

# **Free GIS Tutorial**

## **Einsatz Freier GIS-Software in Vermessung, Fernerkundung und Datenverarbeitung**

Heiko Kehlenbrink\*  
Intevation GmbH Osnabrück†  
Version 0.3

30. November 2001

\*Heiko.Kehlenbrink@intevation.de

†www.intevation.de



Das Free GIS Tutorial ist keine Originaldokumentation zu der behandelten Software.

Die in diesem Buch enthaltenen Angaben, Daten, Ergebnisse usw. wurden von den Autoren nach bestem Wissen erstellt und mit Sorgfalt überprüft. Dennoch sind inhaltliche Fehler nicht völlig auszuschließen. Daher erfolgen alle Angaben ohne jegliche Verpflichtung oder Garantie. Autor und Herausgeber übernimmt daher auch keinerlei Verantwortung oder Haftung für Fehler und deren Folgen. Hinweise auf eventuelle Irrtümer werden gerne entgegengenommen.

Copyright (c) 2001 Heiko Kehlenbrink.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation. A copy of the license is included in the section entitled "GNU Free Documentation License".





# Vorwort

Das FreeGIS Tutorial entstand als ein weiterführender Schritt innerhalb des FreeGIS Projects und baut somit direkt auf das FreeGIS Portal <sup>1</sup> und die FreeGIS CD <sup>2</sup> auf. Die Intention ist, dem Nutzer von GIS-Software, mit der beispielhaften Bearbeitung von Aufgaben aus GIS, Vermessung und Fernerkundung die Vielfältigkeit freier Software in diesen Bereichen nahezubringen. Grundvoraussetzung dafür ein Programm in das FreeGIS Projekt und somit auch in dieses Tutorial aufzunehmen zu können ist, daß es freie Software im Sinne der Free Software Foundation (FSF)<sup>3</sup> ist.

Jede Aufgabenstellung soll mit verschiedenen Programmen gelöst werden, um dem Nutzer Alternativen zu bieten, aus denen er sich das für Ihn passende herauszusuchen kann.

Bei der Erstellung dieses Tutorials wurde darauf Wert gelegt, es nicht auf eine Systemumgebung z.B. GNU/Linux zu beschränken, sondern zu zeigen, daß freie Software auf allen Betriebssystemen existiert und sich produktiv im Zusammenwirken mit vielen anderen, sogar nicht-freien (proprietären) Anwendungen einsetzen lässt.

Das FreeGIS Tutorial wird unter der GNU Free Document License herausgegeben um den Nutzern die Möglichkeit zu geben, es genau wie die Freie Software die es behandelt unter den Bedingungen der vorgenannten Lizenz zu verbreiten, zu verwenden und zu verändern. Auch sind sämtliche in diesem Tutorial bearbeiteten Daten Freie Daten im Sinne der Free Software Foundation und können genau wie die Programme und dieses Tutorial verbreitet, verwendet, verändert, be- und verarbeitet werden. Die einzige Ausnahme sind die EDDBS-Datensätze im Kapitel 16 -EDDBS Daten verarbeiten-.

Um das FreeGIS Tutorial aktuell und für so viele Menschen wie möglich interessant und informativ zu machen wird jede Mitarbeit ausdrücklich begrüßt. Eine mögliche Mitarbeit reicht von der Fehlerkorrektur (sowohl Inhaltlich wie auch Rechtschreibung), über das Einbringen von neuen Aufgaben und die Erstellung von Glossar-Einträge bis hin zum Erarbeiten von ganzen Kapiteln. Überaus willkommen sind weiterhin Übersetzungen von Tutorial - Teilen, oder auch das direkte Erstellen von Kapiteln in den verschiedensten Sprachen.

**zu Version 0.2** Die Version 0.2 wurde aufgrund der Kommentierungen und Kritiken die auf die Version 0.1 (diese Version wurde an einen kleinen Kreis von Fachleuten verteilt) eintrafen überarbeitet und erweitert. Beispielsweise wurde die Dateistruktur so verändert das jetzt jedes Kapitel in einem eigenen Verzeichniss steht. Dies im Verbund mit einer Ände-

---

<sup>1</sup>[www.freegis.org](http://www.freegis.org)

<sup>2</sup><http://www.freegis.org/cd.de.html>

<sup>3</sup>[www.fsf.org/philosophy/categories.de.html](http://www.fsf.org/philosophy/categories.de.html)

rung der Dateibenennung sollte eine gewünschte Übersetzung / Lokalisierung vereinfachen. Die so entstandene Version 0.2 wird nun zur allgemeinen Kommentierung und Kritik im Internet veröffentlicht. Die zugrundeliegende Konzeption erscheint ausreichend gefestigt, so daß schon in diesem Stadium der Beitrag eines weiteren Autoren eingebaut wurden.

Heiko Kehlenbrink

# Dank

Hier möchte den vielen Menschen, Institutionen, Firmen und Gemeinschaften Danken, ohne deren Hilfe dieses Tutorial nicht entstanden wäre. Sollte ich jemanden nicht erwähnt haben, so möge er mir dies verzeihen.

## **Besonders erwähnen möchte ich hier :**

Das Team und die Geschäftsführung der Intevation Gmbh in Osnabrück bei dem ich im Rahmen meines Praxissemesters dieses Tutorial erstellen konnte.

## **Für die Bereitstellung von Daten und Geräten danke ich:**

Der RWE Rheinbraun AG, Abteilung BT3 - Photogrammetrie

Herrn Professor Dr.-Ing. Jörg Reinking, Fachhochschule Oldenburg/Ostfriesland/Wilhelmshaven

Herrn Dipl.-Ing. Peter Lorkowski, Fachhochschule Oldenburg/Ostfriesland/Wilhelmshaven

## **Für das Erstellen von Kapiteln**

Herrn Otto Dassau für das Kapitel Nr. 16 , EDBS Daten verarbeiten , Seiten 153 ff

## **Für die Mitarbeit am Glossar**

## **Für die Erarbeitung von Aufgabenstellungen**

## **Für Korrekturen, Anmerkungen. Kritiken...**

Herrn Professor Dr. Jürgen Weitkämper

Herrn Dipl.-Ing. Peter Lorkowski

Herrn Hon.-Prof. Klaus Kertscher

Frau Melanie Wagner



# Inhaltsverzeichnis

<b>1. Einführung</b>	<b>15</b>
1.1. Rasterdaten . . . . .	15
1.2. Vektordaten . . . . .	15
1.3. Sachdaten . . . . .	15
1.4. Hybride Graphik . . . . .	15
1.5. Datenformate . . . . .	15
1.6. Übersicht der bearbeiteten Aufgaben . . . . .	16
1.7. Übersicht der benutzten Daten . . . . .	17
<b>2. Übersicht zu den verwendeten Programmen</b>	<b>19</b>
2.1. g3DGMV . . . . .	19
2.2. GRASS . . . . .	19
2.3. VTP Enviro . . . . .	19
2.4. VTBUILDER . . . . .	20
2.5. DLG Viewer / dlgv32 . . . . .	20
2.6. dem3D . . . . .	21
2.7. GPS Point . . . . .	21
2.8. GPSMan . . . . .	21
2.9. DPSIP . . . . .	22
2.10. Mapit! . . . . .	22
2.11. MapServer . . . . .	22
<b>3. Freie Geo-Daten</b>	<b>25</b>
3.1. Bedeutung von Freiheit bei Geo-Daten . . . . .	25
3.2. Übersicht zu verfügbaren freien Geo-Daten . . . . .	25
3.3. Bewertung von Daten . . . . .	25
<b>4. Punktdaten einlesen und digitales Höhenmodell erzeugen</b>	<b>27</b>
4.1. Lösung mit GRASS . . . . .	27
4.1.1. Vorbereitungen . . . . .	28
4.1.2. Anmerkungen zum Arbeiten mit GRASS . . . . .	34
4.1.3. Öffnen eines Bearbeitungsfensters . . . . .	34
4.1.4. Importieren von Punktdaten . . . . .	35
4.1.5. Anzeigen der Punktdaten . . . . .	37
4.1.6. Maskierung des befahrenen Hafengebietes . . . . .	37

4.1.7.	Punktdateien in Rasterdateien umwandeln . . . . .	44
4.1.8.	Rasterdateien anzeigen . . . . .	46
4.1.9.	3 - dimensionale Ansicht des Höhenmodells . . . . .	47
<b>5.</b>	<b>Triangulation in Punktdateien</b>	<b>51</b>
5.1.	Lösung mit GRASS . . . . .	51
<b>6.</b>	<b>Digitales Höhenmodell des United States Geological Survey (USGS) importieren.</b>	<b>55</b>
6.1.	Lösung mit g3DGMV (3D Graphical Map Viewer) . . . . .	56
6.2.	Lösung mit VTBUILDER . . . . .	56
6.3.	Lösung mit dem3D . . . . .	58
6.4.	Lösung mit Grass . . . . .	59
6.4.1.	Einlesen des DHM . . . . .	60
6.5.	Lösung mit dlgy32 / DLG Viewer . . . . .	61
<b>7.</b>	<b>Höhenlinien aus Punkttraster erzeugen</b>	<b>63</b>
7.1.	Lösung mit GRASS . . . . .	63
7.1.1.	Erzeugen der Höhenlinien . . . . .	63
7.1.2.	Anzeigen der Höhenlinien . . . . .	64
<b>8.</b>	<b>Überflutungssimulation im Höhenmodell erzeugen</b>	<b>67</b>
8.1.	Lösung mit GRASS . . . . .	67
8.2.	Lösung mit dem3D . . . . .	68
8.3.	Lösung mit dlgy32/ DLG Viewer . . . . .	68
<b>9.</b>	<b>Rasterdateien georeferenzieren</b>	<b>71</b>
9.1.	Lösung mit GRASS . . . . .	71
9.1.1.	Rasterdatei einlesen . . . . .	72
9.1.2.	Georeferenzierung einer unverzerrten ausgerichteten Rasterdatei . . . . .	73
9.1.3.	Georeferenzierung einer unverzerrten nicht ausgerichteten Datei . . . . .	75
9.1.4.	Georeferenzierung einer verzerrten unausgerichteten Datei . . . . .	79
<b>10.</b>	<b>Volumenberechnung aus Rasterdateien</b>	<b>83</b>
10.1.	Lösung mit GRASS . . . . .	83
10.1.1.	Maskierung der Wasseroberfläche . . . . .	83
10.1.2.	Digitalisierung am Bildschirm . . . . .	84
10.1.3.	Erzeugung einer Rastermaske aus der Vektordatei . . . . .	91
<b>11.</b>	<b>Mosaik aus Luftbildern erstellen</b>	<b>97</b>
11.1.	Lösung mit Grass . . . . .	97
11.2.	Vorbereitungen . . . . .	98
11.3.	Transformationsparameter bestimmen . . . . .	100
11.4.	Transformieren des Luftbildes . . . . .	103
11.5.	Abschließendes Vorgehen . . . . .	106

11.6. Alternative Vorgehensweisen . . . . .	106
<b>12. Verarbeiten von GPS - Daten</b>	<b>107</b>
12.1. Programm gpspoint . . . . .	107
12.1.1. Verwalten von GPS-Daten . . . . .	109
12.1.2. Ausgabe der aktuellen Position . . . . .	111
12.2. Programm GPSMan . . . . .	112
12.2.1. Verwalten von Daten . . . . .	113
12.2.2. Anzeigen und Speichern von Echtzeitdaten . . . . .	113
12.2.3. Erstellen von Routen . . . . .	114
<b>13. Differentielle GPS - Messung</b>	<b>117</b>
13.1. Lösung mit DGPSIP . . . . .	117
13.1.1. Durchführung der Messung . . . . .	118
<b>14. Flug über das Höhenmodell</b>	<b>121</b>
14.1. Lösung mit VTP Enviro . . . . .	121
14.2. Lösung mit GRASS . . . . .	122
14.2.1. Zusammenfassen der Flugeinzelbilder zu einem Film . . . . .	124
<b>15. Dynamische Karten im WWW (Web Mapping)</b>	<b>127</b>
15.1. Lösung mit MapIt! . . . . .	127
15.1.1. Vorbereitung . . . . .	128
15.1.2. Struktur des Beispieldatensatzes . . . . .	129
15.1.3. Bildkacheln selbst erzeugen . . . . .	133
15.1.4. Navigation im Browser . . . . .	135
15.2. Lösung mit MapServer . . . . .	135
15.2.1. Vorbereitung . . . . .	136
15.2.2. Erstellung der MapServer - Anwendung . . . . .	138
15.2.3. Einbinden der erzeugten Höhenlinien aus Kapitel 7 Seite 63 . . . . .	146
15.2.4. Erweiterung der Anwendung . . . . .	148
<b>16. EDBS Daten verarbeiten</b>	<b>153</b>
16.1. Lösung mit GRASS . . . . .	153
16.1.1. Das EDBS-Format . . . . .	153
16.1.2. Der EDBS_extra Reader . . . . .	154
16.1.3. Import von EDBS Daten nach GRASS . . . . .	154
16.1.4. Ergebnis . . . . .	158
<b>17. Daten konvertieren</b>	<b>161</b>
17.1. Höhenmodelle . . . . .	161
<b>A. Ausblicke</b>	<b>163</b>

<b>B. GNU Free Documentation License</b>	<b>165</b>
B.1. Applicability and Definitions . . . . .	165
B.2. Verbatim Copying . . . . .	166
B.3. Copying in Quantity . . . . .	167
B.4. Modifications . . . . .	167
B.5. Combining Documents . . . . .	169
B.6. Collections of Documents . . . . .	169
B.7. Aggregation With Independent Works . . . . .	170
B.8. Translation . . . . .	170
B.9. Termination . . . . .	170
B.10. Future Revisions of This License . . . . .	170



