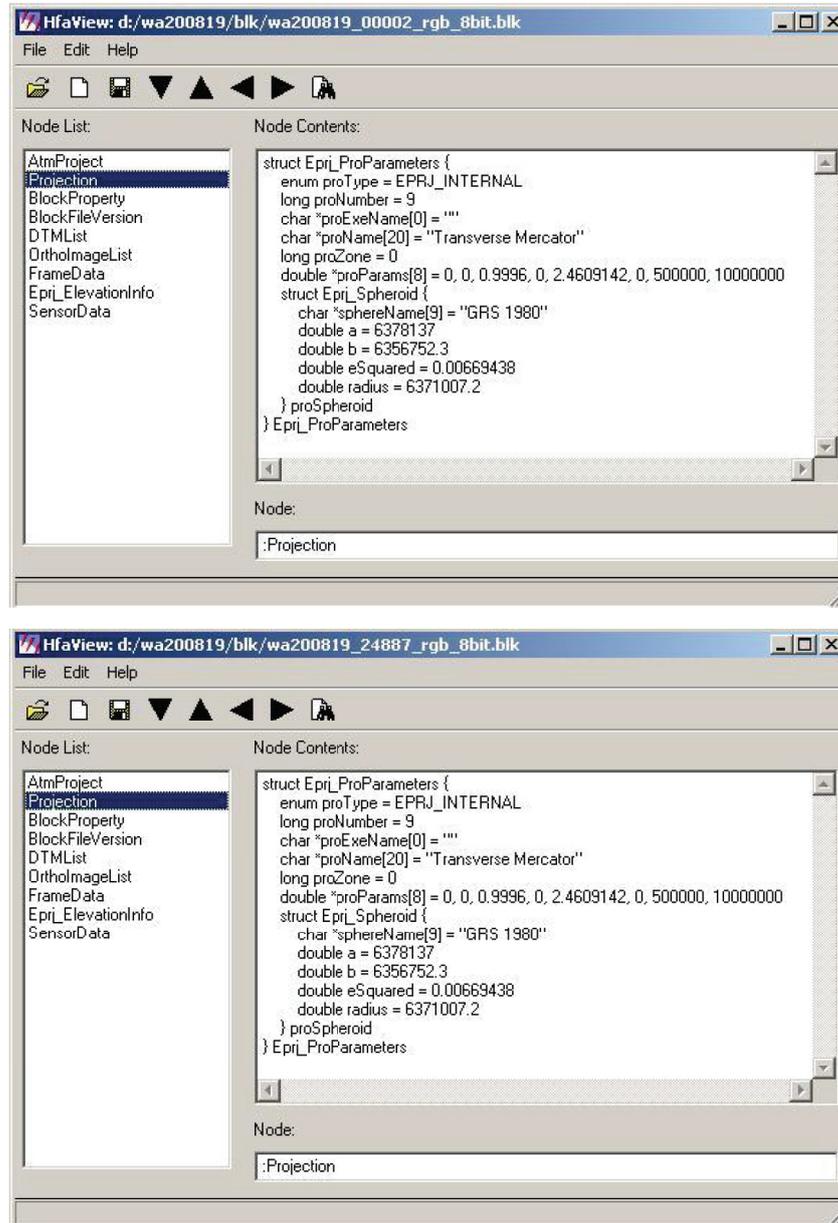


Abbildung 4.3: identische Einträge in der hfa-Struktur zweier Stereobildblöcke

4. Realisierung

Variierende Werte hingegen sind die Bezeichnungen der verwendeten Bilder; konkret die Bildnummern. Die zu jedem Bild gehörenden Werte der äußeren Orientierung sind ebenfalls veränderlich. Auszüge aus der hfa – Struktur zeigt die Abbildung 4.4. Die Unterschiede zwischen den beiden Blöcken sind rot gekennzeichnet.

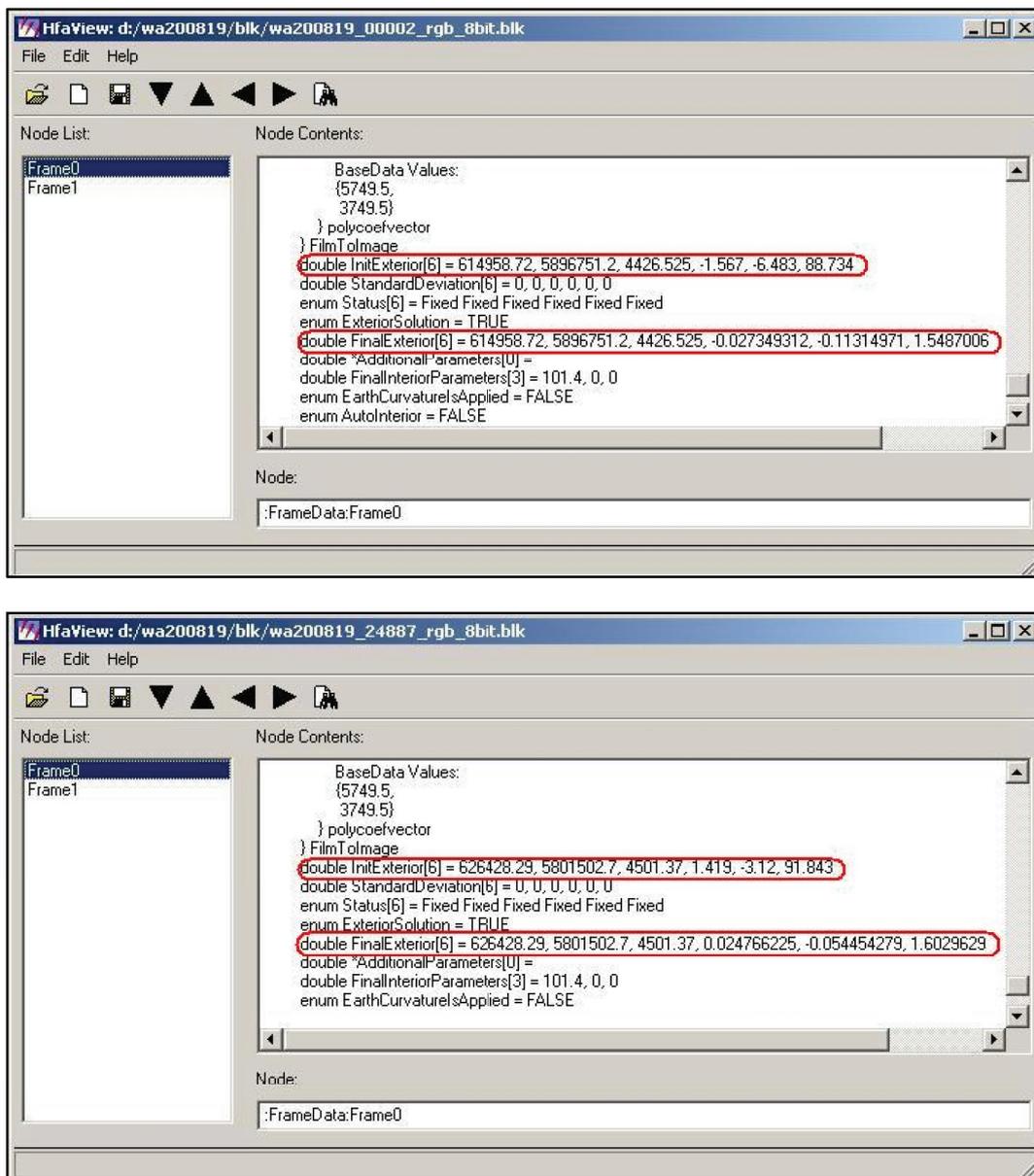


Abbildung 4.4: identische Einträge in der hfa-Struktur zweier Stereobildblöcke

Folgende Werte sind, innerhalb eines Flugprojektes, bei der Nutzung der grafischen Benutzeroberfläche des LPS vom Anwender zu editieren:

	Eingabeparameter	variabel / fix	interne Festlegung
1	o blk - Speicherort	variabel	fix - H:/"projectname"/blk/
2	- Dateiname	variabel	
3	o Sensorik (innere Orientierung)	fix	
4	o Objektkoordinatensystem	fix	
5	o ø Flughöhe	fix	
6	o verwendete Bilder (IDs)	variabel	
7	- Quellort	variabel	fix - H:/"projectname"/lv13/
8	o Eo-daten (äußere Orientierung)	variabel	
	o DTM-Angaben		
9	- DTM-Typ/Format	fix	immer TIN (****.pro)
10	- DTM-Punktabstand	fix	
11	- DTM-Speicherort	variabel	fix - H:/"projectname"/dtm/
12	- DTM-Name	variabel	
	o Orthophoto-Angaben		
13	- Dateiformat	fix	immer TIFF
14	- Pixelgröße (GSD)	fix	
15	- Speicherort	variabel	fix - H:/"projectname"/ortho/
16	- Dateiname	variabel	ortho_"projectname"_ID_rgb_8bit.tif
17	- verwendetes DTM	variabel	

Tabelle 4.1: Eingabeparameter

Die Tabelle zeigt, dass zehn Parameter variabel und sieben Parameter fix sind. Durch firmeninterne Standards werden einige der veränderlichen Werte ebenfalls festgesetzt. So sind dann etwa zwei Drittel der Eingaben gleichbleibend und nur ein Drittel variiert von Block zu Block. Die besagten Standards betreffen eine einheitliche Ordnerstrukturen und einheitliche Dateibezeichnungen in allen Bildprojekten. So sind in jedem Projektverzeichnis dieselben Unterverzeichnisse zu finden. Die Dateinamen weisen immer folgende Imagemask auf: „projectname“_ID_rgb_8bit.tif. Die Bild-ID ist immer fünfstellig. In der detaillierten, hierarchischen Ansicht der Datei ist ebenfalls erkennbar, dass zusätzlich zu den Eingaben, die im GUI erfolgen, noch weitere Werte gespeichert werden. Demzufolge werden während der Arbeit mit der Benutzeroberfläche, im Hintergrund, parallel Berechnungen ausgeführt. Die Ermittlung und die Ergebnisse werden dem Anwender nicht direkt zur Verfügung gestellt. Im weiteren Verlauf der Orthophotoberechnung werden diese Werte benötigt. Es handelt sich um die Eckkoordinaten jedes im .blk befindlichen Bildes.

