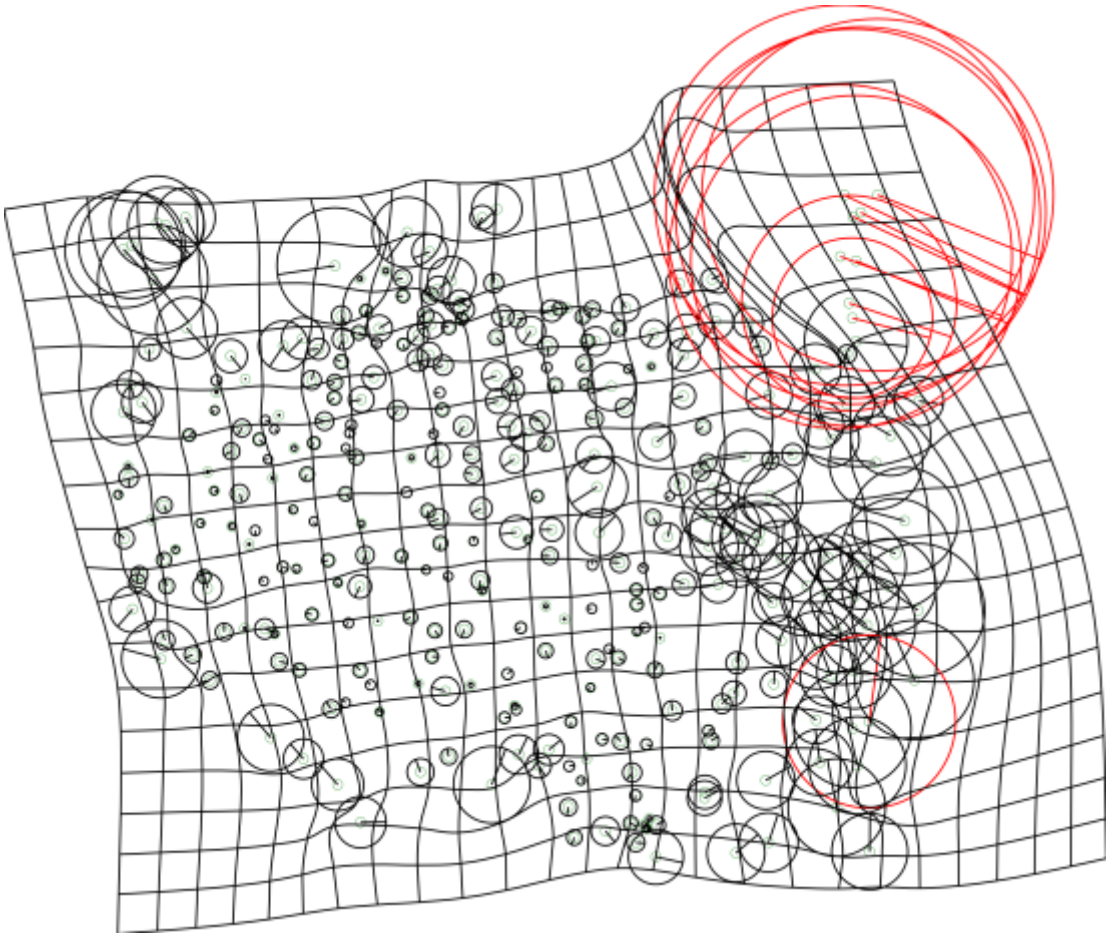
ETH Zürich
Institut für Kartographie

Sommersemester 2005
Geomatik

Software zur Analyse alter Karten
MapAnalyst



Autor
Adrian Weber
Trottenstrasse 16
8037 Zürich

Übungsleitung
Bernhard Jenny
Institut für Kartographie
8093 Zürich

Zusammenfassung

MapAnalyst ist ein neues Programm für die rechnergestützte Untersuchung alter Karten. Der hauptsächliche Verwendungszweck besteht in der Berechnung von Verzerrungsgittern und Fehlervektoren, welche Aussagen über Genauigkeit und Verzerrungen in der alten Karte ermöglichen.
Das Programm verwendet Passpunktpaare in einer alten Karte und in einer neuen Referenzkarte. Die Passpunkte werden zur Berechnung von Verzerrungsgitter verwendet, welches lokale Verzerrungen alter Karten sehr anschaulich darstellt, wie auch zur Berechnung von Fehlervektoren, die die Qualität einzelner Punkte anzeigen.
Zudem rechnet MapAnalyst der Massstab der alten Karte, die Rotation sowie statistische Indikatoren.
Fehlerdarstellungen wie das Verzerrungsgitter oder die Fehlervektoren lassen Gebiete mit ähnlicher Genauigkeit erkennen, was Hinweise auf die Grundlagenkarte geben könnte, die verwendet wurde.

Abstract

MapAnalyst is a new tool for the computer-based analysis of old maps. Its main purpose is the computation of distortion grids and error vectors that illustrate the precision and distortions of old maps.
The software uses pairs of control points on an old map and on a new reference map. The control points are used to construct distortion grids, which are very illustrative representations of local distortions of old maps , and error vectors, which indicate the qualitiy of individual control points. As a by-product, MapAnalyst also computes the old map's scale, rotation and statistical indicators.
Sample applications of distortion grids and error vectors include the identification of areas of similar precision - may give valuable hints on the origin of the different base-maps that were used to compile the map - or the estimation of the positional accuracy of vanished items that were reconstituted from old maps (such as cultural landscapes, villages, roads, buildings etc).