# Abbildungsverzeichnis

4.1. Screenshot GRASS Punktdaten Elsfleth . . . . .	37
4.2. Screenshot GRASS Hüllenvektor Elsfleth . . . . .	42
4.3. Screenshot GRASS interpoliertes Raster Elsfleth 1 . . . . .	45
4.4. Screenshot GRASS interpoliertes Raster Elsfleth 2 . . . . .	47
4.5. Screenshot GRASS Modul nviz2.2 Gui . . . . .	49
4.6. Screenshot GRASS Modul nviz2.2 Gittermodell . . . . .	49
4.7. Screenshot GRASS Modul nviz2.2 3D-Modell . . . . .	50
5.1. Screenshot GRASS Delaunay Triangulation . . . . .	53
5.2. Screenshot GRASS Delaunay Triangulation 3D . . . . .	54
6.1. Screenshot g3DGMV . . . . .	57
6.2. Screenshot VTBUILDER1 . . . . .	57
6.3. Screenshot VTBUILDER2 . . . . .	58
6.4. Screenshot dem3D Craterlake 1 . . . . .	58
6.5. Screenshot dem3D Craterlake 2 . . . . .	59
6.6. GRASS-Ausgabemonitor USGS-DEM . . . . .	61
6.7. Screenshot dlgv32 . . . . .	62
7.1. Screenshot GRASS - Höhenlinien . . . . .	65
8.1. Screenshot GRASS - Überflutung1 . . . . .	68
8.2. Screenshot dem3D - Überflutung . . . . .	69
8.3. Screenshot dlgv32 - Überflutung1 . . . . .	69
9.1. Screenshot GRASS Georeferenzierung 1 . . . . .	79
9.2. Screenshot GRASS Georeferenzierung 2 . . . . .	80
9.3. Screenshot GRASS Craterlake Topographie 3D . . . . .	80
9.4. Screenshot GRASS Craterlake Luftbild 3D . . . . .	81
10.1. Screenshot GRASS Craterlake Maske < 1883 m . . . . .	84
10.2. Screenshot GRASS Craterlake Maskendigitalisierung 1 . . . . .	89
10.3. Screenshot GRASS Craterlake Maskendigitalisierung 2 . . . . .	90
10.4. Screenshot GRASS Craterlake Maskendigitalisierung 3 . . . . .	91
10.5. Screenshot GRASS Craterlake Maskendigitalisierung 4 . . . . .	92
10.6. Screenshot GRASS Craterlake Wasseroberfläche . . . . .	94

11.1. Screenshot GRASS Passpunkte identifizieren . . . . .	103
11.2. Screenshot GRASS Mosaik . . . . .	105
11.3. Screenshot GRASS Mosaik komplett . . . . .	105
12.1. Screenshot GPSMan Empfänger . . . . .	113
12.2. Screenshot GPSMan Listen . . . . .	114
12.3. Screenshot GPSMan Hauptfenster . . . . .	115
12.4. Screenshot GPSMan Echtzeitdatenerfassung Garmin . . . . .	115
12.5. Screenshot GPSMan Echtzeitdatenerfassung NMEA . . . . .	115
14.1. Screenshot VTP Enviro Auswahl Anzeigeparameter . . . . .	122
14.2. Screenshot VTP Enviro Erdansicht . . . . .	123
14.3. Screenshot VTP Enviro Flugansicht . . . . .	124
15.1. Screenshot MapIt! . . . . .	128
15.2. Screenshot MapIt! Staus unter Windows . . . . .	130
15.3. Screenshot MapIt! . . . . .	135
15.4. Screenshot der Testausgabe . . . . .	138
15.5. Screenshot Mapserver Aufruf . . . . .	140
15.6. Screenshot der demo.html Datei . . . . .	143
15.7. Screenshot Mapserver Höhenlinien . . . . .	147
15.8. Screenshot der Mapserver Query Road . . . . .	151
16.1. Screenshot GRASS EDBS Vektorkarte . . . . .	160
16.2. Screenshot GRASS EDBS Rasterkarte . . . . .	160

# 1. Einführung

## 1.1. Rasterdaten

Nach [1] beziehen sich Rasterdarstellungen direkt auf Flächen. Das geometrische Grundelement ist das Pixel (Picture Element), welches zeilen- oder spaltenweise in einer Matrix gleichförmiger quadratischer, rechteckiger oder auch hexagonaler Elemente angeordnet ist und eine einheitliche Flächenfüllung aufweist. Beispiele für Rasterhaft angeordnete Daten sind Bitmaps , Höhenraster (Kotenpläne) aber auch ein Fernseh- bzw. Monitorbild.

## 1.2. Vektordaten

Vektordaten entstehen aus Punkten, Verbindungen zwischen Punkten (Linien) und geschlossenen Linienpolygonen(Flächen).

Weiter können Angaben über Anfangs- und Endpunkt einer Linie oder über nachbarschaftliche (topologische) Beziehungen zwischen Linien und Flächen bzw. sich daraus ergebend Flächen und Flächen gemacht werden.

## 1.3. Sachdaten

Alle weiteren im GIS-Systemen benutzten Daten können als Sachdaten, thematische Daten oder Attribute aufgefasst werden.[1]

## 1.4. Hybride Graphik

Werden Vektor- und Rasterdaten gemeinsam und abhängig voneinander bearbeitet, so spricht man von hybrider Graphik. Die Fähigkeit hybride Graphiken erzeugen und bearbeiten zu können, erschließt GIS-Systemen einen vielfältigen Anwendungsbereich.

## 1.5. Datenformate

Die Menge von Datenformaten alleine für den Gis-Bereich ist sehr groß und auf alle einzugehen würde den Rahmen dieses Tutoriums sprengen. Die benutzen Dateitypen werden im Text oder im Glossar angesprochen.

## 1.6. Übersicht der bearbeiteten Aufgaben

Aufgabe	Programme	ab Seite
Punktdateien einlesen und digitales Höhenmodell erzeugen	GRASS	27
Triangulation in den Punktdateien	GRASS	51
Höhenlinien aus einem Punktraster erzeugen	GRASS	63
Digitales Höhenmodell des United States Geological Survey (USGS) importieren	g3DGMV, VTBUILDER, dem3D, GRASS, dlgv32	55
Überflutungssimulation im Höhenmodell erzeugen	GRASS, dem3D, dlgv32	67
Georeferenzierung einer Rasterdatei	GRASS	71
Volumenberechnung aus Rasterdaten	GRASS	83
Bildschirmdigitalisierung	GRASS	84
Mosaik aus Luftbildern erstellen	GRASS	97
Verarbeiten von GPS - Daten	gpspoint, GPSMan	107
Differentielle GPS - Messung	DGPSIP	117
Flug über das Höhenmodell	VTP Enviro, GRASS	121
Karten im Netz präsentieren -Web Mapping-	Mapit!, MAPSERVER	127
EDBS Daten verarbeiten	GRASS	153
Daten konvertieren		161

## 1.7. Übersicht der benutzten Daten

Aufgabe	Daten verfügbar unter:
Punktdaten einlesen und digitales Höhenmodell erzeugen	<a href="http://freegis.org/freegis_tutorial/download.de.html">http://freegis.org/freegis_tutorial/download.de.html</a> FreeGIS Tutorial Begleit-CD
Triangulation in den Punktdaten	<a href="http://freegis.org/freegis_tutorial/download.de.html">http://freegis.org/freegis_tutorial/download.de.html</a> FreeGIS Tutorial Begleit-CD
Digitales Höhenmodell des United States Geological Survey (USGS) importieren	<a href="http://craterlake.wr.usgs.gov/">http://craterlake.wr.usgs.gov/</a> FreeGIS Tutorial Begleit-CD
Mosaik aus Luftbildern erstellen	FreeGIS Tutorial Begleit-CD
Flug über das Höhenmodell	<a href="http://craterlake.wr.usgs.gov/">http://craterlake.wr.usgs.gov/</a> FreeGIS Tutorial Begleit-CD
Dynamische Karten im WWW (Web Mapping)	<a href="http://craterlake.wr.usgs.gov/">http://craterlake.wr.usgs.gov/</a> FreeGIS Tutorial Begleit-CD <a href="http://www.mapit.de/">http://www.mapit.de/</a>



## 2. Übersicht zu den verwendeten Programmen

### 2.1. g3DGMV

- Betriebssysteme : GNU/LINUX und andere Unices, Windows
- zu beziehen über: <http://g3dgmv.sourceforge.net/index.html>
- Lizenz: GPL
- besprochene Version: 0.60
- Programmiersprache: C

Das Programm g3DGMV (3D Graphical Map Viewer) wurde als freies Betrachtungsprogramm für DEMs (Digital Elevation Model) und DLGs (Digital Line Graphs) entwickelt. Diese Modelle können von g3DGMV als 3-Dimensionales Bild für definierbare Projektionszentren berechnet werden. Weiter können Bilddateien (gif, png, jpeg ,xpmi) 2-Dimensional angezeigt werden. Verschiedene Werkzeuge können zum Hinzufügen von eigenen Informationen in die Darstellungen benutzt werden. In der besprochenen Version sind diese Werkzeuge noch nicht vollständig implementiert.

### 2.2. GRASS

- Betriebssystem:
- zu beziehen über:
- Lizenz:
- Programmiersprache:
- besprochene Version:

### 2.3. VTP Enviro

- Betriebssystem: Win32, FreeBSD, GNU/Linux, und Irix

- zu beziehen über: <http://vterrain.org/privacy.html#License>
- Lizenz: MIT
- Programmiersprache:
- besprochene Version:

Siehe Abschnitt VTBUILDER.

## 2.4. VTBUILDER

- Betriebssystem: Win32, FreeBSD, GNU/Linux, und Irix
- zu beziehen über: <http://vterrain.org/privacy.html#License>
- Lizenz: MIT
- Programmiersprache:
- besprochene Version:

VTBuilder ist ein Teil des VTP (Virtual Terrain Projects). Um eine möglichst genaue Wiedergabe der Landschaft im Rechner zu erzeugen können nicht nur eine Vielzahl von Höhenmodellen gelesen werden, unter anderen: DEM, SDTS DEM, DTED, GTOPO30, CDF, Surfer GRD, BT und PGM, sondern auch Straßendaten im USGS DLG und ESRI SHP Format, Bauwerkdaten (BCF-Dateien), Vegetationsinformationen z.B. USGS LU/LC und ESRI SHP files sowie Artenlisten und Vegetationszonen aus Textdateien. Der Import von Bilddaten wird noch nicht unterstützt. Die bearbeiteten Daten können direkt mit VTP Enviro angezeigt werden.

## 2.5. DLG Viewer / dlgv32

- Betriebssystem: Windows 95, Windows NT
- zu beziehen über: <http://mcmcweb.er.usgs.gov/viewers/>
- Lizenz: Unterliegt dem Public Domain Status
- Programmiersprache:
- besprochene Version: 3.7

Die USGS Software dlgv32 wurde erstellt um eine Vielzahl von kartographischen Daten betrachten zu können. Die Ursprüngliche Entwicklung beschränkte sich auf das SDTS Format. die Software hat keine Editierfunktionen und ist kein Ersatz für ein ausgereiftes GIS - System.



## 2.6. dem3D

- Betriebssystem: Windows 95, Windows NT, Windows 2000
- zu beziehen über: <http://craterlake.wr.usgs.gov/dem3d.html>
- Lizenz:
- Programmiersprache:
- besprochene Version: 2.0a

Das Programm dem3D ist ein frei herunterzuladenes Visualisierungswerkzeug für US-GS ASCII- und SDTS-DEMs.

## 2.7. GPS Point

- Betriebssystem: GNU/Linux, MACOS X, BSD
- zu beziehen über: <http://scampi.physik.uni-konstanz.de/~tschank/gpspoint/>
- Lizenz: GPL
- Programmiersprache: C++
- besprochene Version: 1.010708

GPS Point ist ein kommandozeilenorientiertes Programm um Wegpunkte, Routen und Tracks vom PC auf einem Garmin-GPS-Handheld zu verwalten.

## 2.8. GPSMan

- Betriebssystem: GNU/Linux and other Unices
- zu beziehen über: <http://www.ncc.up.pt/~mig/gpsman.html>
- Lizenz: GPL
- Programmiersprache: Tcl/Tk
- besprochene Version: 5.4

GPS Manager (GPSMan) ist ein grafischer Manager für GPS-Handys. Mit ihm können Daten auf den GPS-Handys inspiziert und zu verändert werden. GPSMan unterstützt Empfänger von Garmin und Lowrance.

## 2.9. DPSIP

- Betriebssystem: Windows 95, Windows 98, Windows NT, Netbsd/Openbsd/Freebsd, GNU/Linux
- zu beziehen über: <http://www.wscc.com/wolfgang/gps/dgps-ip.html#bulk>
- Lizenz: GNU Public License (GPL)
- Programmiersprache: C
- besprochene Version: 1.32 (GNU/Linux, BSD)

DGPSIP Ist ein Programm um Pseudorange-Korrekturdaten über das Internet zu Empfangen und an einen GPS-Empfänger zu übermitteln der die Fähigkeit hat entsprechende Korrekturdaten zu verarbeiten.

## 2.10. Mapit!

- Betriebssystem: GNU/Linux, Windows 98, NT4, 2K, ME
- zu beziehen über: <http://www.mapit.de/>
- Lizenz: LGPL (GNU Library General Public License)
- Programmiersprache: Python
- besprochene Version: 1.0.1-1

MapIt! ist eine Web-Anwendung die es erlaubt via Web-Browser auf Rasterkarten zu navigieren, hinein- und herauszuzoomen sowie darzustellende Objekte/Objektklassen auszuwählen und auf der Karte zu identifizieren.

## 2.11. MapServer

- Betriebssystem: GNU/Linux, Win2k, WinNT, Win9x
- zu beziehen über: <http://mapserver.gis.umn.edu/>
- Lizenz: MapServer License
- Programmiersprache: C
- besprochene Version: 3.5

MapServer ist eine OpenSource Entwicklungsumgebung um Internetanwendungen für die Darstellung von räumlichen Daten zu erstellen. Die Software baut auf einer Vielzahl von OpenSource oder Freeware (freie Software) Systemen auf, wie zum Beispiel Shapelib, FreeType, Proj.4, libTIFF, Perl und anderen. MapServer unterstützt populäre Skriptsprachen um Zugriff auf die Entwicklungsumgebung zu gewähren, darunter Perl, Python, Tk/Tcl, Guile und Java. Dadurch ist es zum Beispiel möglich über das Perl DBI Modul auf Datenbanken wie Oracle, Sybase oder MySQL zuzugreifen.



## **3. Freie Geo-Daten**

### **3.1. Bedeutung von Freiheit bei Geo-Daten**

### **3.2. Übersicht zu verfügbaren freien Geo-Daten**

Alle für die im Tutorial besprochenen Aufgaben nötigen Daten befinden sich auf den beiliegenden CD-ROM.

Weitere Datenquellen finden sich beispielsweise im WWW unter:

- <http://freegis.org/geo-data.de.html>
- <http://duff.geology.washington.edu/data/raster/tenmeter/index.html>
- <http://edcwww.cr.usgs.gov/doc/edchome/ndcdb/ndcdb.html>
- <http://craterlake.wr.usgs.gov/>

### **3.3. Bewertung von Daten**

Für die Bewertung von Daten hinsichtlich ihrer Eignung für die gestellte Aufgabe sind vielerlei Informationen nötig und Abhängigkeiten zu beachten.