Die Untersuchung zeigt, dass sich sämtliche Blöcke in den meisten Objektknoten gleichen. Die daraus resultierende Idee ist, einen Bildblock als Vorlage zu nutzen. Die Vorlage wird kopiert und die nötigen Veränderungen werden eingetragen. Anschließend wird das DTM und das Orthophoto erzeugt.

4.4 Konzeption

4.4.1 Anwendungsfälle

Sowohl im bisherigen als auch im neuen Workflow sollen die Arbeiten nur durch eine Person, den Administrator der Photogrammetrie, erfolgen und überwacht werden. Die Basisfunktionen, die das Werkzeug erfüllen soll sind blk-Erstellung, DTM-Extraction und Ortho-Resampling.

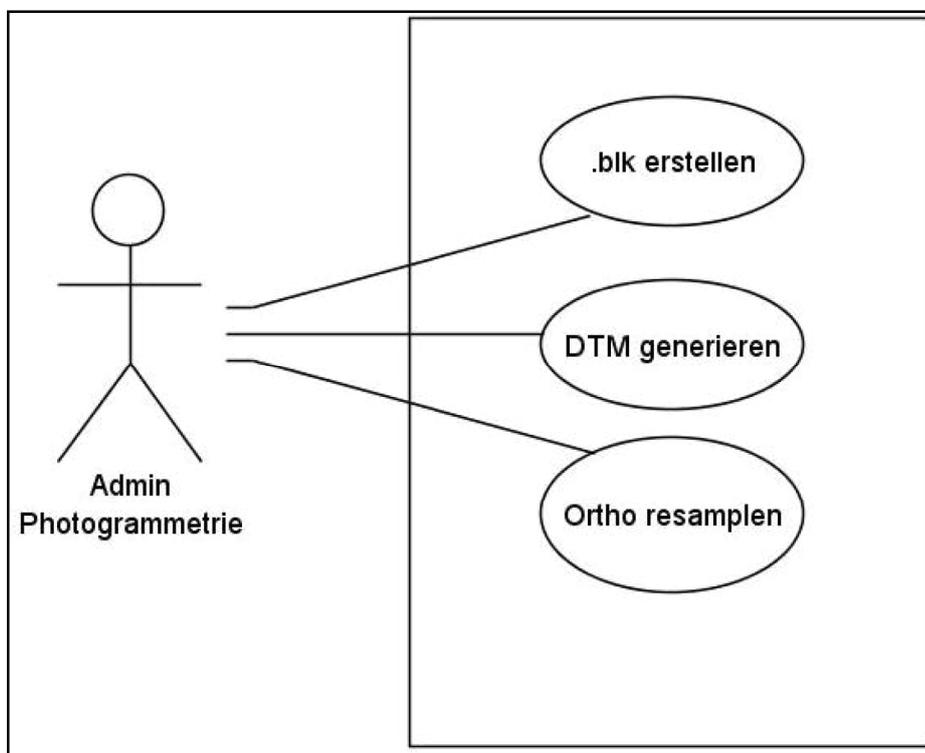


Abbildung 4.5: Anwendungsfälle

4.4.2 Benötigte Module

Die grundsätzliche Arbeitsabfolge kann nicht von dem bisherigen Workflow abweichen. Die drei Module müssen nacheinander abgearbeitet werden.

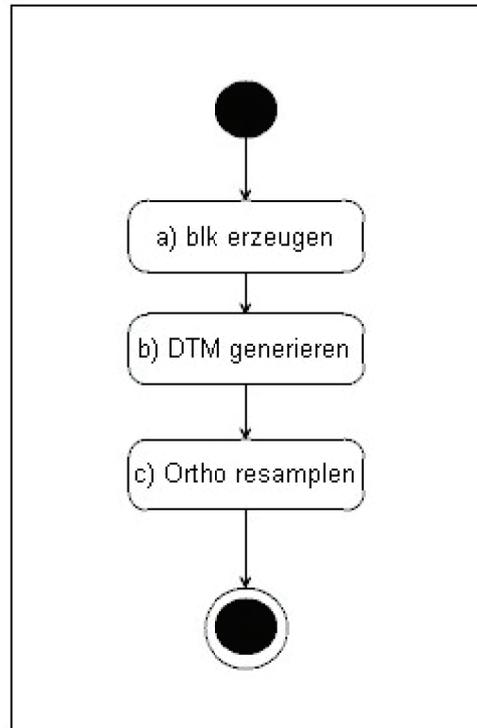


Abbildung 4.6: Ablauf der Module

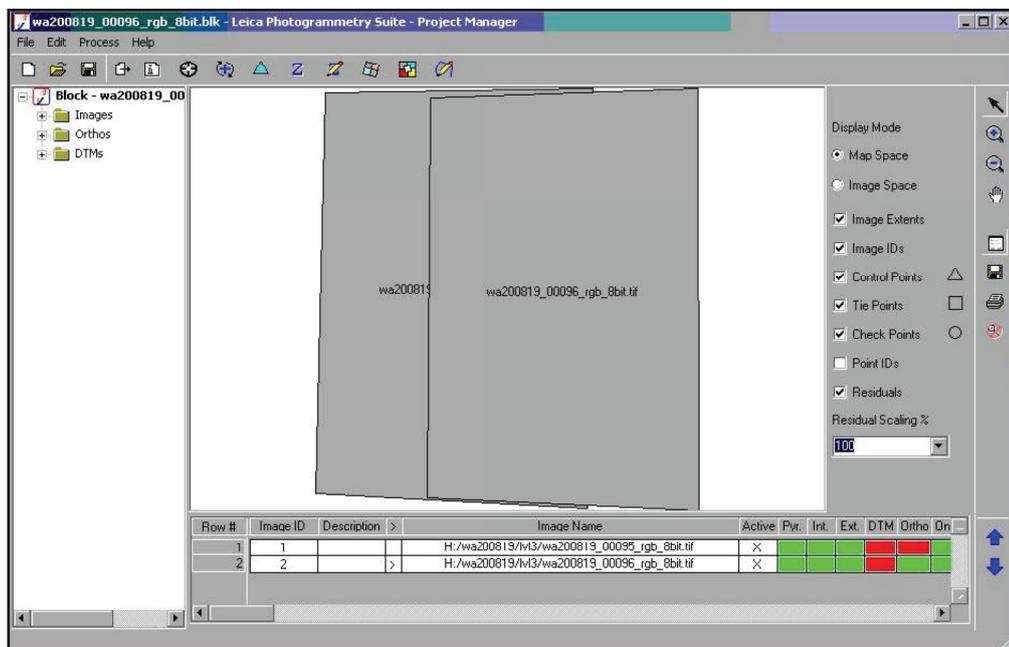
Für Modul b) und c) bestehen feste Prozeduren innerhalb des Leica Photogrammetry Suite. Sie müssen nicht vollständig neu entwickelt werden, sondern mit den korrekten Parametern aufgerufen werden. Für das Modul a) gibt es keinen direkten eml – Befehl wie für die Komponenten b) und c). Das Vorhandensein des blk ist notwendig für die Ausführung der weiteren Schritte. Erst wenn die blk-Datei erzeugt ist, kann der nächste Prozess durchlaufen werden. Ist die DTM-Generierung erfolgreich, kann der letzte Prozess starten. Die neuen Module sind Erweiterungen des bestehenden Coresystems. Der Start der Module erfolgt über Batch-Dateien; durch das Coresystem. Für jede der drei Anwendungen muss eine Stapelverarbeitungsdatei erstellt werden und weiterhin müssen die Module, mit ihren Eigenschaften, in die Datenbank eingepflegt werden.

4.4.3 Bildblöcke erstellen

Der grundlegende Prozess ist Modul A - das Erstellen des Stereoblockes. Dazu wird zunächst manuell, mit Hilfe des LPS, ein Bildblock erstellt. Die Informationen aus diesem Block dienen als Vorlage für die automatisch erzeugten Bildblöcke.

4.4.3.1 Vorlage erstellen

Die Vorgehensweise des Blocksetup ist im Abschnitt 4.1 detailliert beschrieben. In Kapitel 4.2.2 wurde festgelegt, dass nur zwei Bilder in den Block importiert werden um einen Überlappungsbereich zu erzeugen. Der Vorlageblock wird entsprechend der Vorgaben erzeugt. Eine Vielzahl von Parametern wird aus dieser Vorlagedatei in die neuen Dateien übernommen. Deshalb müssen alle Angaben korrekt sein. Das Ausgabeprodukt ist eine Datei mit dem Namen `template.blk`. Im GUI wird der erzeugte Block wie in Abbildung 4.7



dargestellt.

Abbildung 4.7: Ansicht eines Stereoblockes im LPS - GUI

Nach dem Grundsetup werden die Einstellung für die DTM Erzeugung vorgenommen. Das `Terramodel TIN` wird als DTM-Format ausgewählt. Abhängig vom Projekt, bestimmt der Anwender die Punktdichte des Höhenmodells. Dann wird der Speicherort festgelegt und die Berechnung kann gestartet werden. Die Einstellungen für die DTM-Berechnung werden aus dem Vorlageblock ebenfalls für die neuen Blöcke übernommen.

Auch für den dritten Teilabschnitt – OrthoResampling – werden Vorlagewerte benötigt. Dazu muss der Anwender ein Orthophoto berechnen lassen. Er gibt das Zielverzeichnis an, in dem das Orthobild gespeichert wird. Dann browsst er zum zuvor berechneten Höhenmodell, das zur Entzerrung genutzt wird. Der letzte Eingabewert ist die Pixelgröße (GSD) des Ausgabebildes. Dieser wird vom Kunden vorgegeben und wird bereits in der Flugplanung berücksichtigt. Anschließend kann der Rechenprozess gestartet werden. Die Informationen, die für die neuen Blöcke relevant sind, werden nicht nur in der blk-Datei selbst gespeichert. Die manuellen Eingaben erzeugen zwei temporäre Dateien, auf die der Resampleprozess zurückgreift. Eine Datei ist, wie die blk-Datei, hierarchisch strukturiert. Sie wird ebenfalls als Vorlage genutzt. Die Datei wird aus dem Temp-Verzeichnis kopiert und im selben Ordner abgelegt, in dem sich der Vorlageblock befindet. Sie hat die Bezeichnung `resampling_Property_File` bekommen. Die zweite Datei ist im ASCII-Format gehalten. Darin stehen Verweise zum gewählten Höhenmodell. Von ihr wird keine Vorlage benötigt.