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Eine Genauigkeitsanalyse einer alten Karte kann dem Kartenhistoriker wertvolle Hinweise auf die Herkunft und die Erstellungsweise geben. Die Fehler in alten Karten haben unterschiedliche Ursachen. Sie können in systematische Fehler, die mit der Lage und der Orientierung der Karte bezüglich des geographischen Koordinatensystems zusammenhängen, und in zufällige Fehler, die bei der Aufnahme und Herstellung der Karte entstanden, eingeteilt werden [Forstner und Oehrli, 1998].
Eine Berechnung von Fehlern in alten Karten bezieht sich auf ein heutiges Gegenstück, das als fehlerfrei angenommen wird. Diese kann mit verschiedenen Methoden durchgeführt und die Fehler können auf unterschiedliche Weise dargestellt werden. Bekannte und anschauliche Darstellungen sind zum Beispiel Fehlervektoren, Verzerrungsgitter oder der Konturenvergleich. Für die wissenschaftliche Untersuchung gibt es zahlreiche weitere Darstellungen wie Isodeformaten, die Darstellung nach der Kreismethode oder die Berechnung der Tissot'schen Indikatrix [Forstner und Oehrli, 1998].
Die Darstellung der Fehler lassen Aussagen über die Kartenaufnahme oder eine Abgrenzung der benutzten Quellenkarten gegeneinander zu. Ebenfalls weisen grosse Ausreisser auf lokal falsch eingepasste Ortschaften hin.
Im Rahmen dieser Arbeit soll ein Programm zur interaktiven Analyse alter Karten entwickelt werden. Das Programm soll in einer höheren Programmiersprache geschrieben und als Open-Source-Projekt frei erhältlich sein.
In der Wissenschaft sind schon einige Arbeiten über die Analyse alter Karten gemacht worden. Zu erwähnen sind hier insbesondere [Beineke, 2001], dessen Dissertation grösstenteils die mathematischen Grundlagen für diese Arbeit ist. [Weis, 1985] versuchte in ihrer Diplomarbeit Verzerrungsgitter automatisch berechnen zu lassen. Daneben gibt es in der Literatur noch zahlreiche weitere Artikeln, die aber für diese Arbeit nur geringe Bedeutung haben.

2 Anforderungen an die Software

Als erstes müssen die Anforderungen an das Programm festgelegt werden. Das Vorgehen dazu lehnt sich an [Oestereich, 1998], wobei auf Grund der eher geringen Grösse des Projekts gewisse Schritte ausgelassen werden.

2.1 Allgemeine Anforderungen

Funktionalität	Priorität
Input	
Import Bilder in min. einem Rasterformat	1
Import Bilder in verschiedenen Rasterformaten	2
Zusammenfügen einer Karte aus mehreren Bilder	4
Passpunkte wählen am Bildschirm	1
Import Koordinatenliste in vorgegebenem Format	1
Abgespeicherte Projekte öffnen	2
Output	
Fehlervisualisierung auf Bildschirm	1
Fehlervisualisierung in Rasterformat exportieren	2
Transformiertes Bild auf Bildschirm und exportieren	3
Statistische Auswertung auf Bildschirm und exportieren	1
Projekt abspeichern	2
Drucken	3
Fehlervisualisierungen	
Verzerrungsgitter	1
Fehlervektoren	1
Kreismethoden	2
Isodeformaten	3
Konturvergleich	3
Tissot'sche Indikatrix	3
Dreiecksflächen, -seiten, -überschneidungen	3

Funktionalitäten mit Priorität 1 sind die wichtigsten, diejenigen mit Priorität 4 sind am wenigsten wichtig. Die Prioritäten sind vorgängig festgelegt worden unter Berücksichtigung der zur Verfügung stehenden Zeit.

2.2 Der Benutzer

Beim Entwickeln eines Programmes, das nicht nur zu eigenen Zwecken dient, richtet sich die Software an einen bestimmten Benutzerkreis. Damit ein zukünftiger Benutzer das Programm auch einsetzen kann, muss es seinem Wissen und seinen Erwartungen entsprechen. Also muss vorgängig festgehalten werden, was ein Benutzer von dem Programm erwartet und was die Anforderungen seitens der Entwickler an ihn sind.
Es wird von den folgenden zwei Gruppen von Benutzern ausgegangen.

2.2.1 Interessierter Benutzer

z.B. Kartensammler etc.

- Dieser Benutzer ist vor allem an Fehlervisualisierungen interessiert.
- Er hat nur wenige bis keine Kenntnisse in Statistik und Ausgleichsrechnung.

- Er verfügt über Computerkenntnisse eines durchschnittlichen Office-Anwenders und muss das Programm und evtl. Erweiterungen nach Anleitung installieren können.

2.2.2 Wissenschaftlicher Benutzer

z.B. naturwissenschaftlicher Kartenhistoriker

- Dieser Benutzer ist sowohl an Fehlervisualisierungen wie auch an statistischer Auswertung interessiert.
- Er hat Kenntnisse in Statistik und Ausgleichsrechnung und will verschiedene Transformationen miteinander vergleichen können.
- Er verfügt über Computerkenntnisse eines durchschnittlichen Office-Anwenders und muss das Programm und evtl. Erweiterungen nach Anleitung installieren können.

2.2.3 Systemanforderungen an den Computer des Benutzers

- Der Benutzer hat einen modernen Computer mit beliebigem Betriebssystem (Windows, Mac, Linux). Auf Grund der grossen Dateigrössen bei eingescannten Karten soll der Rechner über einen ausreichend schnellen Prozessor und genügend Arbeitsspeicher verfügen.
- Die Java Virtual Machine ist installiert.