Darunter:

- das Alter der Daten,
- die Genauigkeit der Positions- und Inhaltsangaben,
- das Auftreten von Ungenauigkeiten während der Aufnahme oder der Aufbereitung der Daten,
- die verschiedenen Grundlagen von Daten und damit ihre mögliche oder unmögliche Kombination,
- die Aufgabenstellung der die Daten genügen müssen

Dies sind nur einige wenige Kriterien für die Beurteilung von Daten.



## 4. Punktdaten einlesen und digitales Höhenmodell erzeugen

### Aufgabenstellung

Das Problem aus einem Punkthaufen verwertbare Daten zu erzeugen tritt häufig im Vermessungswesen auf, z.B. bei der Auswertung von Höhenaufnahmen, bei denen ein mehr oder weniger enges Raster von Höhenpunkten aufgemessen wird oder bei der Aufnahme von Profilen für eine Trassenplanung. Im folgenden wird eine eher ungewöhnliche Aufgabe besprochen.

### Beispieldaten

Freundlicherweise hat Herr Dr.-Ing. Reinking, Professor an der Fachhochschule Oldenburg/Ostfriesland/Wilhelmshaven für dieses Tutorial einen Datensatz einer „Höhenaufnahme“ oder besser „Tiefenaufnahme“ zur Verfügung gestellt, der im Hafenbecken von Elsfleth mittels Echolot und GPS-Positionsbestimmung ermittelt wurde. Es wird ausdrücklich darauf hingewiesen, daß diese Daten und die daraus entstehenden Ergebnisse in keinsten Weise für Navigationszwecke verwendet werden können. Der Datensatz `elsfleth-1.0` (17 KB) unterliegt der GNU General Public License.

### Weitere Verwendung der Beispieldaten

Die hier bearbeiteten Daten werden in den folgend aufgezählten Kapiteln benutzt:

- Kapitel 5 Seite 51 ff.

### 4.1. Lösung mit GRASS

Es wurde GRASS 5.0pre1 unter einer Debian GNU/Linux Installation (woody) verwendet.

Da diese Aufgabe eine Grundlage für viele weitere Bearbeitungen und Analysen mit GRASS ist, wird sie hier exemplarisch einmal sehr ausführlich besprochen, um dem Nutzer eine Vorstellung von der Arbeit mit GRASS zu vermitteln.

### 4.1.1. Vorbereitungen

Der Programmaufruf erfolgt an der Kommandozeile durch Eingabe von :

```
bash$ grass5
```

Die 5 weist auf die benutzte Programmversion hin. Das von GRASS generierte Startmenue erscheint und wird ähnlich dem unten aufgeführten ausgefüllt und mit der Tastenfolge *Esc Enter* beendet. Je nach Installation und Systemeinstellungen startet GRASS auch einen grafischen Eingabedialog, die Bedienung ist analog zur textbasierten Benutzerschnittstelle.

```

                                GRASS 5.0.0pre1

LOCATION: This is the name of an available geographic location.
        -spearfish-
        is the sample data base for which all
        tutorials are written.

MAPSET:  Every GRASS session runs under the name of a MAPSET.
        Associated with each MAPSET is a rectangular
        COORDINATE REGION and a list of any new maps created.

DATABASE: This is the unix directory containing the
        geographic databases

        The REGION defaults to the entire area of the chosen
        LOCATION.
        You may change it later with the command: g.region
-----
LOCATION:  elsfleth_____ (enter list for a list of locations)
MAPSET:   hafent_____ (or mapsets within a location)

DATABASE: /tmp/test/_____

        AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
        (OR <Ctrl-C> TO CANCEL)
```

Die *LOCATION* ist der übergeordnete Arbeitsbereich in dem ein oder mehrere zu bearbeitende *MAPSET* liegen. Diese *LOCATION* benötigt bei der Erstellung einige Parameter die zur korrekten Darstellung und Bearbeitung des Projektes erforderlich sind:

**UTM Location** Bei auf UTM basierenden Daten zum Beispiel die entsprechende UTM-Zone, die Daten zu dem verwendeten Ellipsoid (grs80, Hayford, Bessel, Krassovski,...) und die genutzte Koordinatenrealisierung, das sogenannte Datum (WGS84, NAD27, NAD83, und weitere).



**x-y Location** Am unproblematischten ist es, ein kartesisches Koordinatensystem zu benutzen (x,y - Menüpunkt *a*). Eine graphische Bedienoberfläche kann mit dem Befehl aufgerufen werden.

**Gauss Krüger Location** Um eine *LOCATION* für die in Deutschland gebräuchlichen Gauss-Krüger Koordinaten zu erstellen, sind einige Angaben erforderlich. Besonders bei einer geplanten Konvertierung in andere Koordinatensysteme ist auf die korrekte Eingabe der Parameter zu achten. Wird eine neue Location erstellt erscheint eine Auswahlmaske ähnlich der folgenden:

```
LOCATION <test> - doesn't exist

Available locations:
-----
craterlake
-----

Would you like to create location <elsfleth> ? (y/n)y
```

Auch die nächste Maske wird mit „ja“ beantwortet.

```
To create a new LOCATION, you will need the following
information:

1. The coordinate system for the database
   x,y (for imagery and other unreferenced data)
   Latitude-Longitude
   UTM
   Other Projection
2. The zone for the UTM database
   and all the necessary parameterOther Projection
   coordinate system? (y/n) [y] y s for projections other
   than Latitude-Longitude, x,y, and UTM
3. The coordinates of the area to become the default region
   and the grid resolution of this region
4. A short, one-line description or title for the location

Do you have all this information for
location <elsfleth>? (y/n)y
```

Da eine Gauss-Krüger Location erstellt werden soll ist „D“ auszuwählen und zu bestätigen.

```
Please specify the coordinate system for location <test>
```

```
A  x,y  
B  Latitude-Longitude  
C  UTM  
D  Other Projection  
RETURN to cancel
```

```
> D
```

```
Other Projection coordinate system? (y/n) [y] y
```

Falls gewünscht kann eine 1-zeilige Beschreibung für die neue Location gespeichert werden. Dies ist sehr hilfreich wenn man in mehreren verschiedenen Locations arbeitet.

```
Please enter a one line description for location <elsfleth>
```

```
>Tiefenaufnahme Hafen Elsfleth
```

```
=====
```

```
Tiefenaufnahme Hafen Elsfleth
```

```
=====
```

```
ok? (y/n) [n] y
```

Die Projektionsart im Gauss-Krüger System ist eine transversale Mercatorprojektion (tmerc), weiter wird das Bessel-Rotationsellipsoid verwendet, durch Eingabe von List werden die verfügbaren Auswahlmöglichkeiten angezeigt. Ein spezielles Datum für die Koordinatenrealisierung kann hier angegeben werden. Im vorliegenden Fall reicht die Voreinstellung aus, da nur eine einzige „Koordinatenrealisierung“ verwendet wird. *Central Parallel* ist der Äquator also 0 Grad. *Central Meridian* der Mittelmeridian des jeweiligen Meridianstreifens, hier 9 Grad östlich von Greenwich. Der Maßstabsfaktor *Scale factor* ist in diesem Fall 0.9996. Um negative Rechtswerte zu vermeiden, werden jeweils 500000 Meter auf den Rechtswert des Mittelmeridians aufaddiert, dieses entspricht dann dem *False Easting*.

```

Please specify projection name
Enter 'list' for the list of available projections
Hit RETURN to cancel request

>tmerc

Please specify ellipsoid name
Enter 'list' for the list of available ellipsoids
Hit RETURN to cancel request
>bessel
Do you want to specify a map datum for this location?(y/n) [n]

Enter Central Parallel [lat_0] (23N) :0N

Enter Central Meridian [lon_0] (96W) :9E

Enter Scale Factor at the Central Meridian [k_0] [1.000]: 0.9996

Enter False Easting [x_0] [0.0000000000]: 500000

Enter plural form of units [meters]: meters

```

Hat man die Regionparameter für die neue Location (*DEFAULT REGION*) vorliegen, können sie jetzt eingegeben werden. Es wird aber auch später gezeigt, wie man die Regionsgrenzen aus den zu bearbeitenden Dateien in GRASS übernehmen kann. Die Auflösung (*GRID RESOLUTION*) sollte bei der Verarbeitung von Rasterdaten der gewünschten Auflösung des Ergebnisses entsprechen, eine Änderung ist aber jederzeit durch den Befehl *g.region* möglich. Diese Eingabemaske wird mit einem *ESC* gefolgt von *ENTER* bestätigt.

## Tips

- Da in der zu bearbeitenden Datei einige Punkte weniger als 1 m Abstand voneinander haben, ist es vorteilhaft die *GRID RESOLUTION* schon jetzt heraufzusetzen, Beispielsweise auf 0.5 Einheiten, in unserem Fall also 0,5 Meter.
- Wird die Auflösung sehr hoch angesetzt, so äußert sich das in einer längeren Bearbeitungsdauer, bringt aber nicht unbedingt aussagekräftigere Ergebnisse.

```

DEFINE THE DEFAULT REGION

===== DEFAULT REGION =====
| NORTH EDGE:1_____ |
WEST EDGE |             | EAST EDGE
0_____ |             | 1_____
| SOUTH EDGE:0_____ |
=====

PROJECTION: 99 (Other Projection)      ZONE: 0

GRID RESOLUTION
  East-West:    0.5_____
  North-South:  0.5_____

AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

```

Da die soeben gemachten Eingaben von grundsätzlicher Bedeutung für die weitere Bearbeitung sind, sie definieren sozusagen die Welt, in der sich die GRASS - Location befindet, werden sie in der folgenden Bildschirmausgabe noch einmal aufgeführt und vom Anwender nach entsprechender Kontrolle bestätigt.

```

projection: 99 (Other Projection)
zone:      0
north:     1
south:     0
east:      1
west:      0

e-w res:   1
n-s res:   1

total rows:      1
total cols:      1
total cells:     1

Do you accept this region? (y/n) [y] > y

```

Nachdem alle weiteren Abfragen bestätigt wurden, kommt man zurück zu dem Begrüßungsbildschirm, in dem man nun die neu erstellte Arbeitsumgebung *MAPSET* mit *ESC* und *ENTER* öffnen kann.

```
GRASS 5.0.0pre1

LOCATION: This is the name of an available geographic location.
        -spearfish-
        is the sample data base for which all
        tutorials are written.

MAPSET:  Every GRASS session runs under the name of a MAPSET.
        Associated with each MAPSET is a rectangular
        COORDINATE REGION and a list of any new maps created.

DATABASE: This is the unix directory containing the
        geographic databases

        The REGION defaults to the entire area of the chosen
        LOCATION.
        You may change it later with the command: g.region
-----
LOCATION:  elsfleth_____ (enter list for a list of locations)
MAPSET:  hafen_____ (or mapsets within a location)

DATABASE: /tmp/test/_____

        AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
        (OR <Ctrl-C> TO CANCEL)
```

Im nächsten Bildschirm kann schon mit GRASS gearbeitet werden, um zusätzlich die GUI aufzurufen gibt man den Befehl *tcltkgrass &* an der Kommandozeile ein. Das „Kaufmannsund“ & sorgt in diesem Aufruf dafür, daß der Befehl im Hintergrund läuft und nicht die Kommandozeile blockiert.

```
Welcome to GRASS 5.0.0pre1 (April 2001)
```

```
Geographic Resources Analysis Support System (GRASS) is Copyright,  
1999-2001 by the GRASS Development Team, and licensed under  
terms of the GNU General Public License (GPL).
```

```
This GRASS 5.0.0pre1 release is coordinated and produced by the  
GRASS Development Team headquartered at University of Hannover with  
worldwide support and further development sites located at Baylor  
University and the University of Illinois.
```

```
This program is distributed in the hope that it will be useful, but  
WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
General Public License for more details.
```

```
This version running thru the Bash Shell (/bin/bash)  
Help is available with the command:      g.help  
See the licence terms with:             g.version -c  
Start the graphical user interface with: tcltkgrass&  
When ready to quit enter:              exit  
GRASS:/tmp/test > tcltkgrass &
```

### Tipps:

- durch das Eingeben von *list* bekommt man bei vielen Eingabeaufforderungen in GRASS Auswahllisten zu Locations, Mapsets, Ellipsoiden, Daten, ... angeboten
- Eingaben an der Kommandozeile werden durch *ESC RETURN* übernommen

### 4.1.2. Anmerkungen zum Arbeiten mit GRASS

Das Geoinformationssystem GRASS ist streng Modular aufgebaut, d.h. zum Beispiel, daß viele einzelne Bearbeitungsschritte (Module/Programme/Kommandos) hintereinander aufgerufen werden können und werden, oftmals ohne das dem Benutzer bemerkt zu diese klar wird. Die Kommandos werden erkennbar durch ihren Namen gegliedert. So beginnen alle Kommandos die Rasteroperationen ausführen mit *r.* Die Vektorprogramme mit *v.* u.s.w.. Eine komplette übersicht über alle Kommandos mit einer kurzen Erklärung ihrer Wirkung befindet sich hier <sup>1</sup>. Die Ausführliche Beschreibungen, die sogenannten „Manual Pages“ mit Beschreibung der verlangten Ein- und Ausgaben, und der Auflistung von Optionen findet sich unter<sup>2</sup> im Bereich „Modules section“.

### 4.1.3. Öffnen eines Bearbeitungsfensters

Zur grundlegenden Benutzung von GRASS ist es erforderlich einen Monitor, d.h. ein Ausgabefenster zu öffnen, in dem die Ergebnisse der einzelnen Arbeitsschritte angezeigt werden

<sup>1</sup>[http://grass.itc.it/gdp/html\\_grass5/grass\\_commandlist.html](http://grass.itc.it/gdp/html_grass5/grass_commandlist.html)

<sup>2</sup>[http://grass.itc.it/gdp/html\\_grass5/index.html](http://grass.itc.it/gdp/html_grass5/index.html)

können. Dies geschieht mit dem Befehl *d.mon* . Hiermit wird ein interaktives Menü zur Verwaltung der Monitore aufgerufen. Es können 7 Monitore (Fenster auf dem Bildschirm) gleichzeitig betrieben werden. Ein Befehl wird immer auf dem jeweils zuvor ausgewählten Monitor ausgeführt. Monitore können geöffnet, geschlossen, ausgewählt und freigegeben werden. Nachdem Auswählen eines freien Monitors und dem beherzten Drücken auf *RETURN* öffnet sich das gewünschte Bearbeitungsfenster.

```
GRASS:~ > d.mon
```

#### MONITOR MENU

Making sure that the graphics monitor is running

- 1 - start a graphics monitor
- 2 - stop a graphics monitor

Choosing a graphics monitor for your graphics

- 3 - select a graphics device for output  
(currently selected monitor: x0)
- 4 - relinquish control of the graphics device  
(let someone else use it)

RETURN - quit

> 1

Currently selected graphics monitor: x0

Enter name of graphics monitor to start

Enter "list" for a list of known monitors

Enter "list -f" for a list with current status

Hit RETURN to return to menu

> x0

using default visual which is TrueColor

Visual is read-only or using a private colormap

ncolors: 32768

allocating memory...