4.4.3.2 Modul „BlockFileCreator“

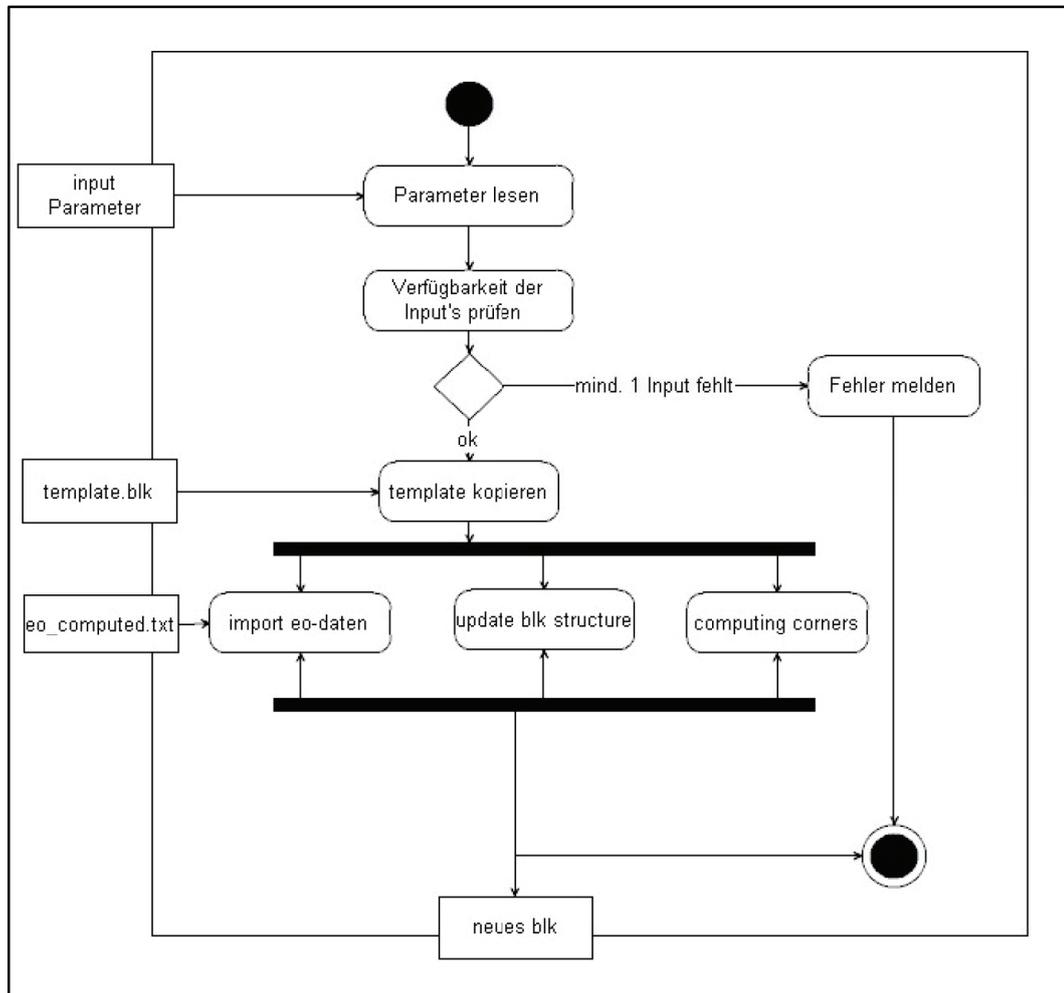


Abbildung 4.8: Aktivitäten zur Bildblockerstellung

Das Modul BlockFileCreator erhält als wichtigsten Input die ID des Bildes, für welches der Stereoblock erstellt werden soll. Es kopiert das `template.blk` und erzeugt einen neuen Block. Die hierarchische Dateistruktur wird dann entsprechend der Bildnummer aktualisiert, sodass die enthaltenen Informationen korrekt sind. Dazu werden aus der Text-Datei `eo_computed.txt` die entsprechenden äußeren Orientierungsparameter importiert. Dann werden die Eckkoordinaten der Bilder berechnet und ebenfalls im Block eingetragen.

4.4.4 Modul „DTMExtractor“

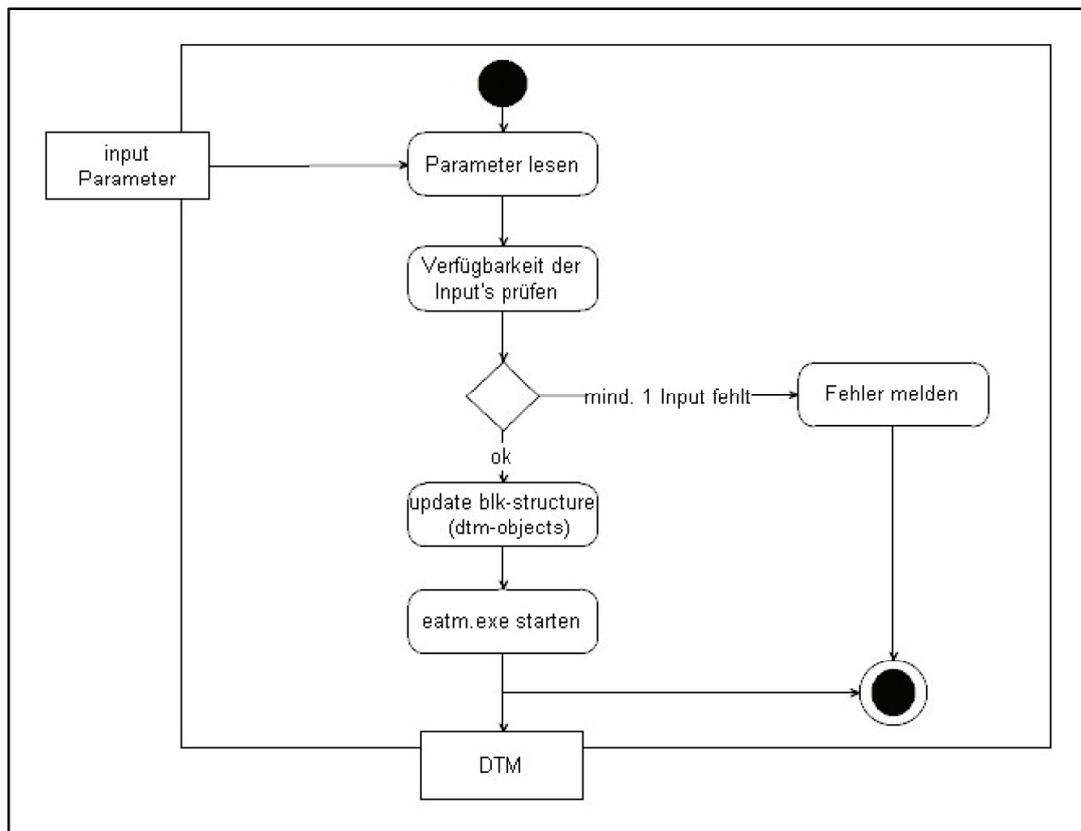


Abbildung 4.9: Aktivitäten zur DTM-Berechnung

Im zweiten Prozess (die *DTM-Extraction*) sollen für diesen Block Höhenpunkte berechnet werden. Das Modul *DTMExtractor* erhält die ID des zu bearbeitenden Block's und verändert zunächst die erforderlichen Objekteinträge in der *blk-Datei*. Dann wird die Berechnung gestartet. Die LPS – Funktion *eatm_batch.exe* errechnet, über drei intern ablaufende Teilschritte, dreidimensionale Koordinaten. Die *eatm_batch.exe* ist Bestandteil der LPS. Voraussetzung für eine ordnungsgemäße Ausführung ist das Existieren eines Stereobildblockes aus dem die erforderlichen Berechnungsvariablen entnommen werden können. Die errechneten Höhenwerte werden in einer *DTM-Datei* gespeichert. LPS unterstützt beispielsweise die Formate TIN, ASCII und TIFF. Nach Absprache mit der IFMS wurde zunächst das Format TIN festgelegt. Mit der Speicherung ist der zweite Prozess abgeschlossen.