2.3 Anwendungsfälle

Ein Anwendungsfall beschreibt einen typischen Arbeitsablauf aus Sicht des Benutzers. Von Relevanz sind alle Eingaben des Benutzers und die daraus folgenden Resultate.
Anhand dieser Anwendungsfälle werden später die Funktionalitäten programmiert. Es bleibt zu erwähnen, dass gewisse Details von Funktionalitäten nicht oder anders realisiert worden sind und dass die Namen einiger Bedienelemente geändert worden sind.

2.3.1 Allgemeiner Ablauf

Der Benutzer öffnet ein neues oder ein bestehendes Projekt. Er erstellt die Passpunktliste, d.h. eine Liste mit zugehörigen Punkten in der alten und der modernen Karte.
Im Reiter 'Scale & Transformation' wählt der Benutzer die gewünschte Transformation und in den Reitern 'Distortion Grid' bzw. 'Error Vectors' die Einstellungen betreffend die Fehlervisualisierungen. Für jede Darstellung bestimmt er, ob sie angezeigt werden soll oder nicht.
Danach führt der Benutzer die Berechnung durch, indem er 'Compute' klickt. Die Fehlervisualisierungen werden dargestellt. Anschliessend können diese in Raster- oder Vektorformaten exportiert werden.
Sobald er weitere Passpunkte hinzufügt, werden die Visualisierungen nicht mehr angezeigt, da sich diese dann nicht mehr auf die aktuell gesetzten Passpunktpaaren beziehen.

2.3.2 Passpunktliste erstellen

Alte Karte importieren :
Invarianten: Die Auflösung der Karte (z.B. 150dpi) muss bekannt sein, damit die Blattkoordinaten in Millimeter bestimmt werden können. Das entsprechende Grafikformat übergibt diesen Parameter an das Programm.
Ablaufbeschreibung: Der Benutzer kann eine alte Karte mit dem Befehl 'Import Old Map' im Menu 'File' importieren.
Nachbedingungen: Die alte Karte wird im linken Fenster dargestellt.

Moderne Karte importieren :
Invarianten: Die neue Karte ist georeferenziert. Die Georeferenzierung erfolgt mit einem sogenannten 'World-File'.
Ablaufbeschreibung: Analog zum Import der alten Karte, importiert der Benutzer eine neue Karte mit dem Befehl 'Import Modern Map' im Menu 'File'.
Nachbedingungen: Die neue Karte wird im rechten Fenster dargestellt.

Koordinatenliste importieren :
Invarianten: Die Koordinaten sind in einem vorgegebenem Format in einer ASCII-Liste geschrieben.
Ablaufbeschreibung: Der Befehl 'Import Point List' ist im Menu 'Points'. Ein Fenster öffnet sich, wo der Benutzer die Textdatei mit den Punktkoordinaten suchen und auswählen kann.
Nachbedingung: Die importierten Punkte werden in beiden Karten dargestellt. Die Punkte sind, je nach dem ob sie verknüpft sind oder nicht, entsprechend symbolisiert.

Navigation in der Karte :
Ablaufbeschreibung: Der Benutzer navigiert und sucht den gewünschten Punkt in der Karte mit den Funktionen Verschieben und Vergrössern bzw. Verkleinern sowie mit Scrollen in horizontaler und vertikaler Richtung. Zusätzlich hat er die Möglichkeit über den Befehl 'Show All' die ganze Karte anzuzeigen.

Zoom in/out :
Ablaufbeschreibung: Der Benutzer wählt den Button 'Zoom In' bzw. 'Zoom Out'. Er klickt in eine Karte, welche vergrössert oder verkleinert und um den geklickten Punkt zentriert wird.

Verschieben :
Ablaufbeschreibung: Es wird der Button 'Pan' gewählt. Anschliessend wird in eine Karte geklickt und die Karte verschoben.

Passpunkte in beiden Karten setzen :
Ablaufbeschreibung: Der Benutzer wählt 'Set Points' und setzt in der einen Karte einen Passpunkt. Dieser Punkt wie auch das Textfeld 'LinkName' werden aktiv. In der anderen Karte sucht der Benutzer den entsprechenden Ort und setzt dort einen Punkt. Dieser wird wiederum aktiv. Während dem Setzen der Punkte werden gleichzeitig im Reiter 'Points' die Koordinaten ausgegeben.

Bis ein nächster Passpunkt in der gleichen Karte gesetzt wird, ist der zuletzt gesetzte Punkt selektiert.

Passpunkte verknüpfen :
Vorbedingungen: In beiden Karten ist der entsprechende Passpunkt aktiv.
Ablaufbeschreibung: Der Benutzer verknüpft die beiden Punkte mit dem Button 'Link'. Ist er im 'Set Points' Modus, kann er der Verknüpfung einen Namen zuweisen. Ist dies nicht der Fall, muss der Benutzer nach dem Verknüpfen diese beiden Punkte wieder selektieren und mit dem Befehl 'Rename' einen Namen zuweisen.
Nachbedingung: Die neu verknüpften Punkte sind nicht mehr aktiv und werden entsprechend symbolisiert, so dass deutlich wird, dass diese Punkte verknüpft sind.

Passpunkte selektieren :
Ablaufbeschreibung: Der Benutzer wählt 'Select Points' und kann anschliessend in einer Karte ein Rechteck aufziehen. Alle Punkte innerhalb dieses Rechtecks werden aktiv. Selektiert der Benutzer nur einen verknüpften Punkt, wird in der anderen Karte der entsprechende Punkt zentriert. Im Reiter 'Points' werden die Koordinaten dieser beiden Punkte und der Name angezeigt.