Graphics driver [x0] started

hit RETURN ->

#### 4.1.4. Importieren von Punktdaten

Mit dem Befehl *s.in.ascii* (das „s“ steht in diesem Fall für „sites“) werden ASCII-kodierte Datensätze als Punktdaten in Grass importiert. Die *ascii*-Datei könnte z.B. Zeilenweise den Rechtswert, Hochwert und die Höhe (des Punktes) enthalten. Die Trennung der einzelnen

Werte erfolgt mittels eines Feldtrenners, des Parameters *fs=* im Befehlsaufruf. Dieser *fs* (Field Separator) kann alle möglichen Zeichen umfassen, Leerzeichen sind die hier Grundeinstellung. Soll ein Tabulator als *fs* benutzt werden wird so wird er mit *fs=tab* gesetzt.

```
GRASS:~ > s.in.ascii input=/home/heiko/elsfleth-1.0/tiefen.txt
sites=tiefen fs=tab
```

Um genauere Angaben über die soeben importierten Daten zu erhalten kann der Befehl *s.info* benutzt werden.

```
GRASS:/tmp/test > s.info tiefen
```

Aus dem folgendem Bildschirm können die Regionsparameter entnommen werden, um die Importierte Datei in den GRASS - Monitoren vollständig darstellen zu können. Dies kann per Hand über *g.region* und die folgende interaktive Benutzerführung geschehen.

```
tiefen read.
-----
SITES FILENAME: tiefen
-----
Header Information:
-----
Number of DIMENSIONS:          2
-----
          - - MIN - -      - - MAX - -
dim  1  3464538.762000  3464746.718000  Easting
dim  2  5901140.217000  5901596.918000  Northing

Type of CATEGORY information: CELL_TYPE
-----
          - - MIN - -      - - MAX - -
                  1                  493

Number of DOUBLE attributes:  1
-----
          - - MIN - -      - - MAX - -
dbl  1          -8.16          -2.6

Number of STRING attributes:  0
-----

TOTAL SITES COUNTED: 493
-----
```

Eine einfachere Möglichkeit ist, GRASS die Parameter aus der importierten Datei direkt zu übergeben. Dies geschieht, indem man dem Befehl *g.region* diejenige Datei im Aufruf



übergibt, aus der die Parameterinformationen extrahiert werden sollen. In diesem Fall also die *sites* - Datei „tiefen“. Auch Vektor- oder Rasterdateien lassen sich so von *g.region* auswerten. Natürlich muß das „Kennwort“ *sites* in einem solchen Fall durch *rast* oder *vect* ersetzt werden.

```
GRASS:~ > g.region sites=tiefen
```

#### 4.1.5. Anzeigen der Punktdaten

Der Befehl *d.sites* zeigt die ausgewählte Punktdatei *sitefile=* im aktiven Monitor an. Es können die gewünschte Farbe *color=*, Größe *size=* und die Symbolart *type=* an den Befehl übergeben werden.

```
GRASS:~ > d.sites type=x sitefile=tiefen color=gray size=5
```

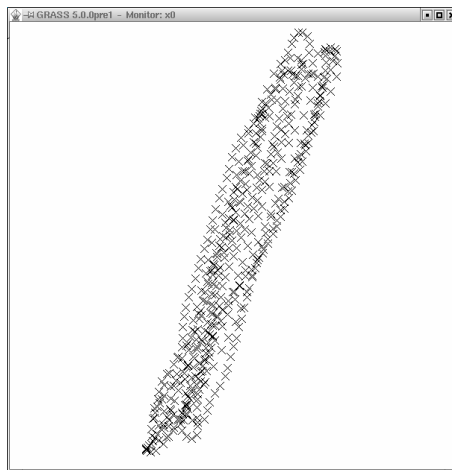


Abbildung 4.1.: GRASS-Monitor mit den importierten Punktdaten „Hafen Elsfléth“

Deulich ist hier die linienhafte „Fahrspur“ des sensortragenden Schiffes zu erkennen.

#### 4.1.6. Maskierung des befahrenen Hafengebietes

Um bei der anschließenden Interpolation zu verhindern, daß über das gesamte Gebiet des Mapsets eine Interpolation gerechnet wird, maskiert man das relevante Gebiet. Das hierzu benutzte Kommando *s.hull* wurde für diese Aufgabe nach einer Frage meinerseits an die GRASS-Mailingliste innerhalb von 3 Tagen aus dem vorhandenen Kommando *s.delauney* abgeleitet. Herzlichen Dank an Markus Neteler und Andrea Aime! Das Kommando *s.hull* erzeugt einen konvexen Hüllenvektor der alle Punkte umschließt.

```

GRASS:/spare/gisdata/grass/bin > s.hull

OPTION:  name of a sites file to be input
         key: sites
required: YES

Enter the name of an existing sites file
Enter 'list' for a list of existing sites files
Hit RETURN to cancel request
> tiefen
<tiefen>

OPTION:  name of a vector file to be output
         key: vect
required: YES

Enter a new binary file file name
Enter 'list' for a list of existing binary file files
Hit RETURN to cancel request
> mask
<mask>

FLAG: Set the following flag?
      Use all sites (do not limit to current region)?(y/n) [n]

```

Diese Vektorlinie kann mit *d.vect map=mask* im GRASS-Monitor angezeigt werden. Dieser von der Vektorlinie umschlossenen Fläche muß ein Wert (Label) zugewiesen werden, um aus ihr eine Rasterfläche zu erzeugen. Hierzu wird das Modul *v.digit* aufgerufen, nachdem mit *d.mon* ein GRASS-Monitor gestartet wurde. Da eine Vektordatei (die eben erzeugte Datei *mask*) ja schon besteht und nicht neu erzeugt werden soll, wird der Menüpunkt 3 *none* aufgerufen und anschließend der Name der Vektordatei übergeben.

```

GRASS:/spare/gisdata/grass/bin > v.digit

          Available Digitizers

      Name                Description
      ----                -
[1]   Calcomp            Calcomp digitizer, format 23 (binary)
[2]   Altek              Altek digitizer, model AC30, format 8 (binary)
[3]   none               Run digit without the digitizer.

Hit return to use digitizer in brackets
or type in number or name of other digitizer.

Select digitizer [none] : 3

Selected digitizer is: none

Enter the name of a map to work with.
  If name is entered that does not already exist, it
  will be created at this time.

DIGIT FILENAME
Enter 'list' for a list of existing digit files
Hit RETURN to cancel request
>mask

```

Im folgenden Menü können Angaben zu der Vektordatei gemacht werden, wichtig ist hier die Zeile *Maps's scale* bei der *1:1* eingetragen wird und die Ausdehnung der *area* die mit *s.info sites=tiefen* ermittelt werden kann. Die Annahme der eingetragenen Informationen geschieht mit *ESC ENTER*. Das anschließende *shall we continue* wird mit *yes* bestätigt.

Provide the following information:

Your organization \_\_\_\_\_  
Todays date (mon,yr) \_\_\_\_\_  
Your name \_\_\_\_\_  
Map's name \_\_\_\_\_  
Map's date \_\_\_\_\_  
Map's scale 1:1 \_\_\_\_\_  
Other info \_\_\_\_\_  
Zone 0 \_\_\_\_\_  
West edge of area 3464438 \_\_\_\_\_  
South edge of area 5901040 \_\_\_\_\_  
East edge of area 3464847 \_\_\_\_\_  
North edge of area 5901697 \_\_\_\_\_

AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE

Der GRASS-Monitor zeigt jetzt die Vektorhülle mit zugehörigem Maßstab und Nordpfeil. An der Kommandozeile wird das Bedienungsmenu für GRASS-DIGIT ausgegeben. Mit dem Kommando *L* für Label wechselt man in das entsprechende Untermenü.

```
l-----k
|GRASS-DIGIT Modified 4.10                               Main menu|
|-----|
|MAP INFORMATION                                         AMOUNT DIGITIZED|
|Name:                                                  # Lines:      0|
|Scale:          1                                     # Area edges: 1|
|Person:                                                # Sites:      0|
|Dig. Thresh.: 0.0300 in.                             - - - - -|
|Map Thresh.: 0.0008 meters                            Total points: 24|
|-----|
|OPTIONS:|
|          Digitizer:      Disabled|
|-----|
m-----j
|Digitize Edit Label Customize Toolbox Window Help Zoom Quit * !^|
|-----|
|GLOBAL MENU:Press first letter of desired command.|
|[Upper Case Only]|
m-----j
```

In diesen Menue wird mit *a* die Option *Label Areas* ausgewählt. Ein Name kann eingegeben werden, eine *Category* muß angegeben werden.

```

l-----k
| GRASS-DIGIT Modified 4.10                               Label Menu |
|-----|
| Label options:                                         |
| a - Label Areas           m - Label Multiple Lines    |
| l - Label Lines           M - Un-Label Multiple Lines |
| s - Label Sites           c - Label Contours           |
|                             i - Contour interval: < 5> |
| A - Un-Label Areas                                     |
| L - Un-Label Lines                                     |
| S - Un-Label Sites                                     |
|                                                         |
| B - Bulk Label Remaining Lines                         |
|                                                         |
| h - Highlight Lines of category #                     |
| d - Display Areas of category #                       |
|                                                         |
| q - Return to main menu                               |
|-----|
| Digitize Edit Customize Toolbox Window Help Zoom * ! ^ |
|-----|
| GLOBAL MENU: Press first letter of desired command.   |
| [Upper Case Only]                                     |
|-----|
| ?-----?                                             |
|                                                         |
| Enter Category Number (0 to quit):[0] 1              |
|-----|

```

Anschließend wird im GRASS-Monitor erst ein Punkt innerhalb der Fläche mit einem linken Mausklick markiert und mit der mittleren Maustaste bestätigt, danach die umschließende Vektorlinie. Die erzeugte *area* wird mit einem linken Mausklick dem GRASS-Monitor zugewiesen (Abb. 4.2). Nachdem man wieder im *Label Menu* angekommen ist wird *v.digit* durch Eingabe eines kleinen und eines großen *q* verlassen.

```

Select point within area:

Buttons:
  Left:  Choose this position
  Middle: Accept chosen point position
  Right: Abort/Quit

Point location:
EAST: 3464633.56
NORTH: 5901339.93

```

Um die Vektorlinie in eine Rasterdatei umzuwandeln verwendet man *v.to.rast*, nachdem

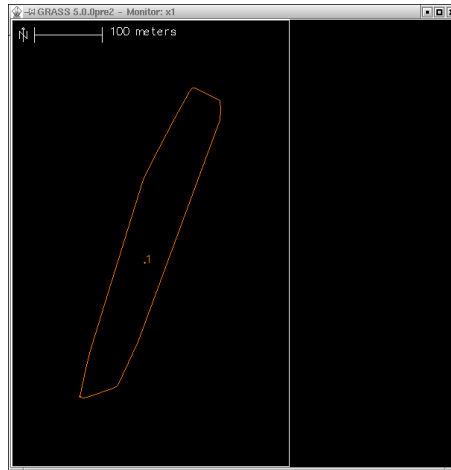


Abbildung 4.2.: GRASS-Monitor mit Digitalisierbildschirm *v.digit*

man mit *v.support mask* die von GRASS benötigten weiteren Dateien erzeugt. Die Frage nach den im Speicher zu haltenen Zeilen kann mit 512 bestätigt werden.

```
GRASS:~GRASS:/spare/gisdata/grass/bin > v.support mask
```

```
V.SUPPORT:
```

```
Selected information from vector header
```

```
Organization:
```

```
Map Name:
```

```
Source Date:
```

```
Orig. Scale: 0
```

```
No snapping will be done
```

```
Reading Vector file.
```

```
100%
```

```
Building areas
```

```
Building islands
```

```
Attaching labels
```

```
Number of lines: 1
```

```
Number of nodes: 1
```

```
Number of areas: 1
```

```
Number of isles: 1
```

```
Number of atts : 0
```

```
Number of unattached atts : 0
```

```
Snapped lines : 0
```

```

GRASS:/spare/gisdata/grass/bin > v.to.rast
OPTION:  vector input file
key: input
required: YES

Enter the name of an existing vector file
Enter 'list' for a list of existing vector files
Hit RETURN to cancel request
> mask

Enter a new raster file name
Enter 'list' for a list of existing raster files
Hit RETURN to cancel request
> mask

Parameters:
input  vector input file
output raster output file
rows  number of rows to hold in memory
default: 512
required: NO
enter option >
Loading vector information ...          0 mins 00 secs
Sorting areas by size ... 1 areas      0 mins 00 secs
Pass #1 (of 2)
  Processing areas ... 1 areas         0 mins 00 secs
  Processing lines ... 0 lines         0 mins 00 secs
  Processing sites ... 0 sites         0 mins 00 secs
  Writing raster map ...              0 mins 00 secs
Pass #2 (of 2)
  Processing areas ... 1 areas         0 mins 00 secs
  Writing raster map ...              0 mins 00 secs
Creating support files for raster map ... 0 mins 01 sec

Raster map <mask> done.
Total processing time:    0 mins 01 sec

```

Die Maske selbst wird in dem Befehl *r.mask* gesetzt, mit dem gleichzeitig auch Masken entfernt werden können.

```

GRASS://spare/gisdata/grass/bin > r.mask mask

MASK: Program for managing current GIS mask

current mask:          none

Options:
  1      Remove the current mask
  2      Identify a new mask
RETURN  Exit program

>2

Enter name of data layer to be used for mask
Enter 'list' for a list of existing raster files
Enter 'list -f' for a list with titles
Hit RETURN to cancel request
>mask

IDENTIFY THOSE CATEGORIES TO BE INCLUDED IN THE MASK

OLD CATEGORY NAME                                     CAT
                                                    NUM
. . . . .                                           0      0_
. . . . .                                           1      1_

Next category: end__ (0 thru 1)

AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
(OOR <Ctrl-C> TO CANCEL)

```

#### 4.1.7. Punktdaten in Rasterdaten umwandeln

**Schnelle Umwandlung** Aus den importierten Punktdaten wird ein Raster entsprechend der gewählten Auflösung *Grid Resolution* in den Regionsparametern interpoliert. Hierbei wird die Methode der numerischen Annäherung durch gewichtete Mittelbildung verwendet. Dem Befehl *s.surf.idw* werden als Parameter die Anzahl der zu Nutzenden Interpolationspunkte übergeben *npoints=5*. Die Ausgangseinstellung ist hier 12 Punkte. Je nach gegebener Punktdichte kann mit diesem Wert das Ergebnis beeinflusst werden (Abb. 4.3).

```

GRASS:~ > s.surf.idw input=tiefen output=tiefenraster npoints=5
Reading sites map (tiefen) ...
Interpolating raster map <tiefenraster> ... 657 rows ... done

```

**Steuerbare Umwandlung** Auch der Befehl *s.surf.rst* interpoliert Rasterdaten aus Punktdaten. Im Unterschied zu *s.surf.idw* können die entstehenden Rasterdateien verschiedene In-



halte haben. Mit *input=* wird die zu umzuwandelnde Datei aufgerufen. Weiter kann man bei den Ausgabedateien zwischen Höhen- *elev=*Dateiname, Neigungs- *slope=*Dateiname, Verhältniss- *aspect=*Dateiname, Profilkrümmungs- *pcurv=*Dateiname, Tangentialkrümmungs- *tcurv=*Dateiname und mittleren Krümmungsdateien *mcurv=*Dateiname wählen. Natürlich können auch mehrere Dateien ausgegeben werden.