4.4.5 Modul „OrthoResampler“

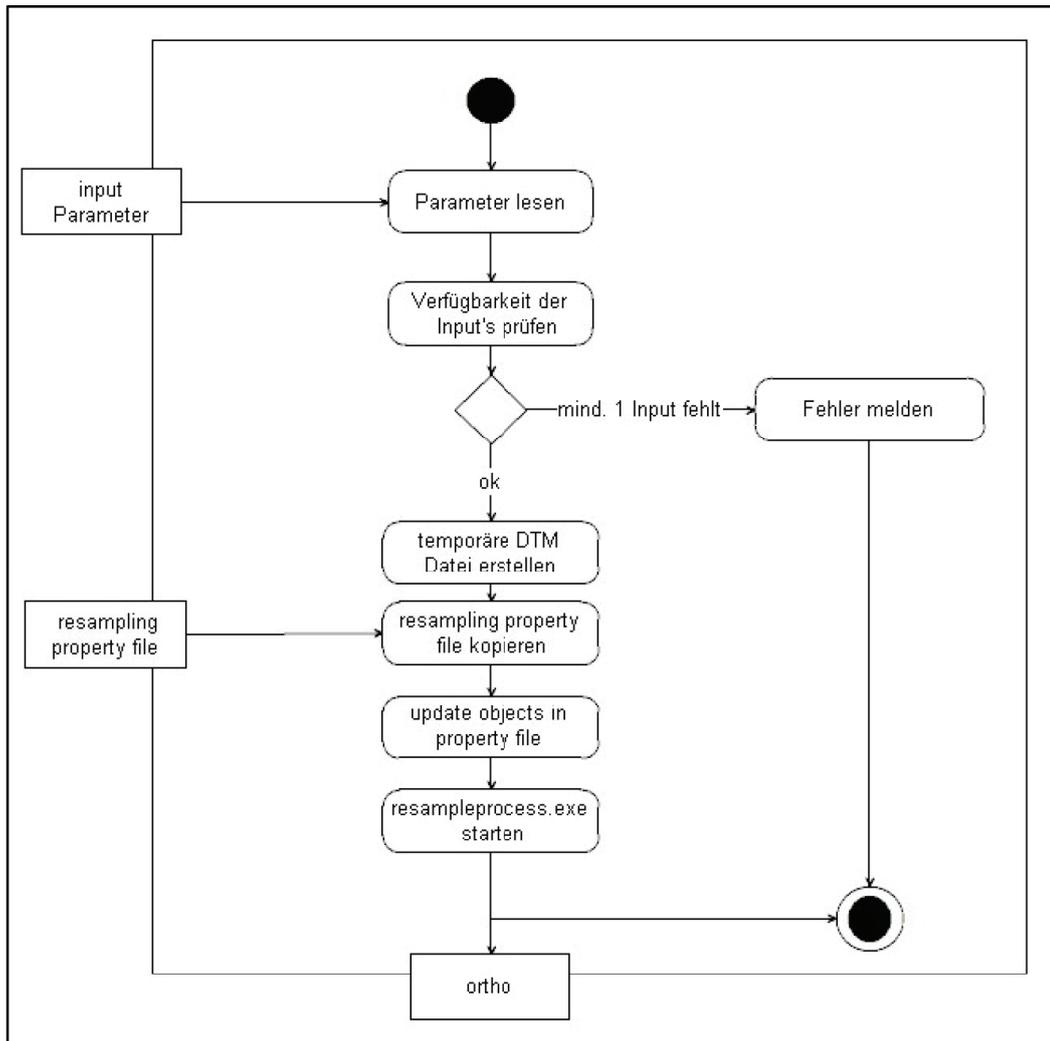


Abbildung 4.10: Aktivitäten zum OrthoResampling

Der dritte und letzte Prozess ist das Resamplen des Luftbildes. Mit dem im zweiten Modul erzeugten Höhenmodell werden die, bei der Aufnahme entstandenen, perspektivischen Verzerrungen entzerrt. Dieses Resamplen ist, wie die DTM-Extraction, eine bereits existierende LPS-Funktion. Das Modul OrthoResampler erzeugt die Elemente, auf welche die `resampleprocess.exe` zugreifen will. Dazu muss eine temporäre Datei erstellt werden, die Informationen zum Höhenmodell enthält. Anschließend wird das `resampling_Property_File` kopiert. Darin werden, entsprechend der Bild-ID, Einträge in der hierarchischen Dateistruktur editiert. Zuletzt wird die Funktion `resampleprocess.exe`, in der

ERDAS-Umgebung, mit den notwendigen Parametern aufgerufen. Das Resultat ist ein georeferenziertes, entzerrtes Luftbild; ein Orthophoto.

4.5 Umsetzung

Die programmiertechnische Umsetzung, der konzipierten Anwendungen, erfolgt in der Programmiersprache C++. Als Entwicklungsumgebung dient dabei Microsoft Visual C++.NET 2003. Das IMAGINE Developers' Toolkit wird zum Bearbeiten der konkreten ERDAS - Anwendungen eingesetzt. Um die Eingriffe in die Strukturen, der professionellen Software, qualitativ hochwertig zu gestalten, ist ein ausgebildeter Programmierer mit der Implementierung beauftragt worden. In enger Zusammenarbeit mit Herrn Jakob Lyko, der seinen Master in „Electronic and Telecommunication“ an der technischen Universität von Stettin erlangte, ist die Umsetzung erfolgt.

4.5.1 Datenbankeinträge

Die Module werden Bestandteil des existierenden Coresystems. Die „Intelligenz“ dieses Systems wird bei der Bearbeitung genutzt. Für die Module erforderliche Parameter stehen zentral in der Datenbank zur Verfügung. Die Parameter werden vom MasterService, der direkten Zugriff auf die Projektdatenbank hat, an den SlaveService übermittelt. Eine Modifizierung der Datenbanktabellen ist erforderlich, damit die drei entwickelten Module auf die benötigten Parameter zugreifen können. In der Tabelle „ps_parameter“ werden die Tupel:

```
-LPS_Ziellaufwerk,  
-LPS_Flughöhe,  
-LPS_GSD_Ortho,  
-LPS_DOP_Properties
```

hinzugefügt. Die Einträge für den Projektnamen und die Bild ID sind bereits vorhanden. Die konkreten Attributwerte müssen projektabhängig editiert werden. Als feste Parameterwerte werden die Pfade zu den Batch-Dateien eingetragen.

Die drei Datensätze haben die Bezeichnungen:

```
-call_LPS_BlockFileCreator_rgb_8bit,
-call_LPS_DTMExtractor_rgb_8bit,
-call_LPS_OrthoResampler_rgb_8bit
```

Damit das Coresystem die neuen Prozesse auch starten kann, müssen diese in die Tabelle `ps_process_list` aufgenommen werden. BlockFileCreator, DTMExtractor und OrthoResampler bekommen eindeutige ID's zugewiesen.

process_id	process_name
63	LPS_BlockFiles_rgb_8bit
72	LPS_DTMExtraction_rgb_8bit
78	LPS_OrthoResampler_rgb_8bit

Tabelle 4.2: Auszug aus der Tabelle „ps_process_list“

Die drei Applikationen müssen pro Bild ID sequentiell abgearbeitet werden. Deshalb ist es notwendig die Abhängigkeiten der Aktionen zueinander zu deklarieren. In der Tabelle `ps_process_child_map` wird den Prozessen der jeweilige Folgeprozess zugewiesen.

process_id	child_id
63	10
72	63
78	72

Tabelle 4.3: Auszug aus der Tabelle „ps_process_child_map“

Aus den Beziehungen zwischen den einzelnen Tabellen erstellt das Coresystem weiter Abhängigkeitstabellen für den sequentiellen Programmablauf. Aus der `ps_photolist` und der `ps_processlist` ergibt sich die `ps_photo_process_list`. In diese wird während der Berechnungen der Bearbeitungsstatus der einzelnen Applikationen eingetragen.

photo_process_id	photo_id	process_id	process_status
1	00096	63	
2	00096	72	
3	00096	78	
16	00098	63	
17	00098	72	
18	00098	78	

Tabelle 4.4: Auszug aus der Tabelle „ps_photo_process_list“

Die Tabelle `status_list` definiert die möglichen Werte, die als Status in die `ps_photo_process_list` eingetragen werden.

Value	Name	Description
0	PS_PROCESS_SLEEP	Process is sleep (execution has been suspended)
1	PS_PROCESS_READY	Process is ready to be executed
2	PS_PROCESS_QUEUED	Process has been queued for execution, but it has not been run yet.
3	PS_PROCESS_EXECUTING	Process is currently being executed.
4	PS_PROCESS_FINISHED	Process finished without any detectable errors
5	PS_PROCESS_ERRONEOUS	Process finished with errors.
6	PS_PROCESS_DRAW	Process preceding it has not yet finished their execution. Once the process will become ready, it will shift the status to PS_PROCESS_READY

Tabelle 4.5: Tabelle „status_list“

Abgeleitet von der `ps_process_child_map` und der `ps_photo_process_list` ergibt sich die nächste Relation: `ps_photo_process_child_map` (Tabelle 4.5) Hier ist die Reihenfolge der Prozesse festgelegt.

photo_process_id	photo_process_child_id
1	2
2	3
3	
16	17
17	18
18	

Tabelle 4.6: Auszug aus der Tabelle „ps_photo_process_child_map“

4.5.2 Modul „BlockFileCreator“

Das Modul „BlockFileCreator“ liest zuerst die Parameter, die es benötigt um einen Bildblock zu erstellen. Das sind der Pfad zum `template.blk`, der Pfad zum `eo_computed.txt` und die ID des zu bearbeitenden Bildes. Damit das Modul flexibel einsetzbar ist, werden auch der Projektname, die Imagemask und das Ziellaufwerk als variable Parameter übernommen. Mit dem Befehl `ehfa_FileOpen()` wird geprüft, ob das `template` vorhanden ist. Ob das `eo_computed.txt` existiert wird ebenfalls sichergestellt. Die Verfügbarkeit der beiden zu importierenden Bilder wird mit `ehfa_ObjectRead()` kontrolliert. Sobald eine dieser Dateien fehlt wird ein entsprechender Eintrag in eine log-Datei geschrieben. In diesem Fall kann kein neuer Block erstellt werden. Sind alle Vorbedingungen erfüllt, wird der neue Stereoblock erstellt. Dies geschieht mit Funktionen des IMAGINE Developers‘ Toolkit. Dazu wird mit `ehfa_ObjectRead()` das `template` gelesen, dann wird der Dateiname geändert und anschließend wird die neue Datei mittels `ehfa_FileCopy()` geschrieben. Nachdem der Block erzeugt ist, werden jetzt einige Objektknoten der Datei editiert. Im kopierten Block stehen noch die originalen Bezeichnungen aus dem `template.blk`. Konkret bedeutet das, dass in den Knoten `FrameData:Frame0` und `FrameData:Frame1` die Einträge geändert werden, in denen Bildnummer, äußere Orientierung und Eckkoordinaten enthalten sind. In der unten aufgelisteten EHFA Struktur sind diese Werte farblich markiert.

```
struct Image_data {
    long id = 1
    char *Description[0] = ""
    char ^ImageName[45]=
    "h:/wa200819/lvl3/wa200819_00015_rgb_8bit.tif"
    enum ImageActive = TRUE
    enum PyramidExist = TRUE
    enum InteriorOrientationExist = TRUE
    enum ExteriorOrientationExist = TRUE
    enum OrthoCreated = FALSE
    enum ImageIsOnline = TRUE
    long *PointIDList[0] =
    long ImageColumns = 11500
    long ImageRows = 7500
    enum ValidCorners = TRUE
```

```

double ImageCorners[8] = 629924.03, 5894554.5, 629840.65,
5898857.8, 632707.74, 5899012.4, 632807.4, 5894516.9
struct Camera_Frame {
char *FrameName[45] =
h:/wa200819/lvl3/wa200819_00015_rgb_8bit.tif"
char *ModelName[15] = "Digital Camera"
long CameraIndex;
enum {Right,Down,Left,Up} FiducialOrientation;
Egda_BaseData* FilmFiducials;
Egda_BaseData* ImageFiducials;
Egda_BaseData* FiducialColors;
Egda_BaseData* FiducialDefined;
BOOL InteriorSolution;
Efga_Polynomial* ImageToFilm;
Efga_Polynomial* FilmToImage;
double InitExterior[6] = 644746.04, 5826126.3, 4251.8552,
2.988361, 2.265355, -87.976519
double StandardDeviation[6] = 0, 0, 0, 0, 0, 0
enum Status[6] = Fixed Fixed Fixed Fixed Fixed Fixed
enum ExteriorSolution = TRUE
double FinalExterior[6] = 644746.04, 5826126.3, 4251.8552,
0.052156744, 0.039537908, -1.5354801
double *AdditionalParameters[0] =
double FinalInteriorParameters[3] = 101.4, 0, 0
enum EarthCurvatureIsApplied = FALSE
enum AutoInterior = FALSE
double PixelSize[2] = 0, 0
    } FrameInfo
} Image_data