Passpunkte löschen :
Ablaufbeschreibung: Nach der Selektion der Punkte wird User gefragt, ob die Punkte in der alten, neuen, oder falls eine Verknüpfung besteht in beiden Karten gelöscht werden sollen. Wird ein Punkt gelöscht, der zu einer Verknüpfung gehört, wird diese aufgehoben.

Verknüpfung umbenennen :
Vorbedingung: Der Benutzer hat eine Verknüpfung selektiert, d.h. in der alten und der neuen Karte ist je ein Punkt selektiert.
Ablaufbeschreibung: Im Reiter 'Points' wird 'Rename' gewählt, worauf sich ein Dialogfenster öffnet, wo der neue Name eingegeben werden kann.

Verknüpfung auflösen :
Vorbedingung: Wie beim Umbenennen ist eine Verknüpfung selektiert.
Ablaufbeschreibung: Der Benutzer löst eine Verknüpfung auf mit dem Befehl 'Unlink'.
Nachbedingung: Die vorher selektierten und verknüpften Punkte werden deaktiviert und deren Symbole ändern sich.
Regeln: Der Button 'Link' übernimmt zwei Funktionen. Werden neue Passpunkte gesetzt, können mit diesem Button Punkte verknüpft werden. Ist hingegen eine Verknüpfung selektiert, so wird über diesen Button die Verknüpfung aufgelöst.

Passpunktliste anzeigen :
Ablaufbeschreibung: Im Menu 'Points' wählt der Benutzer 'Show Point List' , worauf sich ein Textfenster öffnet, in welchem die Koordinaten aller Punkte aufgeführt sind. Die Darstellung ist so gewählt, dass für den Benutzer ersichtlich ist, welche Punkte mit welchen verknüpft sind und welche keine Verknüpfung haben. Diese Punktliste kann direkt aus diesem Fenster exportiert und abgespeichert werden.

Passpunkte speichern/exportieren :

Ablaufbeschreibung: Über den Befehl 'Export Point List' kann der Benutzer alle bisher gemessenen Punkte und deren Verknüpfungen exportieren.
Variation: Anstatt die Liste direkt zu exportieren, kann sie zuerst angezeigt und erst dann gespeichert werden.

2.3.3 Verzerrungsgitter darstellen

Im Reiter 'Distortion Grid' werden die Einstellungen für das Verzerrungsgitter angegeben. Neben den Einstellungen für die Darstellung wie Strichmuster, -stärke und -farbe gibt es Optionen für die Berechnung des Netzes. Der Benutzer kann die Gitterweite wählen und bestimmen, ob sich das Gitter auf kartesische oder geografische Koordinaten bezieht.
Das Verzerrungsgitter wird angezeigt, sobald die nächste Berechnung durchgeführt wird, nachdem das Häkchen 'visible ' gesetzt worden ist.

2.3.4 Fehlervektoren darstellen

Der Benutzer kann im Reiter 'Error Vectors' Einstellungen vornehmen betreffend die Darstellung der Vektoren. Er kann die Strichart, -stärke und -farbe sowie die Skalierung der Vektoren wählen.
Um die Fehlervektoren darzustellen setzt er das Häkchen 'visible' und führt anschliessend eine Berechnung durch.

3 Mathematische Grundlagen

3.1 Transformationen

Als wichtigste Transformationen werden vorerst die Helmert Transformation mit vier Parametern und zweimal die affine Transformation einmal mit fünf und einmal mit sechs Parametern implementiert.
Im Verlaufe der Arbeit wird noch die robuste Helmert Transformation [Carosio, 1996] programmiert, bei der zwischen dem Hampel-, Huber- und V-Schätzer [Beineke, 2001] ausgewählt werden kann.

3.2 Interpolation des Verzerrungsgitters

In der bisherigen Forschung sind verschiedene Methoden zur Interpolation des Verzerrungsgitters vorgeschlagen worden. Nachfolgend sind die wichtigsten mit ihren Vor- und Nachteilen dargestellt.

3.2.1 Multiquadratische Interpolation [Beineke, 2001]

Vorteile
Keine getrennte Berechnung der Verzerrungslinien in den Hauptkoordinaten-Richtungen. Geschlossene Verzerrungslinien können nicht auftreten, daher gut geeignet zu Präsentationszwecken.

Nachteile
Die Multiquadratische Interpolation verlangt eine hohe Rechenleistung, da mit grossen Matrizen gerechnet wird.

3.2.2 Lineare Interpolation auf den Dreiecksseiten [Beineke, 2001]

Vorteile
Topologische Unstimmigkeiten können aufgedeckt werden.
Nachteile
Geschlossene Linienzüge können auftreten.
Getrennte, unabhängige Berechnung der Verzerrungslinien in den Richtungen der Hauptkoordinaten.
Nur benachbarte Punkte fliessen in die Rechnung ein.

3.2.3 Interpolation $1/s^3$ [Weis, 1985]

s ist die Distanz zwischen zwei Stützpunkten.
Vorteile
Sämtliche Punkte werden in die Berechnung der Stützpunkte einbezogen.
Anstelle von s^3 kann auch s^2 oder s^4 verwendet werden um den Einfluss der Distanz zu verändern.