Die Interpolation kann durch weitere Parameter beeinflusst werden, so können Minimalabstände zwischen zu interpolierenden Punkten angegeben werden *dmin=*, Überhöhungsfaktoren für die Attributwerte, in diesem Fall die Höhe, *zmult=*, Spannungs- und Glättungsparameter für die Oberflächenglättung *tension=* *smoothing=*. Eine Mindestanzahl von zu interpolierenden Punkten kann angegeben werden *npmin=*, die Ausgangseinstellung ist hier 200, dieser Wert sollte aber bei heterogenen, also zum Beispiel bei Daten die nicht gleichen Ursprungs sind, erhöht werden.

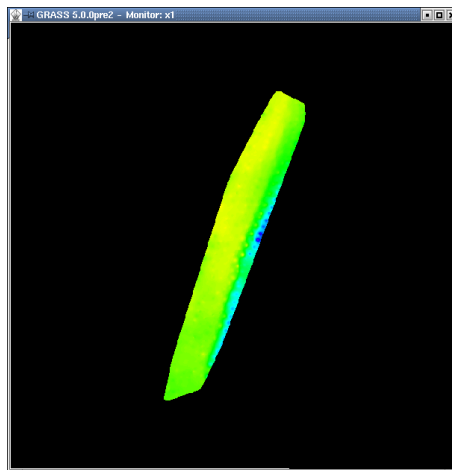


Abbildung 4.3.: GRASS-Monitor mit den interpolierten Rasterdaten „Hafen Elsfleth“

```

GRASS:/spare/gisdata/grass/bin > s.surf.rst -d -t input=tiefen
elev=tiefen2

Authors: original version L.Mitas, H.Mitasova
        GRASS implementation I.Kosinovsky, D.P.Gerdes
see references in manual page or at:
http://www2.gis.uiuc.edu:2280/modviz/papers/listsj.html

Warning: strip exists with insufficient data
Warning: points are more dense than specified 'DMIN'--ignored
2 points
Warning: there is less than 700 points for interpolation,
no segmentation is necessary, to run the program faster,
set segmax=700 (see manual)

The number of points in sites file is 493
The number of points outside of region 0
The number of points being used (after reduction) is 491

Processing all selected output files
will require 1075492 bytes of disk space for temp files

dnorm = 196.687138, rescaled tension = 7.867486
Warning: taking too long to find points for interpolation-
-please change the region to area where your points are.
Continuing calculations...
WARNING:
Overshoot -- increase in tension suggested.
Overshoot occurs at (115,367) cell
The z-value is -8.716646,zmin is -8.160000,zmax is -2.600000
Warning: taking too long to find points for interpolation-
-please change the region to area where your points are.
Continuing calculations...
100%
Percent complete: history initiated
GRASS:/spare/gisdata/grass/bin >

```

### Tipps:

- da selten alle Parameter benötigt werden hält sich der Mehraufwand bei dem Aufruf von *s.surf.rst* gegenüber *s.surf.idw* in Grenzen. Die Ergebnisse sind aber oftmals eher zur Erstellung von Digitalen Geländemodellen geeignet.
- Zu bedenken bleibt aber, das sich die Dauer der Berechnung von *s.surf.rst* in die Länge ziehen kann.

### 4.1.8. Rasterdaten anzeigen

Einer der am häufigsten genutzten Befehle bei GRASS ist sicherlich *d.rast*. Er dient dazu Rasterdateien in dem ausgewählten Monitor anzuzeigen. Die gewünschte Rasterdatei wird

mit `map=` ausgewählt. Die Option `-o` filtert alle Rasterzellen mit dem Wert 0 aus.

```
GRASS:~ > d.rast -o map=tiefen2
```

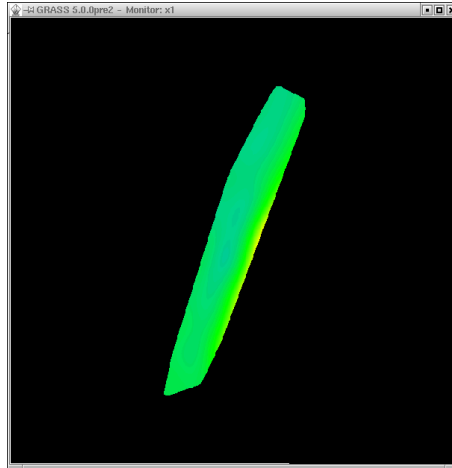


Abbildung 4.4.: GRASS-Monitor mit den interpolierten maskierten Rasterdaten „Hafen Elsfleth“ aus dem Modul `s.surf.rst`.

### Tipps

- sollte sich eine Rasterdatei von GRASS nicht anzeigen lassen sollte zuallererst überprüft werden ob nicht eine beispielsweise vorangehend fälschlicherweise aufgerufene „leere“ Datei die aktuell gewünschte Anzeige verdeckt. Mit `d.erase` kann eine solche die Anzeige verdeckende Rasterdatei beseitigt werden
- weiter sollte mit `d.mon` überprüft werden ob der gewünschte Monitor aktiv ist
- ein weißer Bildschirm nach dem Befehl `d.rast` kann daraufhin deuten das die Ausdehnung der Region und der Rasterdatei nicht übereinstimmen. Durch den Befehl `g.region rast=Rasterdatei` werden die Regionsparameter angepasst. Dieser Befehl funktioniert auch bei Vektordateien `g.region vektor=Vektordatei`

### 4.1.9. 3 - dimensionale Ansicht des Höhenmodells

Um das aus den Punktdaten interpolierte Höhenmodell in einer dynamischen 3-dimensionalen Ansicht ausgeben zu lassen, wird das GRASS-Modul `nviz2.2` aufgerufen. Hierbei ist zu beachten das das Programm beim Aufruf `nviz` klein geschrieben wird, wie fast alle `GRASS` - Kommandos. Ein Ausführliches Tutorial zu `nviz` findet sich unter <sup>3</sup>.

<sup>3</sup>[http://grass.itc.it/gdp/html\\_grass5/nviz/index.html](http://grass.itc.it/gdp/html_grass5/nviz/index.html)

```
GRASS:~ > nviz2.2 el=tiefen2
```

```
Version: @(#) GRASS 5.0.0pre1 (April 2001)
```

```
Authors: Bill Brown, Terry Baker, Mark Astley, David Gerdes  
modifications: Jaro Hofierka
```

```
Please cite one or more of the following references in publications  
where the results of this program were used:
```

```
Brown, W.M., Astley, M., Baker, T., Mitasova, H. (1995).  
GRASS as an Integrated GIS and Visualization System for  
Spatio-Temporal Modeling, Proceedings of Auto Carto 12,  
Charlotte, N.C.
```

```
Mitasova, H., W.M. Brown, J. Hofierka, 1994, Multidimensional  
dynamic cartography. Kartograficke listy, 2, p. 37-50.
```

```
Mitas L., Brown W. M., Mitasova H., 1997, Role of dynamic  
cartography in simulations of landscape processes based  
on multi-variate fields.
```

```
Computers and Geosciences, Vol. 23, No. 4, pp. 437-446
```

```
http://www2.gis.uiuc.edu:2280/modviz/viz/nviz.html
```

```
The papers are available at
```

```
http://www2.gis.uiuc.edu:2280/modviz/
```

Es folgt noch eine ganze Reihe von Statusmeldungen über hoffentlich erfolgreich geladene Programmteile. Das Ergebniss sollte eine GUI zur Steuerung von *nviz2.2* sein (Abb. Abb. 4.5). Gleichzeitig erscheint der Ausgabemonitor von *nviz2.2* mit einem Drahtgittermodell des aufgerufenen Höhenmodells (Abb. Abb. 4.6). Durch einen Mausklick auf die Schaltfläche *Surface* kann dieses Gittermodell mit einer farbkodierten Oberflächentextur belegt werden. Grundlage dieser Farben sind die Höheninformationen zu den einzelnen Punkten. Blickwinkel (*perspektive*), Anordnung des Projektionszentrums (Maussensitive „Windrose“), (*height*) und Überhöhungsfaktor der Höhen (*zexag*) können durch die entsprechenden Schieberegler justiert werden. Weitere Werkzeuge zur Anpassung der Darstellung finden sich unter *Panel*, hier können Menues zur Anpassung des Beleuchtungsmodells, Änderung der Texturen, Auswahl anderer Texturen u.s.w. aufgerufen werden. Die aktuelle Darstellung kann mit *File/Image Dump* gespeichert werden. Nach einigem Experimentieren sollten solche oder ähnliche Ergebnisse zu erzielen sein (Abb. 4.7).

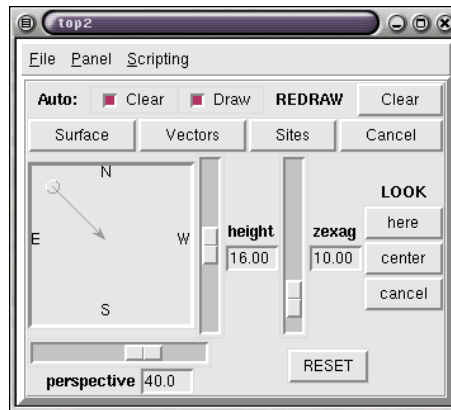


Abbildung 4.5.: Graphical User Interface des GRASS-Moduls nviz2.2

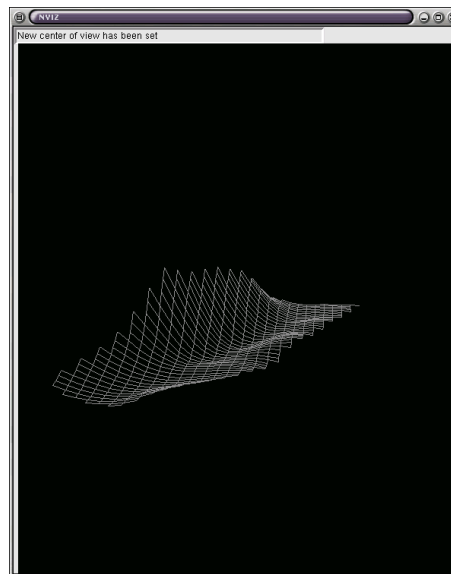


Abbildung 4.6.: Ausgabebildschirm von nviz2.2 mit Gittermodell „Hafen Elsfleth“

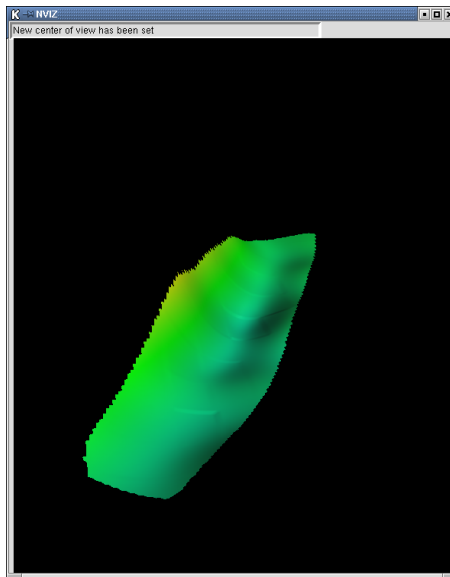


Abbildung 4.7.: Ausgabebildschirm von nviz2.2 mit 3D-Ansicht „Hafen Elsfleth“

# 5. Triangulation in Punktdaten

## Aufgabenstellung

Um digitale Geländemodelle zu erzeugen wird durch Triangulation, d.h. die Erzeugung einer geeigneten Dreiecksvermaschung versucht, zwischen den gegebenen Punkten einen sinnvollen Interpolationsansatz zu erhalten. Das zu erzeugende Modell ist der Örtlichkeit soweit wie möglich anzunähern. Zum Erzeugen einer solchen Triangulation gibt es mehrere Ansätze.

## Beispieldaten

Es werden die in Kapitel 4 eingelesenen Punktdaten der Echolotmessung im Hafen von Elsflëth `elsfleth-1.0` (17 KB) verwendet.

## 5.1. Lösung mit GRASS

Es wurde GRASS 5.0pre1 unter einer Debian GNU/Linux Installation (woody) verwendet.

Eine in GRASS implementierte Möglichkeit zur Triangulation ist die Dreiecksvermaschung nach Delaunay. Eine Beschreibung der Delaunay Triangulation findet sich hier <sup>1</sup>. Der Aufruf erfolgt denkbar einfach durch den Befehl `s.delaunay`. Im darauffolgenden Dialog werden die Punkt- und die zu erzeugende Vektordatei benannt, im Beispiel *tiefen* und *dreieck*.