```

Abbildung 4.11: Auszug aus der EHFA Struktur

Für die Bearbeitung der markierten Einträge wird mit `printf()` die korrekte Information in diese Zeilen geschrieben. `Frame0` und `Frame1` sind die beiden Bilder, aus denen sich der Block zusammensetzt. `Frame1` bekommt die Bezeichnung „projektname“_„Bild-ID“_rgb_8bit.tif. `Frame0` ist das Vorgängerbild und bekommt demzufolge die Bezeichnung „projektname“_„Bild-ID-1“_rgb_8bit.tif.

Die sechs Werte der äußeren Orientierung werden ebenfalls mit `printf()` in die Datei geschrieben. Zuvor liest eine Funktion die Zeilen für `Frame1` und `Frame0` aus dem `eo_computed.txt` aus. Diese Werte werden anschließend für die Berechnung der Eckkoordinaten verwendet. Dazu ist eine weitere Hilfsfunktion notwendig. Sie nutzt zusätzlich zu den sechs Parametern der äußeren Orientierung die drei Parameter der inneren Orientierung. Die innere Orientierung wird durch die Bildkoordinaten `o`, `o` des Bildhauptpunktes und die Kamerakonstante `c`

beschrieben. Diese werden vom Kamerahersteller ermittelt und bekannt gegeben. Die Gleichungen:

$$X=X_o+(Z-Z_o) \frac{r_{11}(\xi - \xi_o) + r_{12}(\eta - \eta_o) - r_{13}C}{r_{31}(\xi - \xi_o) + r_{32}(\eta - \eta_o) - r_{33}C} \quad (4.1)$$

$$Y=Y_o+(Z-Z_o) \frac{r_{21}(\xi - \xi_o) + r_{22}(\eta - \eta_o) - r_{23}C}{r_{31}(\xi - \xi_o) + r_{32}(\eta - \eta_o) - r_{33}C} \quad (4.2)$$

ermöglichen die Berechnung der vier Ecken der Bilder.

Eine Herleitung befindet sich im Anhang.

Die Editierung der Kopie ist somit abgeschlossen. Der Name der fertigen Datei setzt sich aus den festen Bestandteilen „projektname“_„Bild-ID“_rgb_8bit.blk zusammen. Diese kann – wenn vom Nutzer gewünscht - mit dem GUI weiterverarbeitet werden. Geplant ist aber die weitere Verwendung im Modul „DTMExtractor“.

4.5.2.1 Modul „BlockFileCreator“ aufrufen

Der Aufruf des Moduls erfolgt über eine Batch-Datei. Die Kommandozeile für das Ausführen beinhaltet, zusätzlich zum Pfad der zur Applikation führt, die Parameter, die im Programm benötigt werden. Diese Parameter sind der Projektname, die Bild ID, das Ziellaufwerk, die Imagemask, der Pfad zum template.blk, der Pfad zum eo_computed.txt und die durchschnittliche Flughöhe. Ein beispielhafter Aufruf sieht wie folgt aus:

```
"C:\Programme\IMAGINE 8.7\bin\ntx86\BlockFileCreator.exe" wa200819
25551 "H:" wa200819_#_rgb_8bit.tif "H:\wa200819\blk\template.blk"
"H:\wa200819\eo_daten\eo_computed.txt" 4300
```

Abbildung 4.12: Anwendungsaufruf

Sämtliche Parameter sind in der Projektdatenbank eingetragen. Bereitgestellt werden sie vom Masterservice, der direkten Zugriff auf die Datenbank hat. Diese bereitgestellten Parameter werden in der Batch als Variablen definiert.

```
@SET ERDAS_BIN=C:\Programme\IMAGINE 8.7\bin\ntx86
@SET LPS_UTILITY=C:\arcforest\process\LPS
@SET LPS_BLOCK_FILE_CREATOR=%ERDAS_BIN%/BlockFileCreator.exe
@SET PROJECT_NAME=%1
@SET IMAGE_ID=%2
@SET STORAGE_DRIVE=%3
@SET LPS_DRIVE=%4
@SET IMAGE_MASK=%PROJECT_NAME%#_rgb_8bit.tif
@SET BLOCK_TEMPLATE=%LPS_DRIVE%\%PROJECT_NAME%\blk\template.blk
@SETEOCOMPUTED=%LPS_DRIVE%\%PROJECT_NAME%/eo_daten/eo_computed.txt
@SET AVERAGE_FLIGHT_HEIGHT=%5
```

Abbildung 4.13: Definieren von Variablen

Die Kommandozeile verändert sich aufgrund dessen zu:

```
"%LPS_BLOCK_FILE_CREATOR%" %PROJECT_NAME% %IMAGE_ID%
"%STORAGE_DRIVE%" %IMAGE_MASK% "%BLOCK_TEMPLATE%" "%EOCOMPUTED%"
%AVERAGE_FLIGHT_HEIGHT%
```

Abbildung 4.14: Anwendungsaufruf mit Variablen

Ist die Applikation „BlockFileCreator“ durchlaufen, wird abschließend geprüft, ob die neue blk-Datei tatsächlich erzeugt wurde. Das Ergebnis wird in die Datenbank eingetragen und zusätzlich in eine log-Datei. Hierfür liest die Funktion `dblogwriter.exe` das temporäre log-File des „BlockFileCreator“ aus und schreibt das Resultat in die Datenbank. Ist der Block für die zugewiesene ID erfolgreich erstellt worden, dann wird der Nachfolgeprozess „DTMExtractor“, für die diese ID auf den Status `1=ready to process` gesetzt.

4.5.3 Modul „DTMExtractor“

Der Zweck des „DTMExtractor“ ist die Berechnung des Höhenmodells für einen Bildblock. Als Parameter benötigt das Modul die ID des zu bearbeitenden Bildblocks, die Imagemask, den Verzeichnispfad zur blk-Datei und den Verzeichnispfad für das zu generierende DTM. Der Pfad zur Funktion `erdasJobRunner.exe` wird als weiterer Parameter gelesen. Zu Beginn prüft das Modul das Vorhandensein der Ausgangsobjekte. Das sind zum Einen die blk-Datei und zum Anderen die beiden Bilder, die den Stereoblock erzeugen. Wie

im vorangegangenen Prozess geschieht das mit den Befehlen `ehfa_FileOpen()` und `ehfa_ObjectRead()`. Tritt ein Fehler auf, wird dies in ein temporäres log-File geschrieben. Für den Normalfall, dass alle Voraussetzungen erfüllt sind, folgt die Aktualisierung der hierarchischen Struktur der blk-Datei. In den dtm-spezifischen Objektknoten werden die Einträge geändert, die noch vom `template.blk` stammen.

Die markierten Zeilen der unten aufgeführten Knoten bedürfen einer Editierung.

```
struct AtmProject {
    char *DtmFilePrefix[33] = "h:/wa200819/dtm/template_tin.pro"
    char *DtmFileVersionSuffix[2] = "0"
    char *DisplayCellUnits[7] = "meters"
    double DtmCellSize[2] = 30, 30
    enum OutputForm = ATM_SINGLEFILE
    enum OutputType = ATM_OUTPUTTIN
    double OverlapShrink = 0
    double OverlapLimit = 0.1
    double DtmTrimPercent = 0
    double DemBackground = 2e+037
    char *BlkFileName[29] = "h:/wa200819/blk/template.blk"
} AtmProject
struct ImagePair {
    enum Active = TRUE
    long ID = 1
    char *Name[48] =
"wa200819_00015_rgb_8bit_wa200819_00016_rgb_8bit"
    enum DtmGenerated = FALSE
    double DtmCellSize[2] = 4, 4
    char *DtmFileName[56] =
"dtm_wa200819_00015_rgb_8bit_wa200819_00016_rgb_8bit.img"
    char *ReportFileName[56] =
"dtm_wa200819_00015_rgb_8bit_wa200819_00016_rgb_8bit.rpt"
    double ActiveArea[4] =
    long StopPydNum = 0
    long MaxAllowStopPyd = 2
    double OverlapAmount = 0.60987623
} ImagePair

//Every ImagePair has 2 FrameModels:
struct FrameModel {
    double ImageHeight = 7500
    double ImageWidth = 11500
    long FrameIndex = 1
    char *ImageName[45] =
"h:/wa200819/lvl3/wa200819_00016_rgb_8bit.tif"
    double MeanElevation = 192.755
} FrameModel

//4 x Corner structure for every FrameModel:
struct Point2D {
    double X;
    double Y;
};
```

Abbildung 4.15: Auszug aus dtm-relevanten Objektknoten

Für die Bearbeitung wird der Befehl `sprintf()` verwendet. Die Veränderungen setzen sich aus den Parameterwerten zusammen, die das Modul zu Beginn gelesen hat. So wird der originale Eintrag `"h:/wa200819/dtm/template_tin.pro"` ein durch Variablen umschriebener Eintrag für den DTM-Namen:

```
"Laufwerk":/"projektname"/dtm/"projektname"_"ID".pro"
```

Die Bezeichnung `BlkFileName[29] = "h:/wa200819/blk/template.blk"` wird geändert zu:

```
"Laufwerk":/"projektname"/blk/"projektname"_"ID"_"rgb_8bit.blk".
```

Einträge, die Bildnummern beinhalten, werden, wie im `BlockFileCreator`, dahingehend abgewandelt, dass `Frame0` und `Frame1` die Bezeichnung `„projektname"_"ID-1"_"rgb_8bit.tif` und `„projektname"_"ID"_"rgb_8bit.tif` erhalten.