3.2.4 Geometrische Konstruktion [Weis, 1985]

Vorteile
Einfache Methode, gut vorstellbar, was gerechnet wird.
Nachteile
Manuelle Methode, die auf einen Computer übertragen wurde.
Für die Berechnung der Verzerrungslinie werden nur benachbarte Punktepaare berücksichtigt.
Die Resultate sind unbefriedigend.

3.2.5 Wahl des Verfahrens

Nach dem Beurteilen dieser Verfahren ist der Entscheid auf die multiquadratische Interpolation gefallen. Diese Interpolation ist ein modernes Verfahren, das nebst in [Beineke, 2001] auch in [Bretterbauer, 2005] dargestellt und empfohlen wird. Sollte es zu Problemen mit zu langen Rechenzeiten kommen, kann gegebenfalls der Algorithmus so angepasst werden, dass zu weit entfernte Punkte nicht mehr berücksichtigt werden.

3.3 Massstab

Der Massstab der alten Karte kann unterschiedlich bestimmt werden. In [Beineke, 2001] sind verschiedene Verfahren aufgeführt.
In der vorliegenden Arbeit wird ein sogenanntes Vergleichspunkt-Verfahren verwendet, d.h. der Massstab wird direkt aus der Transformation berechnet. Da für eine Fehlerdarstellung

immer eine Transformation gerechnet werden muss, wäre eine andere Massstabsbestimmung überflüssig und nicht zuverlässiger.

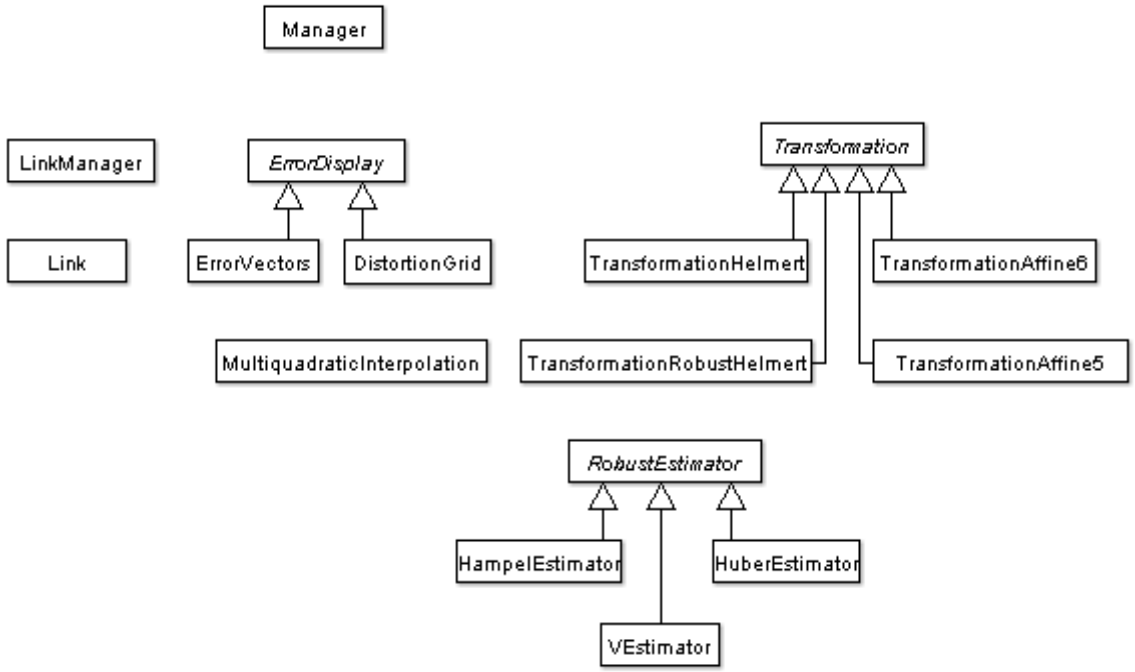


Abbildung 1: Das Fachklassenmodell des Programms

4 Entwickeln der Software

Vor dem Beginn mit Programmieren wird ein sogenanntes Fachklassenmodell erstellt. Dies geschieht mit dem frei erhältlichen Programm ArgoUML, welches zahlreiche Werkzeuge zur Modellierung zur Verfügung stellt. Ist das Fachklassenmodell fertig, können mit ArgoUML direkt die Klassen und deren Methoden generiert werden.
Der inhaltliche Ablauf beginnt mit der Identifizierung aller Klassen. Es wird festgestellt welche Klassen es gibt und welche Klassen von welchen abgeleitet sind.
Wie bereits erwähnt soll das zu entwickelnde Programm frei zugänglich und jederzeit erweiterbar sein. Darauf wird bei der Modellierung speziell geachtet. So gibt es die abstrakte Klasse 'ErrorDisplay' von der verschiedene Fehlerdarstellungen abgeleitet werden. Vorläufig gibt es die beiden Klassen 'ErrorVectors' und 'DistortionGrid', zu der auch die Klasse 'MultiquadraticInterpolation' gehört. Werden später weitere Visualisierungen programmiert, werden die ebenfalls von 'ErrorDisplay' abgeleitet. Dasselbe gilt für die abstrakte Klasse 'Transformation' und deren abgeleiteten Klassen. So wird erreicht, dass das Programm jederzeit modulartig erweitert werden kann.
Die Klasse 'Manager' verwaltet inhaltlich die verschiedenen Module und stellt die Verbindung zur Benutzungsoberfläche her. Der 'LinkManager' verwaltet die erstellten oder

importierten Verknüpfungen zwischen den Passpunkten in der alten und neuen Karte. Diese Verknüpfungen werden als Instanzen der Klasse 'Links' realisiert.