---

<sup>1</sup><http://www.rz.tu-ilmenau.de/~japel/vortrag/gdv/delaunay.htm>

```

GRASS:/tmp/test > s.delaunay

OPTION:  name of a sites file to be input
         key: sites
required: YES

Enter the name of an existing sites file
Enter 'list' for a list of existing sites files
Hit RETURN to cancel request
> tiefen
<tiefen>

OPTION:  name of a vector file to be output
         key: vect
required: YES

Enter a new binary file file name
Enter 'list' for a list of existing binary file files
Hit RETURN to cancel request
> dreieck
<dreieck>

FLAG: Set the following flag?
      Use all sites (do not limit to current region)?(y/n) [n]

FLAG: Set the following flag?
      Quiet?(y/n) [n]
Doing Delaunay triangulation ...      100%
GRASS:/tmp/test >

```

Die Ausgabe erfolgt an einem GRASS - Monitor mit *d.vector*.

```

d.vect -v map=drei

Initializing [drei] ... Plotting ... Done
GRASS:/tmp/test > d.vect -v map=drei
Vector file [drei]

Selected information from dig header
Organization:  US Army Const. Eng. Rsch. Lab
Map Name:
Source Date:
Orig. Scale:   1

North: 5901697
South: 5901040
East:  3464847
West:  3464440

Initializing [drei] ... Plotting ... Done

```

Durch einem gemeinsamen Aufruf der grundlegenden Rasterdatei und der erzeugten



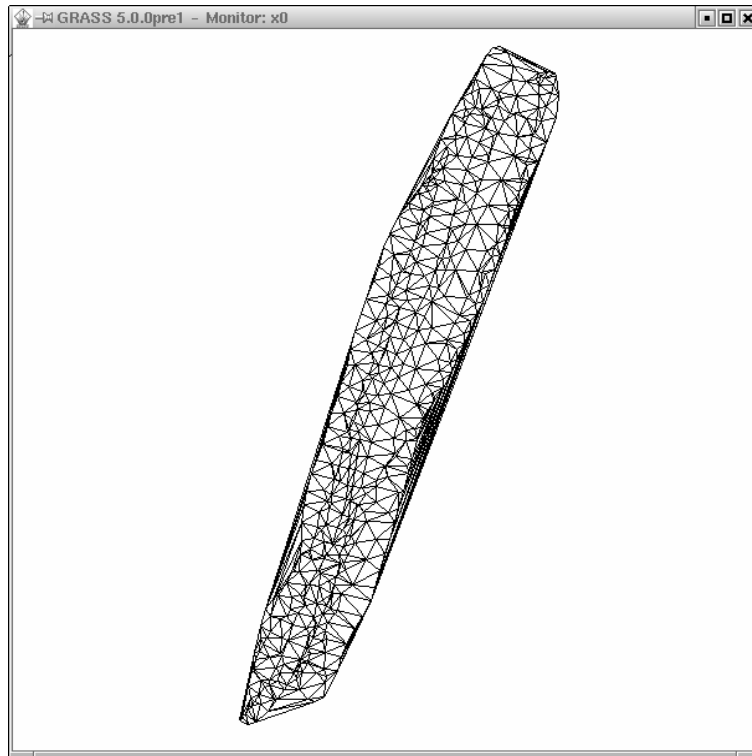


Abbildung 5.1.: Ausgabebildschirm mit Delaunay-Triangulation „Hafen Elsfleth“

Triangulation in *Nviz2.2* siehe Kapitel 4 wird eine 3-Dimensionale Darstellung des Triangulationsmodells erzeugt. Hierzu wird dem Aufruf *Nviz2.2* zusätzlich die entsprechende Vektordatei übergeben. Ein Mausklick auf die Schaltfläche *Vectors* in der *Nviz2.2*-GUI baut die gewünschte Ansicht auf ( s. Abb 5.2).

```
GRASS:~ > nviz2.2 el=tiefen2 ve=delau
```

```
Version: @(#) GRASS 5.0.0pre1 (April 2001)
```

```
Authors: Bill Brown, Terry Baker, Mark Astley, David Gerdes  
modifications: Jaro Hofierka
```

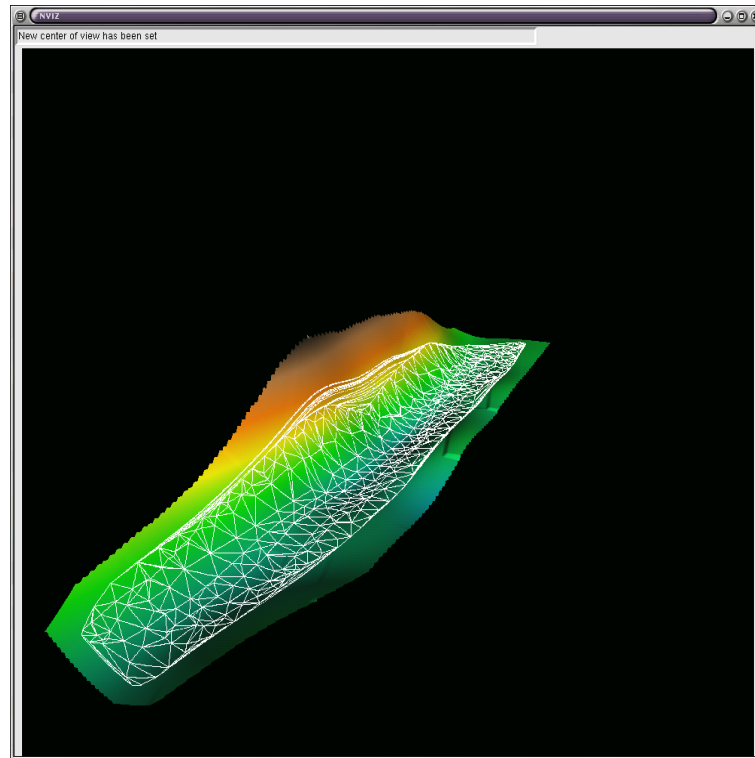


Abbildung 5.2.: Ausgabebildschirm mit 3-dimensionaler Ansicht der Delaunay-Triangulation „Hafen Elsfleth“

## 6. Digitales Höhenmodell des United States Geological Survey (USGS) importieren.

### Aufgabenstellung

Das USGS hält Digitale Höhenmodelle (digital elevation models) in den Maßstäben 1 : 250.000 und 1 : 24.000 zum Herunterladen vor <sup>1</sup>. Diese Höhenmodelle sind im SDTS <sup>2</sup>(Spatial Data Transfer Standard) gespeichert. Dieses Format wurde speziell zum verlustfreien Austausch von Georeferenzierten Daten zwischen verschiedenen Computersystemen geschaffen. Viele Programme zum Betrachten und Bearbeiten von Höhenmodellen arbeiten noch mit dem Vorgängerformat, dem sogenannten USGS ASCII Format. Programme zum konvertieren von SDTS in das USGS ASCII Format können bei <sup>3</sup> heruntergeladen werden. Im Folgenden wird das Importieren eines USGS ASCII formatierten Höhenmodelles beschrieben.

### Beispieldaten

Die hier verwendeten Beispieldaten hält das USGS in dem sogenannten „Crater Lake Data Clearinghouse“<sup>4</sup> i zusammen mit einer Menge an zusätzlichen Informationen, (Metadaten) über diesen Nationalpark der USA zum herunterladen vor. Unter anderem sind dort Daten über die Geomorphologie, Ökologie, Kulturgeschichte und vieles mehr gespeichert.

### Weitere Verwendung der Beispieldaten

Die ab Seite 59 ff bearbeiteten Daten werden in den folgend aufgezählten Kapiteln benutzt:

- Kapitel 7 Seite 63 ff.
- Kapitel 8 Seite 67 ff.
- Kapitel 9 Seite 71 ff.
- Kapitel 10 Seite 83 ff.

---

<sup>1</sup><http://edcwww.cr.usgs.gov/doc/edchome/ndcdb/ndcdb.html>

<sup>2</sup><http://mcmcweb.er.usgs.gov/sdts/>

<sup>3</sup><http://geopotential.com/docs/sdts2dem/sdts2dem.html>

<sup>4</sup><http://craterlake.wr.usgs.gov/intro.html>

- Kapitel 14 Seite 122 ff.
- Kapitel 15 Seite 135 ff.

## 6.1. Lösung mit g3DGMV (3D Graphical Map Viewer)

Es wurde die Version 0.60 von g3DGMV auf einer Windows 2000 Installation verwendet.

Die Bedienung von g3DGMV erfolgt durch die aufgerufene GUI. Wie die aktuelle Versionsnummer (0.60) vermuten lässt sind noch nicht alle Features implementiert, aber eine grundlegende Betrachtung und Bearbeitung von Höhenmodellen ist möglich.

Die Schaltfläche *File* öffnet ein Menü zur Auswahl der gewünschten Datei (Abb. 6.1). Ein Mausklick auf den Reiter *Map Viewer* öffnet einen Betrachtungsmonitor. Ein rechter Mausklick in den Monitor ruft ein Auswahlmenü zur Bearbeitung des Höhenmodelles auf. Hier kann z.B. zwischen einer 2D- und 3D-Ansicht, *Position Default* und *Position Rotated*, umgeschaltet werden. Die Menüauswahl *Map Line* erzeugt eine Linienansicht des 3D-Modelles. Mit *Rotate Map* kann man das Modell um alle 3 Achsen drehen. Die Ausgangsposition erreicht man mit *Default Map*. Die Farbe und die Koordinaten eines angewählten Punktes können mit *Pick Color* und *Find Position* ermittelt werden. Durch *Rotate Wire Frames* wird ein Ausschnitt des DEM als drehbares Drahtgittermodell erzeugt, *Move Wire Frame* verschiebt den Ausschnitt im Fenster. *Tools* öffnet ein Menü mit Werkzeugen zur Bildbearbeitung. Eine erzeugte Ansicht kann mit *Save* als PNG-Datei gespeichert werden.

### Tipps:

- Bei einer deutsch lokalisierten Linux-Installation muss vor dem eigentlichen Programmaufruf ein `export LC_ALL=en` erfolgen, um die benutzte Shell anzuweisen, Punkte anstatt Kommata als Dezimaltrenner aufzufassen.

## 6.2. Lösung mit VTBUILDER

Es wurde eine Windows 2000 Installation verwendet.

Das Programm VTBUILDER ist eine Application des Virtual Terrain Project (VTP)<sup>5</sup>. Neben dem reinen Darstellen von DEMs (Abb. 6.2) mittels *Layer/Import Data* und der Auswahl *Elevation* kann VTBUILDER unter anderem geographische und UTM Koordinaten wechselseitig konvertieren und so gemeinsam darstellen. Auch Koordinaten verschiedenen Datums werden zusammen verarbeitet. VTBUILDER kann eine Vielzahl von DEM - Dateitypen lesen, z.B.: USGS ASCII DEM, SDTS DEM, DTED, GTOPO30, CDF, Surfer

---

<sup>5</sup>[www.vterrain.org](http://www.vterrain.org)

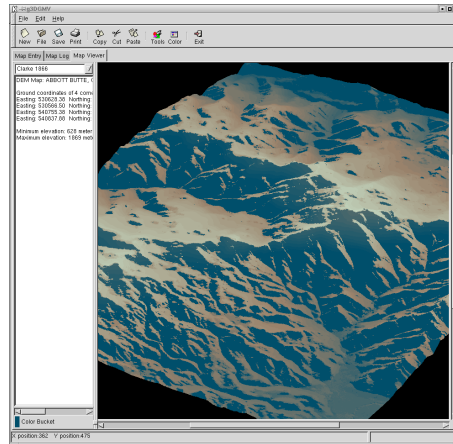


Abbildung 6.1.: Ausgabebildschirm mit importiertem USGS Modell „Abbott Butte, Oregon“

GRD, PGM und es importiert und schreibt Binary Terrain (BT) Dateien<sup>6</sup>. Die DEMs werden auf Wunsch auf einer Weltkarte lagerichtig angezeigt *View/Worldmap* und die UTM-Meridianstreifen können eingblendet werden (Abb. 6.3) *View/UTM Boundaries*.

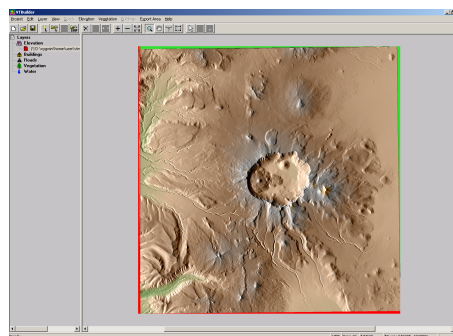


Abbildung 6.2.: Ausgabebildschirm mit importiertem USGS Modell „Crater Lake“

Im unteren rechten Bildrand wird die UTM-Region, das Datum der Koordinatenrealisierung und die aktuellen Mauskoordinaten angezeigt. Mit *View/Obtain Distance* kann in den Meß-Modus geschaltet werden. Durch einen Mausklick und das Bewegen des Mauszeigers bei gedrückter Maustaste können zwei Punkte angefahren werden, zwischen denen die Entfernung in Metern und entsprechend die Unterschiede in UTM- und Geographischen Koordinaten angezeigt werden.

Weiter unterstützt VTBUILDER das Einlesen von USGS DLG-Dateien und ESRI Shape-Files.

<sup>6</sup><http://vterrain.org/Implementation/BT.html>

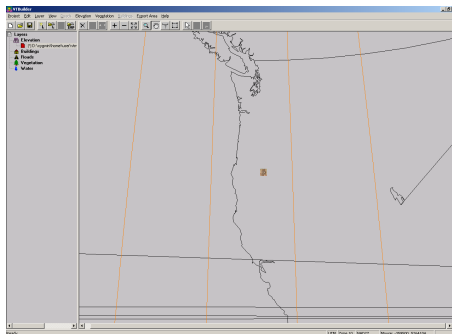


Abbildung 6.3.: VTBUILDER mit Weltkarte und UTM-Meridianstreifen

### 6.3. Lösung mit dem3D

Es wurde die Version 2.0a von dem3D auf einer Windows 2000 Installation verwendet.

Um USGS ASCII-DEMs und SDTS-DEMs unter Windows anzeigen zu lassen, hat das USGS das Programm dem3D zur Verfügung gestellt. Nach dem Start wird das gewünschte DEM mit *File/Open* aufgerufen und angezeigt (Abb. 6.4). Durch Gedrückthalten der linken Maustaste und gleichzeitiges Bewegen der Maus kann das Modell in allen 3 Achsen bewegt werden. Mit der Schaltfläche *Shader* kann zwischen verschiedenen Oberflächen- und Beleuchtungsmodellen gewählt werden. Wahlweise kann unter der Schaltfläche *Szene* mit *Camera* das Projektionszentrum bestimmt und mit *Lights* Farbe und Standpunkt der Beleuchtungsquelle eingestellt werden. Die Himmelsrichtungen können durch *Compass* angezeigt und ein Gitter mit *Grid* bzw. eine Wasseroberfläche mit *Water* eingeblendet werden (Abb. 6.5). Eine Statuszeile im linken unteren Fensterrahmen informiert über die Mausposition, Lage des Projektionszentrums, Überhöhung der Höhenkoordinaten und die gewählte Auflösung.