Weitere Modifizierungen sind für die vier Eckpunkte jedes Bildes nötig. Dazu werden im Knoten: `AtmProject:ImagePairList:ImagePair0` die Werte, die mit den Gleichungen (4.1) und (4.2) aus Kapitel 4.5.2 ermittelt werden, eingetragen. Damit sind die, für die DTM Berechnung, relevanten Strukturen aktualisiert. Anschließend wird der Befehl zur DTM Berechnung in das ERDAS-command eingetragen. Die mit dem `Imagine Developers Toolkit` erstellte Funktion `erdasJobRunner` stellt die Verbindung zur ERDAS-Session her und übergibt den

Befehl:

```
c:/programme/Imagine 8.7/bin/ntx86/eatm_batch.exe
```

```
"Laufwerk":/"projektname"/blk/"projektname"_"ID"_"rgb_8bit.blk" -m
```

Das Höhenmodell wird nun berechnet. Im ERDAS Session-log wird der Prozess dokumentiert. Ist die Berechnung abgeschlossen, wird auch die Verbindung vom `erdasJobRunner` zur ERDAS-Session beendet. Die `eatm_batch.exe` erstellt das Höhenmodell und dazugehörend ein `report-file`. Beide werden für einen ordnungsgemäßen Ablauf des anschließenden Moduls, „OrthoResampler“ benötigt.

4.5.3.1 Modul „DTMExtractor“ aufrufen

Wie auch beim vorangegangenen Modul erfolgt der Start über eine Batch-Datei. Die Kommandozeile für das Ausführen beinhaltet, zusätzlich zum Pfad der zur Applikation führt, die Parameter, die im Programm benötigt werden. Diese Parameter sind: der Projektname, die Bild ID, das Ziellaufwerk, die Imagemask, das Verzeichnis für die blk-Datei, das Verzeichnis für das DTM und der Pfad zur erdasJobRunner-Funktion. Ein beispielhafter Aufruf sieht wie folgt aus:

```
"C:\Programme\IMAGINE 8.7\bin\ntx86\DTMExtractor.exe"  
25551 wa200819_#_rgb_8bit.tif "H:\wa200819\blk" "H:\wa200819\dtm"  
"C:\Programme\  
IMAGINE 8.7\bin\ntx86\erdasJobRunner.exe"
```

Durch die Definition der Datenbankparameter als Variablen, gestaltet sich die aufrufende Zeile wie folgt:

```
"%LPS_DTM_EXTRACTOR%" %ID% %IMAGE_MASK% "%BLOCK_FILES_FOLDER%"  
"%DTM_FILES_FOLDER%" "%ERDAS_JOB_RUNNER%"
```

Diese Batch ist ebenfalls so gestaltet, dass sie zum Abschluss prüft, ob das Höhenmodell tatsächlich erzeugt wurde. Mit der Funktion `dblogwriter.exe` wird das Resultat in die Datenbank geschrieben. Ist das Höhenmodell für die ID erfolgreich erstellt, dann wird der Nachfolgeprozess „OrthoResampler“, für diese ID auf den Status `1=ready to process` gesetzt.

4.5.4 Modul „OrthoResampler“

Die Aufgabe des „OrthoResampler“ besteht darin, den Befehl `resampleprocess.exe` in der ERDAS-Umgebung auszuführen, mit dem Ziel, das Orthophoto für das zugewiesene Bild „ID“ zu berechnen. Zum Abschluss der Applikation muss der Aufruf in der Kommandozeile folgende Syntax aufweisen:

```
c:/programme/imagene 8.7/bin/ntx86/resampleprocess.exe
„Input_Image_file“ „output_image_file“ -xform
„resampling_property_file“ -deleteflag -meter orthoresample -i
Bilinear Interpolation -u meters -p „projection_information“ -map
„dtm_upperleft_corner“ „dtm_lowerright_corner“ „pixel_size_X“
„pixel_size_Y“
```

Abbildung 4.16: Befehlssyntax beim `resampleprocess`

Das `resampling_property_file` enthält wichtige Informationen zum Resamplen. Wie beim `template.blk` wird die Vorlage dieser Datei kopiert und anschließend, mit den spezifischen Werten für die betreffende ID, aktualisiert. Im Objektknoten: `GenericXForm:XForm0` müssen zwei Einträge editiert werden.

```
char *demfilename[30] = "H:/wa200819/dtm/tempDem_25551"
und double coeffs[9] = 101.4,0,0,644746.04,5826126.3, 4251.8552,
0.052156744, 0.039537908, -1.5354801.
```

Die in der Zeile: `double coeffs[9]` hervorgehobenen Werte sind die sechs Parameter der äußeren Orientierung des Luftbildes. Die vorhandenen Werte werden mit den korrekten Daten, die aus dem `blk`-file gelesen werden, ersetzt. Modifizierung zwei betrifft die Zeile: `char *demfilename[30] =`. An dieser Stelle wird auf eine ASCII-Datei verwiesen, die Informationen, zu dem zu verwendenden Höhenmodell enthält. Der Eintrag wird zu: `„Laufwerk“:/„projektname“/dtm/tempDem_„ID“` abgeändert.

Bevor auf das `tempdem`-file verwiesen werden kann, muss es erzeugt werden. Nach dem unten aufgezeigten Schema, wird zur ID eine solche Datei generiert. Die Koordinaten der DTM-Boundingbox werden aus dem `report`-file des DTM-Prozesses entnommen.

```
IMAGINE Rasterized TIN Access"Laufwerk":/"projektname"/dtm/"projektname"_"ID".pro
1 1
"dtm_boundingbox_ul_X lr_X_lr_Y_ul_Y"
0.000000000000
```

Abbildung 4.17: Syntax des `tempdem`-file

Um die beschriebenen Teilschritte ausführen zu können, verarbeitet das Modul „OrthoResampler“ mehr Parameter als die beiden Vorgängerapplikationen. (siehe Abb.4.18) Eingangs werden die Komponenten überprüft, die physikalisch existieren müssen. Dazu gehören:

- der Bildblock
- das DTM
- das Luftbild
- das Resampling_Property_File

Das Prüfergebnis wird in ein temporäres log-File geschrieben. Fehlen eins oder mehrere dieser Elemente, wird das Modul sofort beendet. Sind alle Elemente vorhanden, wird das tempdem-file erzeugt, das resampling_property_file wird aktualisiert und der Befehl `resampleprocess()` kann im ERDAS ausgeführt werden. Der erdasJobRunner stellt die Verbindung zur ERDAS-Session her und erzeugt den geforderten Aufruf (vgl. Abb. 4.16) in der ERDAS-Kommandozeile. Die einzelnen Werte dafür setzen sich aus den definierten Parametern, aus fixen Elementen und aus den Informationen des tempdem-file zusammen.

4.5.4.1 Modul „OrthoResampler“ aufrufen

Damit das Modul vom Coresystem gestartet werden kann, ist die passende Batch-Datei unumgänglich. Sie ähnelt stark der Batch des „DTMExtractor“ und weicht nur geringfügig in der Anzahl und Reihenfolge der Parameter von ihr ab. Abbildung 4.18 zeigt den Aufruf mit Variablen, so wie er in der Batch zu finden ist und Abbildung 4.19 zeigt ein konkretes Beispiel.

```
"%LPS_ORTHO_RESAMPLER%" %PROJECT_NAME% %ID% %IMAGE_MASK%  
"%BLOCK_FILES_FOLDER%" "%DTM_FILES_FOLDER%" "%ORTHO_FILES_FOLDER%"  
"%RESAMPLING_PROPERTY_FILE_PATH%" %RESAMPLING_CELL_SIZE%  
"%ERDAS_JOB_RUNNER%"
```

Abbildung 4.18: Befehlssyntax der OrthoResampler.bat

```
"C:\Programme\IMAGINE 8.7\bin\ntx86\OrthoResampler.exe" wa200819  
25551 wa200819_#_rgb_8bit.tif "H:\wa200819\blk" "H:\wa200819\dtm"  
"H:\wa200819\ortho" "H:\wa200819\blk\eb1k_001116" 0.35  
"C:\Programme\IMAGINE 8.7\bin\ntx86\erdasJobRunner.exe"
```

Abbildung 4.19: Befehlssyntax der OrthoResampler.bat

5. Vergleich der Ergebnisse

Um eine Aussage treffen zu können, inwieweit die entwickelten Applikationen, die in der Aufgabenstellung geforderten Bedingungen erfüllen, ist ein Vergleich zwischen dem alten und dem neuen Workflow durchzuführen. Für die Gegenüberstellung ist ein Testprojekt, von circa 500 Luftbildern, mit beiden Methoden bearbeitet worden. Die durchschnittliche Flughöhe betrug 4300m, der Punktabstand für das DTM wurde mit 30m gewählt und die Pixelgröße im Orthophoto war auf 0.35m festgesetzt. Dabei wurde der manuelle Workflow in zwei unterschiedlichen Varianten durchlaufen. Einmal wurden Bildblöcke mit jeweils 50 Bildern erstellt. Diese Variante diente zum zeitlichen Vergleich mit dem automatisierten Workflow. Bei der zweiten Version wurden Abweichungen in der Qualität untersucht. Hierfür sind die Blöcke mit je zwei Luftbildern erstellt worden.