4.1 Gewählte Sprache und Entwicklungsumgebung

Bereits in der Ausschreibung der Arbeit ist vorgeschlagen entweder in C oder in Java zu programmieren. Mehrere Gründe führen schliesslich zum Entscheid Java zu verwenden. Neben den persöhnlichen Präferenzen des Autors spricht auch die vereinfachte Realisierung der Plattformunabhängigkeit für Java.
Als Entwicklungsumgebung wird auf Empfehlen des Betreuers 'NetBeans IDE' gewählt. Diese ist von dem Unternehmen 'Sun Microsystem', ist frei erhältlich und verfügt über sämtliche Funktionalitäten wie Debugger etc. einer kostenpflichtigen Entwicklungsumgebung.

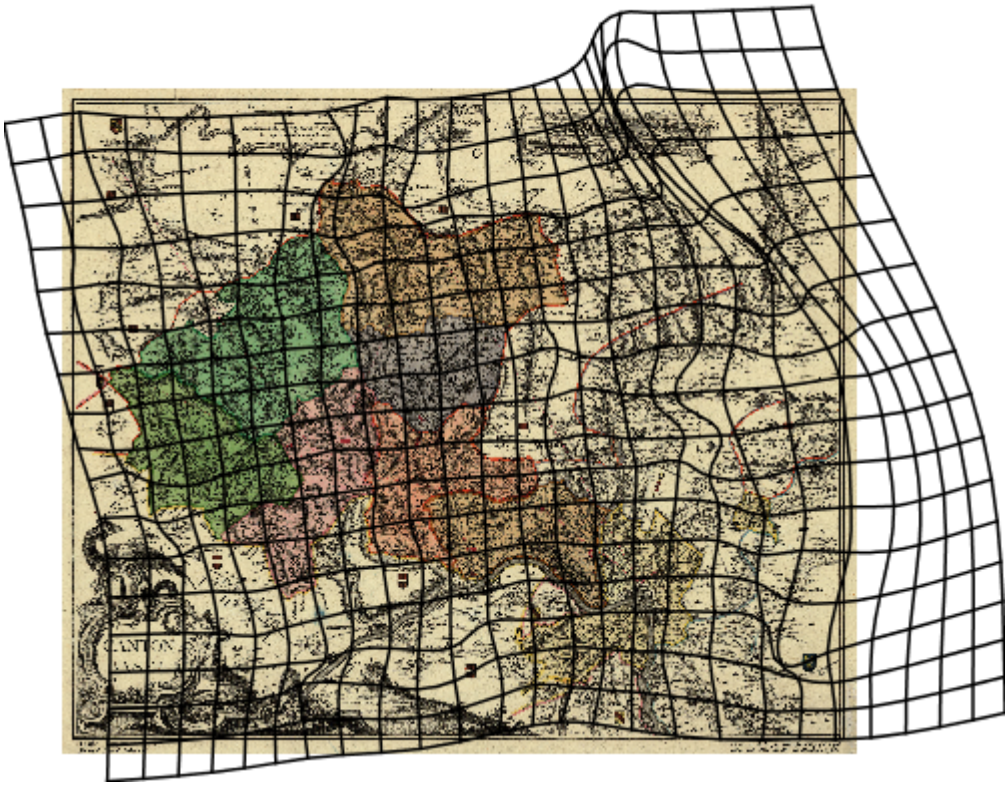


Abbildung 2: Fehlerdarstellung: Verzerrungsgitter mit MapAnalyst berechnet und dargestellt.
Grundlagenkarte und Passpunkte sind freundlicherweise von Martin Rickenbacher, Swisstopo, zur Verfügung gestellt worden.

5 Fehlerdarstellungen

5.1 Verzerrungsgitter

Die Darstellung der Verzerrungen in einer alten Karte anhand von Gitter ist eine anschauliche Methode. Darum ist diese Methode in der Wissenschaft seit längerer Zeit sehr beliebt [Forstner und Oehrli, 1998].

5.2 Fehlervektoren

Die Fehlervektoren in einer alten Karte verbinden die Passpunkte in der alten Karte mit den heutigen, als fehlerfrei angenommenen Punkten, die in das Blattkoordinatensystem der alten Karte transformiert worden sind [Forstner und Oehrli, 1998].