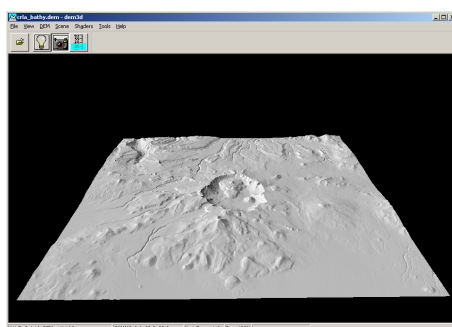


Abbildung 6.4.: Anzeige des USGS-DEM „Craterlake“ mit dem3D

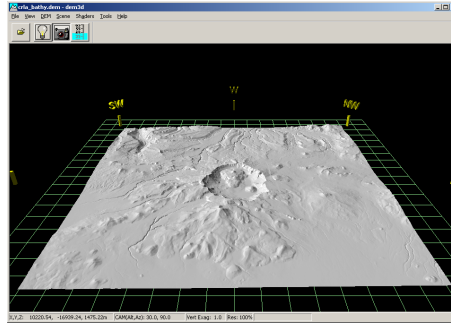


Abbildung 6.5.: Anzeige des USGS-DEM „Craterlake“ mit Himmelsrichtung und Gitter mit dem3D

## 6.4. Lösung mit Grass

Es wurde GRASS 5.0pre1 unter einer Debian GNU/Linux Installation (woody) verwendet.

Um ein digitales Höhenmodell zu importieren, muss eine Umgebung aus passender *Location*, *Mapset* und *Region* geschaffen werden. Dies sollte gleich beim Aufruf von *grass5* erfolgen, da hier am einfachsten alle Parameter eingegeben werden können. Ein nachträgliches Ändern und Anpassen ist zwar mit *g.region* für die Regionsparameter und *g.setproj* für die Projektion möglich, aber die Änderung der Projektion bewirkt hierbei keine Neuprojektion schon vorhandener Daten in der Location.

Es sollten also bekannt sein: die Projektionsart, bei Daten des USGS üblicherweise UTM (Universal Transversal Mercatorprojection), der Meridianstreifen, zusätzlich können noch das verwendete Ellipsoid und das Datum der Koordinatenrealisierung angegeben werden. Sollte die Ausdehnung des DHM nicht bekannt sein so kann man mehr oder weniger mühsam mit einem Editor den Header durchforschen oder man importiert das Modell mittels *m.dem.extract* unter Angabe der Importdatei *input=* und ohne Angabe einer Ausgabedatei *output*. Das Programm wird nun im Examination - Modus ausgeführt. Hierdurch erhält man in der Programmmeldung Angaben über die aktuellen Regionsparameter *Current Region Settings*, und interessanter, über die zu importierende Datei zum Beispiel:

```

GRASS:~ > m.dem.extract input=/home/heiko/g3d/crla_bathy.dem
Current Region Settings-----
rows:      973
cols:      507
north:     5901608.25
south:     5901122.25
east:      3464760.75
west:      3464507.75
ns_res:    0.500000
ew_res:    0.500000

Reading Elevation Tape...
-----
File # 1

min elevation: 946.000000  max elevation: 2721.000000
ns_res: 30.000000  ew_res: 30.000000
# of columns in file = 1374
C-record: 0
file north UTM = 4775095.04759036
file south UTM = 4733028.06587524
file east UTM = 592070.83615184
file west UTM = 550840.83667175
(outside current geographic region)
-----tape min elevation:946  tape max elevation:2721

```

Somit haben wir alle Informationen um das Digitale Höhenmodell zu importieren.

#### 6.4.1. Einlesen des DHM

Nachdem mit *g.region* die Parameter des Mapset angepasst wurden kann der nochmalige Aufruf *m.dem.extract* wiederum mit Angabe des Importpfades *input=* und der zu erzeugenden Rasterdatei *output=* diesmal korrekt bearbeitet werden und man wird mit der Meldung *intersects with current geographic region* belohnt.

Das Anzeigen der Rasterdatei geschieht wieder mit *d.rast* in einem mit *d.mon* geöffneten oder ausgewählten Monitor.

```

GRASS:~ > m.dem.extract input=/test/crla.dem output=rasterdata

```

#### Tipps:

- Beim Importieren von USGS-DEMs kommt es vor das ein oder mehrere meist am Rand gelegene Höhenkoordinaten einen falschen Wert von -32767 Metern aufweisen. Ob die ein Fehler in der Ursprungsdatei, ein Übertragungsfehler oder ähnliches ist kann noch nicht gesagt werden. Man kann nun die Rasterdatei per *r.out.ascii* in eine ASCII-Datei konvertieren und die Werte per Editor berichtigen, oder man benutzt den Befehl *r.mapcalc 'Ergebnisdatei=if(Ausgangsdatei == -32767, null(),*



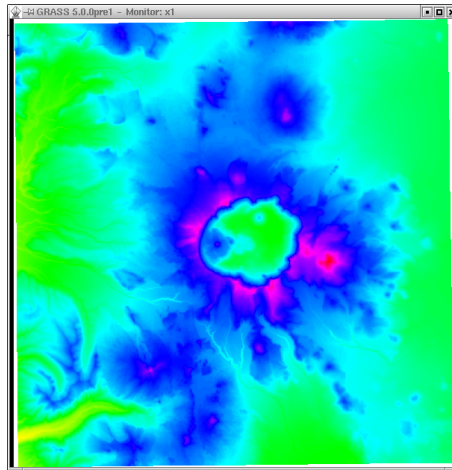


Abbildung 6.6.: GRASS-Ausgabemonitor mit USGS DEM Crater Lake

*Ausgangsdatei*)'. Hierdurch werden alle Werte der Ausgangsdatei die gleich -32767 sind auf null() also nicht vorhanden gesetzt.

- anstatt == können auch andere logische Operatoren benutzt werden, so z.B. kleiner oder gleich <=, größer oder gleich >=, nicht gleich != und so weiter. Eine Liste aller Möglichkeiten findet sich in der Manualpage zu `r.mapcalc` *man r.mapcalc*.

## 6.5. Lösung mit dlgv32 / DLG Viewer

Es wurde der DLG Viewer 3.7 unter einer Windows 2000 Installation verwendet.

Ein weiteres Programm Des USGS ist dlgv32 (Digital Line Graph Viewer). Neben der Möglichkeit USGS Vektor-Dateien (DLGs) anzuzeigen kann es auch USGS ASCII-DEMs und SDTS DEMs verarbeiten. Unter Anderem importiert dlgv32 in der Version 3.7 USGS DEMs, Orthophotos (DOG) und Rastergraphiken(DRG) im GEOTIFF- Format, GTOPO30 Höhendaten und CDED-Dateien (Canadian Digital Elevation Data).

Die Anzeige in dlgv32 alias DLG Viewer kann in *Tools/Control Center/Options* Beeinflusst werden. Informationen über die dargestellten Daten erhält man in der Statusanzeige im unteren Fensterrahmen und unter *Tools/Control Center/Metadaten*. Weiterhin zu erwähnen ist das dlgv32 die Möglichkeit zur Druckausgabe besitzt.

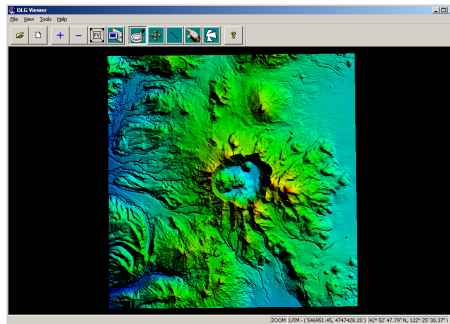


Abbildung 6.7.: Importiertes USGS DEM „Crater Lake“ in dlgv32

# 7. Höhenlinien aus Punktraster erzeugen

## Aufgabenstellung

Das Erzeugen von Höhenlinien ist eine häufig gestellte Anforderung in vielen Bereichen der Vermessung und speziell der Karthographie. Durch den geschickten Einsatz zum Beispiel in Topographischen Karten kann in der 2-Dimensionalen Karte die visuelle Wahrnehmung von Höhen unterstützt werden.

## Beispieldaten

Es werden die in Kapitel 6 Seite 59 ff importierten Daten des USGS verwendet.

## Weitere Verwendung der Beispieldaten

Die hier bearbeiteten Daten werden in den folgend aufgezählten Kapiteln benutzt:

- Kapitel 15 Seite 135 ff.

## 7.1. Lösung mit GRASS

Es wurde GRASS 5.0pre1 unter einer Debian GNU/Linux Installation (woody) verwendet.

### 7.1.1. Erzeugen der Höhenlinien

Zum Erzeugen von Höhenlinien verwendet man den Befehl *r.contur*. Hiermit werden Konturlinien aus einer Rasterdatei erzeugt. Diese Konturlinien werden in einer Datei im GRASS-Vektorformat *output=* gespeichert. Mit *step=* wird die Äquidistanz, also das Intervall zwischen den Konturlinien vorgegeben.

Es können aber auch direkt die auszugebenden Konturlinien benannt werden *levels= , , ,*. Die Auswahl eines Bereiches für den die Höhenlinien erzeugt werden sollen, ist durch die Definition einer Untergrenze *minlevel=* und einer Obergrenze *maxlevel=* möglich.

```

GRASS: > r.contour -n input=crater output=hlinien step=150
Reading data.
Percent complete: 100%
FPRange of data:   min = 0.000000 max = 2721.000000
Minimum level will be 0.000000
Maximum level will be 2700.000000
Continue?(y/n) [y] y
Displacing data.
Percent complete: 100%
Total levels: 19 Current level:

Building topology...

    V.SUPPORT:

Selected information from vector header
Organization:
Map Name:      from raster map crater
Source Date:
Orig. Scale:   0
No snapping will be done

    Reading Vector file.
100%
    Building areas
    Building islands
    Attaching labels
Number of lines: 261
Number of nodes: 262
Number of areas: 0
Number of isles: 0
Number of atts : 261
Number of unattached atts : 0
Snapped lines  : 0

Finished.

```

### 7.1.2. Anzeigen der Höhenlinien

Die als Vektordaten gespeicherten Höhenlinien kann man mit *d.vect* auf den aktiven Monitor ausgeben.

Die anzuzeigende Datei wird mit *map=* ausgewählt. Die Farbe der Höhenlinien wird mit *color=* ausgewählt. Mit den Optionen *-m* wird der Gebrauch von Hauptspeicher eingeschränkt und mit *-v* werden Rückmeldungen des Programms am Bildschirm ausgegeben (*-v* = verbose = Wortreich).

```

GRASS:~ > d.vect -m -v map=hlinien color=white

```

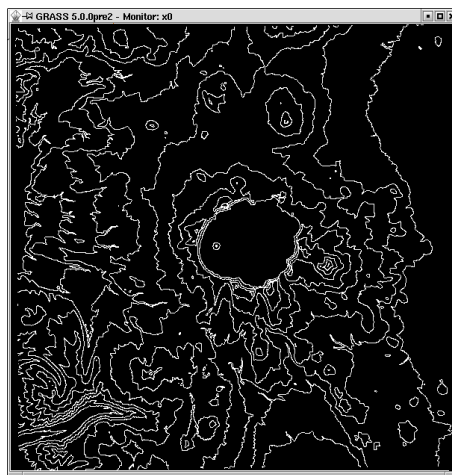


Abbildung 7.1.: Höhenlinien von GRASS aus den Daten des Craterlakes ermittelt



# 8. Überflutungssimulation im Höhenmodell erzeugen

## Aufgabenstellung

Dieser Teil des Tutorials beschäftigt sich mit der Darstellung eines Wasserspiegels beliebigen Pegels in einem Höhenmodell ohne Berücksichtigung von besonderen Abhängigkeiten wie Verbindungen zur Wasserquelle, Wasserscheiden und Ähnlichem. Gerade GRASS stellt in diesem Bereich ausgefeilte Simulationen zur Verfügung.

## Beispieldaten

Grundlage für dieses Kapitel sind die Daten des USGS Crater Lake Data Clearinghouse, die in Kapitel 6 Seite 59 importiert wurden.

## 8.1. Lösung mit GRASS

Es wurde GRASS 5.0pre1 unter einer Debian GNU/Linux Installation (woody) verwendet.

Eine pragmatische Lösung für diese Aufgabe ist es, alle Punkte einer Rasterdatei, deren (Höhen) - Wert unter einer bestimmten Vorgabe liegen auszublenden, also durch eine Maske verdecken. Hierzu bedient man sich des schon in Kapitel 6.4 ab der Seite 59 vorgestellten Programmes *r.mapcalc*, das direkte arithmetische und logische Operationen auf ein Raster zulässt. In diesem Fall genügt ein *r.mapcalc* '*Ergebnisdatei=if(Ausgangsdatei <= 1900, null(), Ausgangsdatei)*'. Hierdurch wird eine neue Ergebnisdatei erzeugt, in der alle Werte der Ausgangsdatei, die kleiner oder gleich 1900 sind, auf Null(), also „nicht existent“ gesetzt werden. Alle anderen Werte (der, „Else-Fall“) werden beibehalten. Eine Anzeige mit *d.rast input=Ergebnisdatei* zeigt das Ergebnis im aktiven Monitor an (Abb. 8.1)

```
GRASS:~ > r.mapcalc 'craterflood=if(crater <= 1900,null(),crater)'
```

Soll die Ergebnisdatei als Maske gebraucht werden so kann man mit dem Aufruf *r.mapcalc* '*Ergebnisdatei=if(Ausgangsdatei <= 1900, null(), 200)*' dem nicht gefluteten Bereichen ein beliebiger Wert (hier 200) zugewiesen oder durch *r.mapcalc* '*Ergebnisdatei=if(Ausgangsdatei <= 1900, 200, null())*' das inverse Ergebnis erzeugt werden (alle Bereiche über 1900 werden ausgeblendet). Der Zahlenwert der Maske ist nicht entschei-

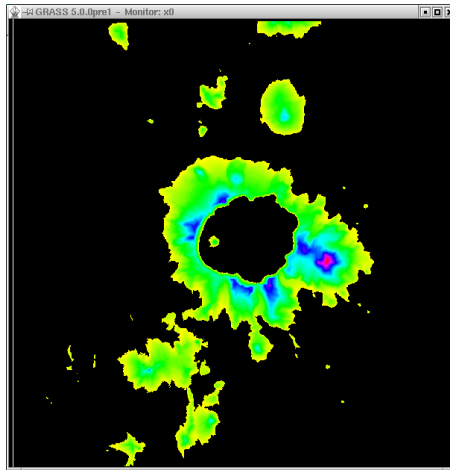


Abbildung 8.1.: Überflutungssimulation „Crater Lake“ Pegelhöhe bei 1900 Metern mit GRASS

dent, da durch GRASS alle Gebiete maskiert werden die nicht *null()* sind.

```
GRASS:~ > r.mapcalc 'cratermask=if(crater <= 1900,200,null())'
```

## 8.2. Lösung mit dem3D

Es wurde die Version 2.0a von dem3D auf einer Windows 2000 Installation verwendet. Hat man das DEM in das Programm dem3D mit *File/Open* eingelesen und durch Bewegen der Maus im Anzeigefenster bei gedrückter linker Maustaste in die gewünschte Lage gebracht, kann man den Wasserstand durch Klicken auf die *Pegeltaste* in der Menüleiste und anschließendes auf- und abbewegen der Maus mit gedrückter linker Maustaste einstellen. Die numerische Eingabe erfolgt im Menue *Shader/Scene/Options/Water*, in dem auch die Transparenz des Wassers bestimmt wird (Abb. 8.2).