5.1 Qualitativer Vergleich

Bei der Benutzung der grafischen Schnittstelle und bei den automatisch ablaufenden Applikationen werden für die DTM-Berechnung und die Orthophoto-Berechnung dieselben Funktionen verwendet. Da die Stereobildblöcke, die der „BlockFileCreator“ erstellt, eine modifizierte Kopie und kein Original sind, besteht trotzdem die Möglichkeit, dass Fehler auftreten können. Eine Aussage über die Qualität kann nicht über einen Vergleich der blk-Dateien erfolgen. Die generierten Produkte der Module DTMExtractor und OrthoResampler eignen sich dazu besser. Die verglichenen Höhenmodelle haben dieselbe Anzahl an extrahierten Höhenpunkten. Der Vergleich der Höhenpunkte erbrachte identische Koordinatenwerte in beiden Methoden. Die Gegenüberstellung der Orthophotos zeigt ein entsprechendes Resultat: die unterschiedlich produzierten Dateien sind absolut deckungsgleich. Die Qualität der Endprodukte wird durch den Automatisierungsvorgang nicht beeinflusst.

5.2 Quantitativer Vergleich

Der Faktor Zeit stellt eine weitere Möglichkeit dar, die beiden Arbeitsabläufe zu unterscheiden. Für beide Arbeitsmethoden wurden die Zeitmessungen in personellen Zeitaufwand und maschinellen Zeitaufwand unterteilt.

Um einen Bildblock für die Orthophotoberechnung vorzubereiten, der 50 Bilder enthält, bedarf es durchschnittlich 15 Minuten mit dem GUI. Für den Vorlageblock (template.blk) der automatisierten Methode benötigt man weniger Ladezeit beim Importieren der Bilder, sodass hier nur 12 Minuten vergehen. Bei 500 Bildern und 50 Bilder pro Block, in der manuellen Methode, muss der Vorgang zehnmal durchgeführt werden. Somit stehen 150 Minuten zu Buche. Das Modul BlockFileCreator benötigt nur zwei Sekunden um eine Datei zu erzeugen. Im Gegensatz zu 15 Minuten, ein Wert, den man fast vernachlässigen kann. Bei größeren Projekten mit höherer Bildanzahl steigt der Zeitgewinn exponentiell an. Das Diagramm veranschaulicht den deutlichen Unterschied.

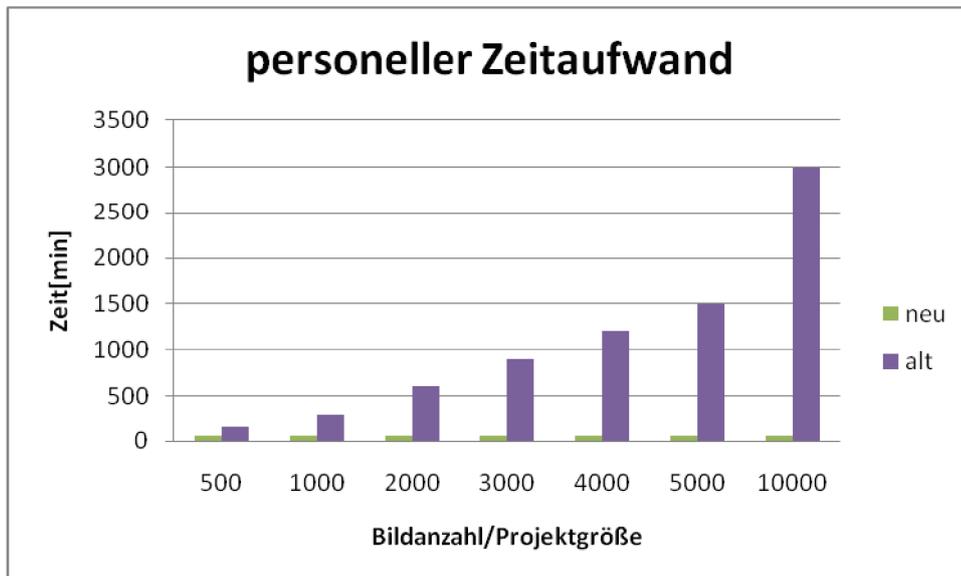


Abbildung 5.1: Vergleich - Zeitaufwand

Der Nutzer hat durch die neuen Module viel weniger manuellen Aufwand. Er muss lediglich einen Stereoblock erstellen. Mit zunehmender Projektgröße (steigende Bildanzahl) wird der zeitliche Gewinn immer größer.

Ein weiterer Unterschied ist zwischen den beiden Arbeitsmethoden messbar: die Laufzeiten der Rechenprozesse. Bei der DTM-Generierung sind die Rechenzeiten noch annähernd gleich. Für das Testprojekt ist eine durchschnittliche Rechenzeit von 45s pro Überlappungsbereich ermittelt worden. Bei der Variante mit 50 Bildern pro Block, wird das DTM als eine große Datei gespeichert, die sich aus 49 Überlappungsbereichen zusammensetzt. Währenddessen wertet die neue Methode nur jeweils einen Bereich aus und speichert diesen. Diese Tatsache kommt im nächsten Schritt, dem OrthoResampling, zum tragen. Bereits das Lesen der größeren Datei dauert spürbar länger. Doch relevanter ist die Dauer des Resamplings. Das zusammenhängende Höhenmodell deckt die gesamte Bildfläche der Luftbilder ab und so wird auch das gesamte Luftbild entzerrt. Im neuen Workflow wird, projektabhängig, 60-80% des Luftbildes entzerrt. (siehe Abb. 5.2 und 5.3) Das schlägt sich in der Rechenzeit und im Speicherbedarf der Datei nieder.

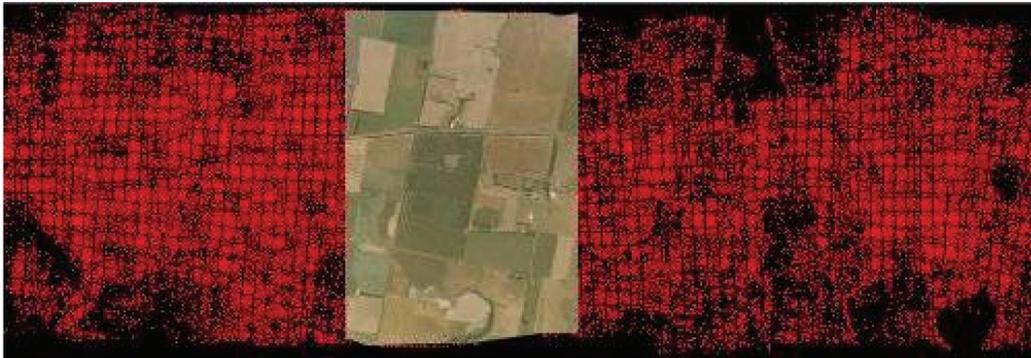


Abbildung 5.2: herkömmlich Methode



Abbildung 5.3: automatisierte Methode

Die gemittelte Zeit mit der herkömmlichen Methode betrug 4,64min. Der Speicherbedarf eines Bildes belief sich auf 290MB. Dem entgegen stehen 2,5min Laufzeit und 140MB Dateigröße mit der aktuellen Methode. Die kumulierten Rechenzeiten ergeben Werte von 2320min (alt) zu 1250min (neu). Die Zeiten relativieren sich jedoch noch. Bei der herkömmlichen Methode wurden die zehn Blöcke, durch den Anwender, auf zehn Rechner verteilt. Bei der neuen Methode übernimmt das Coresystem die Verteilung. Das bedeutet weniger Arbeit für den Anwender – oder mehr Zeit für andere Tätigkeiten. Der Punkt Speichernutzung ist ebenfalls nicht außer Acht zu lassen. Es werden knapp 50% der Ressourcen gespart.

6. Zusammenfassung und Ausblick

Die wachsende Nachfrage an aktuellen Geoinformationen und die daraus resultierend steigende Zahl von geoinformationsverarbeitenden Unternehmen zwingt viele Dienstleister dazu bestehende Arbeitsabläufe weiter zu verbessern. Die IFMS hat sich dieser Herausforderung gestellt. Der manuelle Orthophoto-Workflow soll durch den optimierten Einsatz der vorhandenen Mittel automatisiert werden. Die Arbeitszeiten und Fehlereinflüsse sollen minimiert werden.

Ein Anwendungspaket, bestehend aus drei Teilen, übernimmt jetzt den Großteil der Arbeiten, die vorher, im LPS, von Hand ausgeführt wurden. Nachdem die drei Module installiert und getestet worden sind, kann man sagen, dass ein funktionsfähiges Basissystem erstellt wurde. Der Vergleich mit den alten Handlungsabläufen zeigt, dass 50 Prozent an Speicherkapazität eingespart werden können.

Der Anwender selbst übernimmt kaum noch manuelle Arbeitsschritte und kann so nur sehr wenig Fehler einbringen. Durch die Nutzung der entwickelten Applikationen, in Verbindung mit dem Coresystem ist der Zeitgewinn immens. Die geforderten Funktionalitäten werden vom Werkzeug erfüllt.

Bei genauer Betrachtung des Erreichten, zeigt sich jedoch, dass die Funktionalität noch ausbaufähig ist. So ist geplant, Kontrollmechanismen für die Höhenmodelle einzubauen. Die Koordinatentripel der DTM-Dateien könnten auf grobe Fehlerwerte untersucht werden. Diese "Ausreißer" könnten dann gelöscht oder angepasst werden. Weiterhin ist vorstellbar, dass mehrere Dateiformate angeboten werden. Momentan wird das DTM ausschließlich im Format TerraModel TIN abgespeichert. Ein weiterer Punkt für zukünftige Projekte, ist die Integration von extern bereitgestellten Höhenmodellen.

Ogleich der Auftraggeber mit dem Ergebnis zufrieden ist, ist der jetzt geschaffene Workflow noch weiter optimierbar.