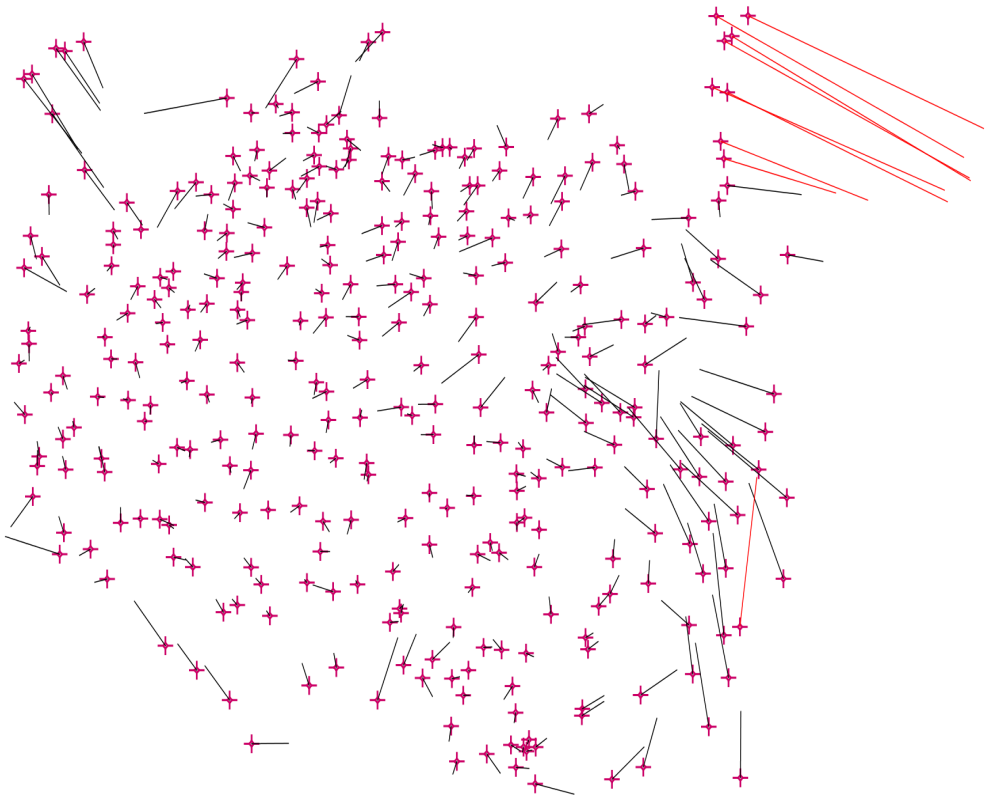


Abbildung 3: Fehlerdarstellung: Fehlervektoren mit MapAnalyst berechnet und dargestellt. Zur verbesserten Anschaulichkeit ist die Hintergrundkarte ausgeblendet. Vektoren mit einem grösseren Betrag als dreimal die Standardabweichung $\sigma 0$ sind rot eingefärbt. Grundlagenkarte und Passpunkte sind freundlicherweise von Martin Rickenbacher, Swisstopo, zur Verfügung gestellt worden.

5.3 Kreismethode [Mekenkamp, 1990]

Bei der Kreismethode werden die Summen der relativen Distanzfehler aller Punkte in Bezug auf den betrachteten Punkt als Kreise unterschiedlichen Durchmessers um die jeweilige Ortsposition gezeichnet [Forstner und Oehrli, 1998].

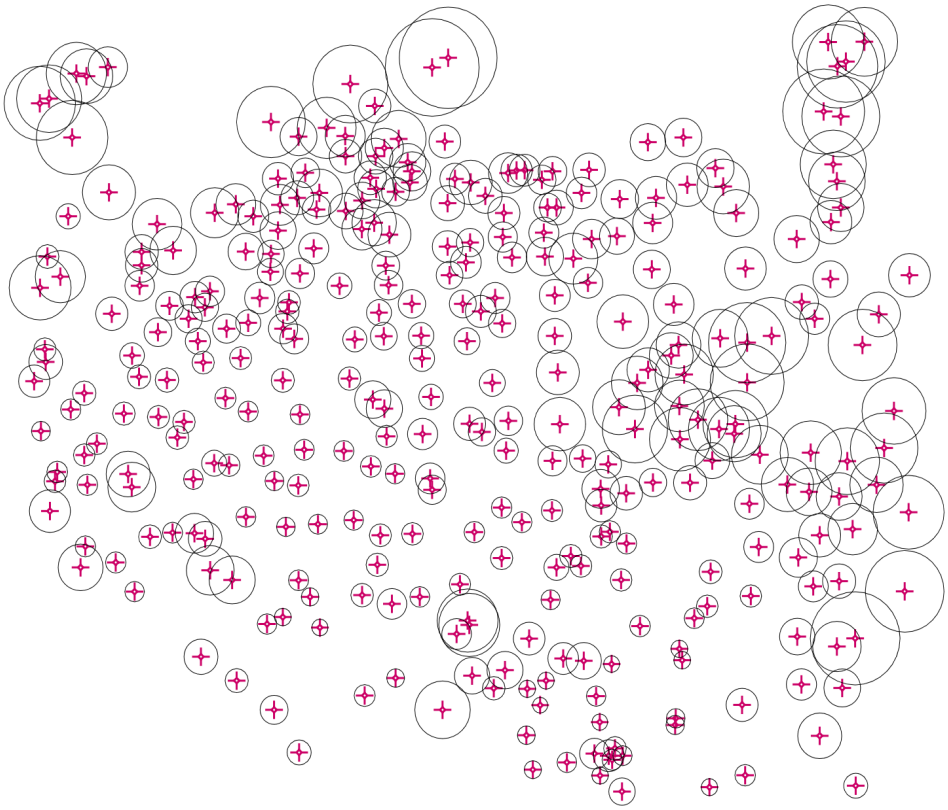


Abbildung 4: Fehlerdarstellung: Kreismethode mit MapAnalyst berechnet und dargestellt. Zur verbesserten Anschaulichkeit ist die Hintergrundkarte ausgeblendet.
Grundlagenkarte und Passpunkte sind freundlicherweise von Martin Rickenbacher, Swisstopo, zur Verfügung gestellt worden.

6 Nachfolgearbeiten

Bereits zu Beginn der Arbeit wird deutlich, dass zahlreiche Funktionalitäten aus Zeitgründen nicht realisiert werden können. Selbst später wird noch auf Funktionen verzichtet, die Anfangs programmiert werden sollten. Nachfolgend sind Erweiterungen aufgelistet die in einer nachfolgenden Arbeit realisiert werden können.