Anhang

Die Berechnung der vier Eckkoordinaten eines Bildes, bei bekannter innerer Orientierung und bekannter äußerer Orientierung erfolgt nach den Formeln[Kraus1]:

$$X=X_0+(Z-Z_0) \frac{r_{11}(\xi - \xi_0) + r_{12}(\eta - \eta_0) - r_{13}C}{r_{31}(\xi - \xi_0) + r_{32}(\eta - \eta_0) - r_{33}C} \quad (\text{A.1})$$

$$Y=Y_0+(Z-Z_0) \frac{r_{21}(\xi - \xi_0) + r_{22}(\eta - \eta_0) - r_{23}C}{r_{31}(\xi - \xi_0) + r_{32}(\eta - \eta_0) - r_{33}C} \quad (\text{A.2})$$

- X, Y, Z sind die Objektkoordinaten der gesuchten Bildeckpunkte
- ξ_0 und η_0 sind die Bildkoordinaten des Bildhauptpunktes
- C ist die Kamerakonstante
- X_0, Y_0, Z_0 sind die Koordinaten des Aufnahmeortes
- ω, φ, κ bezeichnen die 3 Drehungen des Bildes im Moment der Aufnahme
- Die Parameter r_{ik} sind die Elemente der räumlichen Drehmatrix

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

Die Gleichungen A.1 und A.2 lassen sich aus dem mathematischen Zusammenhang zwischen Bild- und Objektkoordinaten herleiten.

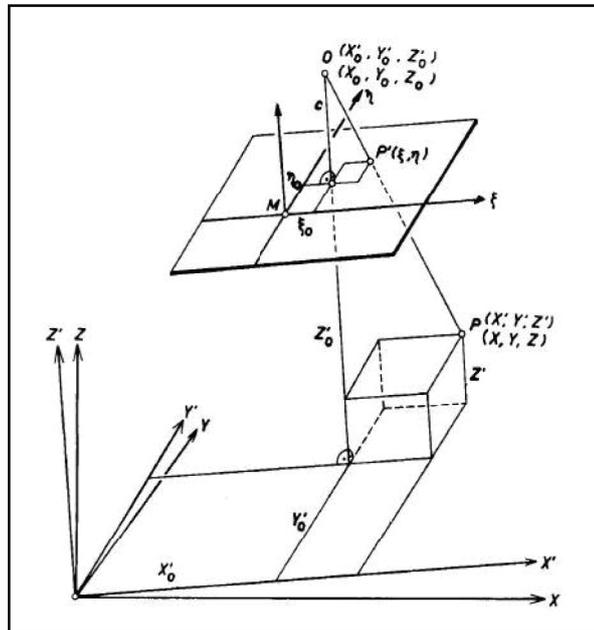


Abbildung 7.1: Geradlinige Verbindung des Aufnahmeortes O mit dem Bildpunkt P' und dem Objektpunkt P (Kollinearitätsbeziehung) [Kraus1]

Der Objektpunkt P , der Bildpunkt P' und der Aufnahmeort O liegen auf einer Geraden.

$$\frac{\xi - \xi_0}{c} = \frac{X' - X_0'}{Z_0' - Z'} \quad (A.3)$$

$$\frac{\eta - \eta_0}{c} = \frac{Y' - Y_0'}{Z_0' - Z'}$$

A.3 umgestellt nach den Bildkoordinaten:

$$\xi = \xi_0 - c \frac{X' - X_0'}{Z_0' - Z'} \quad (A.4)$$

$$\eta = \eta_0 - c \frac{Y' - Y_0'}{Z_0' - Z'}$$

Die Koordinaten X' des Objektpunktes und X_0' des Aufnahmeortes kann man durch die Koordinaten X bzw. X_0 des Objektkoordinatensystems ersetzen, wobei beachtet werden muss, dass beide Systeme räumlich verdreht sind:

$$\begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} X' - X_0' \\ Y' - Y_0' \\ Z' - Z_0' \end{pmatrix} \quad (\text{A.5})$$

Durch Einsetzen dieser Gleichung in die beiden Vorherigen erhält man die gesuchte Beziehung zwischen Bild- und Objektkoordinaten, wobei die Gleichung A.5 zuvor mit $R^T=R^{-1}$ zu multiplizieren ist:

$$\xi = \xi_0 - c \frac{r_{11}(X - X_0) + r_{21}(Y - Y_0) - r_{31}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) - r_{33}(Z - Z_0)} \quad (\text{A.6})$$

$$\eta = \eta_0 - c \frac{r_{12}(X - X_0) + r_{22}(Y - Y_0) - r_{32}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) - r_{33}(Z - Z_0)}$$

A.6 umgestellt nach den Objektkoordinaten X und Y ergibt die eingangs genannten Gleichungen A.1 und A.2.

Bei der Drehung im Raum ist eine Hierarchie der Achsen zu beachten. Dabei ist bzw. x die Primärachse, φ bzw. y die Sekundärachse und κ bzw. z die Tertiärachse. Die räumliche Drehmatrix der Transformationsgleichung hat in dem Fall die Form:

$$\begin{pmatrix} \cos \Phi \cos \kappa & -\cos \Phi \sin \kappa & \sin \Phi \\ \cos \omega \sin \kappa + \sin \omega \sin \Phi \cos \kappa & \cos \omega \cos \kappa - \sin \omega \sin \Phi \sin \kappa & -\sin \omega \cos \Phi \\ \sin \omega \sin \kappa - \cos \omega \sin \Phi \cos \kappa & \sin \omega \cos \kappa + \cos \omega \sin \Phi \sin \kappa & \cos \omega \cos \Phi \end{pmatrix}$$

Technical Specifications of UCD

Image Product Specification

Image format Analogous to an aerial film image at a format of 23 cm x 15 cm

Image data formats JPEG; TIFF with options for 8 and 16 bits, scan-line, stripped or tiled

Image format (Level 2) Full resolution panchromatic, separate color channels at color resolution

Image format (Level 3) Full resolution (fused) R, G, B, Near-IR channels, planar or pixel-interleaved

Digital Camera Sensor Unit (SU)

Panchromatic image size 11500 * 7500 pixels

Panchromatic physical pixel (detector) size 9 μ m

Physical format of the focal plane 103.5 mm * 67.5 mm

Panchromatic lens focal distance 100 mm

Lens aperture $f = 1/5.6$

Field-of-view, cross-track (along-track) 55° (37°)

Color (multi-spectral capability) 4 channels -- RGB & NIR

Color image size (Level 2) 3680 * 2400 pixels

Color physical pixel (detector) size 9 μ m

Color lens focal distance 28 mm

Color lens aperture $f = 1/4.0$

Color field of view, cross-track (along-track) 61° (42°)

Shutter speed options 1/500 to 1/60 second

Forward-motion compensation (FMC) TDI-controlled

Maximum FMC-capability 50 pixels

Smallest pixels on the ground at flying height of 300 m (1000 feet) 3 cm (1.1 inch)

Frame rate per second (minimum inter-image interval) > 1 frame per second

Analog-to-digital conversion at 14 bits

Radiometric resolution in each color channel better than 12 bit

Physical dimensions of the camera unit 45cm x 45cm x 60 cm

Weight ~45 kg

Power consumption at full performance 150 W

Storage and Computing Unit (SCU)

In-flight storage capacity 1.5 TB (including mirrored data)

Capacity to collect in-flight uncompressed frames 2692

Storage and computing configuration Parallel arrangement with multiple CPUs and disks

Data redundancy Dual disk sets on-board containing mirror images

Data transfer into office environment Removable, dual use as office post-processor

Physical dimensions 40cm x 55cm x 65 cm

Weight ~ 65 kg

Power consumption at full performance 700 W

Quellenverzeichnis

- [bull] Internetservice für Webdesign und Optimierung
<http://www.bullhost.de/>
- [highscore08] Onlineplattform zum Thema Programmierung
<http://www.highscore.de/>
- [hro] Geoinformatik-Service der Universität Rostock
<http://www.geoinformatik.uni-rostock.de/lexikon.asp>
- [IDT] Manual und Whitepaper des ERDAS IMAGINE Developers‘ Toolkit (September 2006)
- [kowoma] <http://www.kowoma.de/gps/index.htm>
Kartenprojektionen (Stand November 2008)
- [Kraus1] Karl Kraus (2004). *Photogrammetrie Band1 - Geometrische Informationen aus Photographien und Laserscanneraufnahmen*. 7., vollständig bearbeitete und erweiterte Auflage
- [Kraus2] Karl Kraus (1996). *Photogrammetrie Band2 –Verfeinerte Methoden und Anwendungen*. 3., völlig neue und erweiterte Auflage. Dümmler / Bonn
- [Kraus3] Karl Kraus (1996). *Photogrammetrie Band3 –Topographische Informationssysteme*. Dümmler
- [Leberl] Leberl F., M. Gruber (2003) *Economical Large Format Aerial Digital Camera*. GIM Internaitonal, Juni 2003.
- [Leica] Internetauftritt des Unternehmens LeicaGeosystems
<http://www.geosystems.de/>
- [LTA] LandesTechnologieAnzeiger 1/2008 *Zukunftsträchtig – Markt der Geoinformationen*. Ulf Klammer

- [Luhmann] Thomas Luhmann (2000). *Nahbereichsphotogrammetrie – Grundlagen, Methoden und Anwendungen*. Herbert Wichmann Verlag
- [msdn] Internetseiten des Microsoft Developers Network
<http://msdn.microsoft.com/de>
- [OLANIS] Internetseiten der OLANIS Expertensysteme GmbH
<http://www.olanis.de/>
- [OMG] Object Management Group <http://www.omg.org/> (Stand 11.09.2007)
- [Pfeiffer] B. Pfeiffer; G. Weimann. *Geometrische Grundlagen der Luftbildinterpretation Einfachverfahren der Luftbilddauswertung* 2.,neubearbeitete Auflage
- [uni-magdeburg] UML – Tutorial
<http://ivs.cs.uni-magdeburg.de/~dumke/UML/inhalt.htm>
- [wiki08] Wikipedia – Die freie Enzyklopedie
<http://de.wikipedia.org/wiki/> (01.12.08-28.02.09)
- Vorlesungsmaterial Angewandte Informatik FH-NB
Software Engineering
- Tutorial für C++ <http://tutorial.schornboeck.net/inhalt.htm>
- Tutorial für C++ <http://www.onlinetutorials.de/cpp-index.htm>
- A. Heuer, G. Saake Datenbanken—Konzepte und Sprachen
<http://www2.informatik.uni-halle.de/lehre/db/heusa.html>