6.1 Bekannte Programmfehler (Bugs)

Fehlerhafte 'Smoothness' Funktion Dieser Fehler betrifft nur Benutzer mit einem 'Windows' Betriebssystem. Die Glättung der Verzerrungsgitterlinien, die mit dem Befehl 'Smoothness' verändert werden kann, funktioniert in der alten Karte für kurze Linienabschnitte nicht. Die Darstellung von Fehlerkreisen, die unter einer bestimmten Grösse sind, funktioniert in der alten Karte aus gleichen Gründen nicht.
Davon *nicht betroffen* sind die exportierten Graphiken. D.h. fehlerhaft dargestellte Visualisierungen werden korrekt exportiert.

6.2 Erweiterungsmöglichkeiten und Verbesserungen

Weitere Darstellungsarten Wie bereits in der Einleitung erwähnt, gibt es in der Forschung verschiedenste Arten die Fehler alter Karten darzustellen. Die unterschiedlichen Visualisierungen lassen unterschiedliche Aussagen zu. Daher ist es sinnvoll weitere Darstellungen im Programm zu implementieren. Einige mögliche Darstellungen sind in [Forstner und Oehrli, 1998] zu finden.

Weitere und verbesserte Transformationen Weitere Transformationen können hinzugefügt werden.
Von den bestehenden Transformationen sollen insbesondere die robusten Transformationen erweitert werden. Neben der robusten Helmert Transformation sollen auch die affinen Transformationen mit einem robusten Schätzer berechnet werden können. Weiter gilt es wenn möglich die Berechnungszeit der robusten Ausgleichung zu verkürzen und dem Benutzer die Möglichkeit zu geben, gewisse Einstellungen vorzunehmen.
Eine weitere hilfreiche Funktionalität ist die Anzeige der Transformationseigenschaften in einem Informationsfenster. Zudem soll es eine Vergleichsmöglichkeit der verschiedenen Transformationen zur Unterstützung des Benutzers bei deren Wahl geben.

Geographische Koordinatensysteme Es soll möglich sein mit geographischen und nicht nur mit kartesischen Systemen rechnen zu können. Das Verzerrungsgitter in der Referenzkarte wird dementsprechend nicht mehr ein rechtwinkliges sein.

Projektionssystem Bevor eine Transformation gerechnet wird, sollen die Passpunkte auf geographische Koordinaten umgerechnet werden. Dazu werden vordefinierte Projektionssysteme, oder die Möglichkeit ein solches zu definieren, benötigt. Der Einfluss der Projektion fliesst so nicht mehr in die Transformation.
Dies bedingt oben genannter Punkt der geographischen Koordinatensystemen.

Wizard Ein Wizard ist ein Assistent, der den Benutzer durch die verschiedenen Arbeitsschritte führt. Wird der Wizard gestartet, wird der Benutzer nach den Karten wie auch nach den Punkten, die importiert werden sollen, gefragt. Anschliessend wählt der Benutzer im Wizard die Transformation und die Darstellungen. Schliesst er den Wizard, erscheint direkt die berechneten Fehlerdarstellungen.
Ein Wizard ist vor allem für ungeübte Benutzer eine hilfreiche Unterstützung.

7 Bezugsquelle

Die Applikation wie auch der Quellcode ist frei erhältlich auf www.ika.ethz.ch/mapanalyst. Dort wird die neuste Version mit deren Änderungen und Neuerungen veröffentlicht.

8 Danksagung

An dieser Stelle möchte ich allen danken, die zum guten Gelingen dieser Arbeit beigetragen haben. Als erstes geht mein Dank an Bernhard Jenny, der mich mit tatkräftig unterstützt hat. Er hat einen grossen Teil der Applikation programmiert und zahlreiche Fehler, sogennante Bugs, korrigiert.
Weiter richtet sich meinen Dank an Martin Rickenbacher für die zur Verfügung gestellten Testdaten und die hilfreichen Anregungen, an Dieter Beineke, der ebenfalls Testdaten zur Verfügung gestellt hat, an Markus Oehrli und Jana Niederöst für die Anregungen und die Vorschläge von Verbesserungsmöglichkeiten.

Literatur

[Beineke, 2001] Beineke, D. (2001): Verfahren zur Genauigkeitsanalyse für Altkarten. Universität der Bundeswehr München.

[Bretterbauer, 2005] Bretterbauer, K. (2005): Zur Konstruktion von Verzerrungslinien mittels der multiquadratischen Methode von Hardy. Kartographische Nachrichten, Heft 2, April 2005. 83-85.

[Carosio, 1996] Carosio, A. (1996): Les méthodes de la statistique robuste au service de la compensation de réseaux géodediques. ETH Zürich, Institut für Geodäsie und Photogrammetrie. Bericht 257 f.

[Forstner und Oehrli, 1998] Forstner, G. und Oehrli, M. (1998): Graphische Darstellungen der Untersuchungsergebnisse alter Karten und die Entwicklung der Verzerrungsgitter. Cartographica Helvetica, Heft 17, Januar 1998. 35-43.

[Mekenkamp, 1990] Mekenkamp, P. (1990): Die Entwicklung einer neuen Methode für die Bestimmung der Genauigkeit von alten Karten. 5. Kartographiehistorisches Colloquium Oldenburg 1990, Vorträge und Berichte. 111-116.

[Oestereich, 1998] Oestereich, B. (1998): Ojektorientierte Softwareentwicklung: Analyse und Design mit der Unified Modeling Language.

[Weis, 1985] Weis, I. (1985): Automatische Erstellung von Verzerrungsgitter alter Karten. Technische Universität München.

Adrian Weber