## 8.3. Lösung mit dlgv32/ DLG Viewer

Es wurden die Version 3.7 des DLG Viewers auf einer Windows 2000 Installation verwendet.

Denkbar einfach gestaltet sich die Lösung in dem USGS-Programm dlgv32. Nach dem Importieren des DEMs mit *File/Open as New* kann der gewünschte Wasserpegel unter *Tools/Control Center/Options* in der Karteikarte *Vertical Options* mit dem Schieberegler *Water Level* bestimmt werden. Die Einstellung *Water Transparency* kontrolliert den Grad der Transparenz des Wassers. In der Stellung *Opaque* wird das Wasser undurchsichtig, bei *Clear* ganz ausgeblendet.



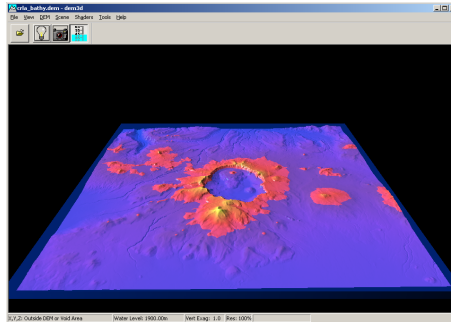


Abbildung 8.2.: Überflutungssimulation „Crater Lake“ Pegelhöhe bei 1900 Metern mit dem3D

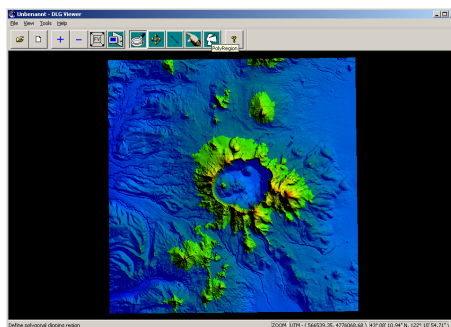


Abbildung 8.3.: Überflutungssimulation „Crater Lake“ Pegelhöhe bei 1900 Metern mit dlvg32



# 9. Rasterdateien georeferenzieren

## Aufgabenstellung

Durch eine Georeferenzierung werden Daten mit einer Information (wir erinnern uns: ein Datum über ein Datum = Metainformation) über ihre Lage in einem bekannten Bezugssystem ausgestattet. Die kann sowohl Raster-, Vektor und Sachdaten betreffen. Hierunter fällt das Geokodieren von Karten, Luft- und Satellitenbildern und allen anderen Raster-Geodaten mittels Koordinaten. Um aussagekräftige Ergebnisse bei Luftbildern zu erhalten, sollten ausschließlich entzerrte Luftbilder, sogenannte Orthophotos verwendet werden. Die Erzeugung von Orthophotos wird in einem weiteren Kapitel besprochen.

## Beispieldaten

Zusätzlich zu den schon im Kapitel 6 Seite 59 importierten Daten werden weitere Datensätze des USGS Crater Lake Data Clearinghouse<sup>1</sup> benutzt. Hier wären das Orthophoto Digital Orthophoto Quadrangle (DOQ) oder genauer ein Mosaic aus 12 7,5-Bogenminutenminuten DOQs und die digitalisierte Topographische Karte als Digital Raster Graphic (DRG) in der selben Ausdehnung zu nennen. Da die Ost-West Ausdehnung einer Bogenminute ihr Maximum am Äquator und ihr Minimum am Pol hat, jedoch die Nord-Süd Ausdehnung immer gleich ist (bei der Vorstellung der Erde als Kugel), sind die DOQs und DRGs hier rechteckig, nicht quadratisch.

## Weitere Verwendung der Beispieldaten

Die ab Seite 71 ff bearbeiteten Daten werden in dem folgenden Kapitel weiter benutzt:

- Kapitel 10 Seite 83 ff.

## 9.1. Lösung mit GRASS

Es wurde GRASS 5.0pre1 unter einer Debian GNU/Linux Installation (woody) verwendet.

Man kann mehrere Fälle unterscheiden, im einfachsten soll eine Rasterdatei in eine GRASS LOCATION importiert werden. Hierbei kann es sich um ein Orthophoto, eine Karte oder ähnliches handeln. Voraussetzung ist, daß die Koordinatenbegrenzung der Datei bekannt ist, daß sie optimal auf das benutzte Koordinatensystem ausgerichtet ist und daß

---

<sup>1</sup><http://craterlake.wr.usgs.gov/>

sie keine internen Verzerrungen aufweist. Diese Voraussetzungen wird man in den seltensten Fällen haben, da z.B. eine Karte auf einem Scanner nur beschränkt richtig, d.h. dem Scannereigenen Koordinatensystem folgend, ausgerichtet werden kann [3].

### 9.1.1. Rasterdatei einlesen

Grass stellt unter anderem Importfilter für TIFF, PPM und PNG -Dateien zur Verfügung.

**TIFF-Format** Der Befehl *r.in.tiff* liest eine 8 bit oder 24 bit - codierte Bilddatei im TIFF (tagged image file format) ein. Die Eingabedatei wird mit *input=* ausgewählt, die Rasterausgabedatei mit *output=* benannt. Die Option *-v* schaltet in den kommunikativen Modus, *-b* importiert 24 bit codierte TrueColor-Dateien. Die Anzahl der möglichen Farben kann mit *nlev=* bestimmt werden. Möglich sind hier Angaben von 1-256 wodurch die Anzahl der Intensitäten in den drei Farbkanälen Rot, Grün und Blau bestimmt wird. Grundeinstellung sind hier 20 Farbwerte also insgesamt  $20 * 20 * 20 = 8000$  verschiedene Farben.

```
GRASS:/spare/gisdata/grass/bin > r.in.tiff -v /  
input=/tmp/test/luftbild.tif output=craterluftbild
```

**PPM-Format** Um eine Portable Pixel Map (ppm) Datei einzulesen, benutzt man den Befehl *r.in.ppm* . Auch hier geben *input=* und *output=* die Namen der Ein- und Ausgabedatei an. Durch *nlev=* wird die Anzahl der möglichen Farben begrenzt, die Voreinstellung ist 20, also wiederum 8000 Farben.

Der Schalter *-v* gibt Statusmeldungen über den Programmverlauf aus, *-b* erzeugt bei Import 3 Grauwertdateien der einzelnen Farbkanäle. Diese werden durch ein an den *output=* Dateinamen jeweils angehängtes *.r*, *.g* und *.b* unterschieden.

```
GRASS:/spare/gisdata/grass/bin > r.in.ppm /  
input=/tmp/test/luftbild.ppm /  
output=craterluft
```

### Achtung !

- Eine eventuell erfolgte Georeferenzierung wird nicht auf die Grauwertdateien übertragen und muß somit nochmals erfolgen.

**PNG-Format** Das Importieren einer Portable Network Graphic (png) geschieht mit dem Befehl *r.in.png*. Mit *input=* und *output=* wird die Ein- und Ausgabedatei gewählt. Der Ausgabedatei kann ein Titel *title=* gegeben werden und das schon bekannte *-v* schaltet in den verbose - Modus.

### 9.1.2. Georeferenzierung einer unverzerrten ausgerichteten Rasterdatei

Im diesem Fall reicht es zur Georeferenzierung aus, die Nord-, Süd-, West- und Ostwerte der Bildkanten an GRASS zu übergeben. Dies geschieht mit dem Befehl *r.support*, da die importierten Bilddateien als Raster gespeichert sind. Nach dem Aufruf an der Kommandozeile fragt der Befehl nach der zu bearbeitenden Rasterdatei. Durch Eingabe von *list* können vorhandene Dateien angezeigt werden. Hat man die gewünschte Datei ausgewählt, fragt das Programm, ob der Header der Datei bearbeitet werden soll; dieses bejaht man. Die Aufforderung *hit RETURN to continue* ignoriert man und drückt die *ESC-Taste* gefolgt von einem *RETURN*. Nun bekommt man Informationen über die Auflösung der Rasterdatei *rows* = Reihen oder Spalten, *cols* = Zeilen und die Farbtiefe in Byte *bytes per cell*. Durch Eingabe von *ESC* und *RETURN* kommt man zu der eigentlichen Georeferenzierung. Der CELL HEADER ist Träger der Geoinformation einer Rasterdatei in GRASS. Zur Orientierung wird die Ausdehnung der Region bzw. des Mapset ausgegeben. Die Bestätigung der Eintragungen erfolgt durch *ESC* gefolgt von *RETURN*. Die weiteren Abfragen können mit *RETURN* bestätigt werden.

```
GRASS:/spare/gisdata/grass/bin > r.support

Enter name of raster file for which you will create/modify support files
Enter 'list' for a list of existing raster files
Enter 'list -f' for a list with titles
Hit RETURN to cancel request
> craterluft
<craterluft>
Edit the header for [craterluft]? (y/n) [y]
Edit header for [craterluft]
cellhd compression: 1
3.0 compression indicated
pre 3.0 compression not indicated

hit RETURN to continue -->
```

```
Please enter the following information for [craterluft]

4207   Number of rows
4123_  Number of cols
2_____ Number of bytes per cell

AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
(OVER <Ctrl-C> TO CANCEL)
```



```

projection:  1 (UTM)
zone:        10
north:       4775095
south:       4733028
east:        592070
west:        550840

e-w res:     10
n-s res:     9.9992869

total rows:  4207
total cols:  4123
total cells: 17,345,461

warning - north falls outside the default region
warning - east falls outside the default region

Do you accept this header? (y/n) [n] >yheader for [craterluft] updated

hit RETURN to continue -->
Update the stats (histogram,range) for [craterluft]? (y/n) [n] y

Edit the category file for [craterluft]? (y/n) [n] n
Create/Update the color table for [craterluft]? (y/n) [n] n
Edit the history file for [craterluft]? (y/n) [n] n
The null file for [craterluft] might indicate that some cells contain
no data. If null file for [craterluft] doesn't exist all
zero cells in [craterluft]
are treated by GRASS application programs as no data.

Do you want to create/reset null file for [craterluft] so that
all cell values are considered valid data? (y/n) [n] n

Do you want to delete null file for [craterluft]
(all zero cells will then be considered no data)? (y/n) [n] n

```

### 9.1.3. Georeferenzierung einer unverzerrten nicht ausgerichteten Datei

Solche Daten ergeben sich oft durch das Scannen. Hierbei ist wichtig, daß die Scanvorlage z.B. eine Karte geometrisch so einwandfrei wie möglich ist und auch durch den Scanvorgang keine unnötigen Verzerrungen auftreten. Sind diese Voraussetzungen gegeben, ist bei der Georeferenzierung nur zu beachten, daß die Vorlage auf dem Scanner nie exakt ausgerichtet werden kann. Die so entstehende Rotation kann durch eine einfache Transformation berichtigt werden.

**Vorbereitung einer Transformation** Zum Ausführen einer Transformation benötigt GRASS zwei *LOCATION* Umgebungen, eine für das Ausgangskordinatensystem *start* und

die andere für das Zielsystem *ziel*, dies ist im Normalfall das "Arbeitssystem" also die schon erstellte *LOCATION*. Für das Ausgangssystem wird eine neue *LOCATION* mit einem *MAPSET* in einem X,Y- Koordinatensystem erstellt. Dies entspricht ja den durch den Scannvorgang erzeugten ebenen Koordinaten. In die Ausgangs - *MAPSET* wird die Rasterdatei je nach Dateiformat z.B. mit *r.in.tiff* eingelesen. Die *REGION* - Parameter des *MAPSET* können mit dem Befehl *g.support rast=Rasterdateiname* angepasst werden.

```
GRASS:/spare/gisdata/grass/bin > g.support rast=craterluft
```

Um das Transformieren und damit gleichzeitig das Geokodieren von mehreren Rasterdateien in einem Schritt zu ermöglichen wird *i.group* aufgerufen. Dies hat auch dann erfolgen, wenn nur eine Datei zu bearbeiten ist. Im folgenden Dialog werden die zu transformierenden Dateien einer Gruppe zugeordnet. Zum definieren des Zielsystems dient der Befehl *i.target*, hier werden über einen Dialog zuerst die Dateien *group* und dann die Ziel-*LOCATION* benannt. Durch die Eingabe von *list* werden alle möglichen Gruppen und Zielorte ausgegeben.

```
GRASS:/spare/gisdata/grass/bin > i.group

LOCATION: start          i.group          MAPSET:start

This program edits imagery groups. You may add data layers to, or remove
data layers from an imagery group. You may also create new groups

Please enter the group to be created/modified

GROUP:   start_____ (list will show available groups)

AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
        (OR <Ctrl-C> TO EXIT)

start - does not exist, do you wish to create a new group?(y/n)[n] y
```



```
LOCATION: start          GROUP: start          MAPSET:start

Please mark an 'x' by the files to be added in group [start]

MAPSET: tutorial

x_ craterluft

                AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
                (OR <Ctrl-C> TO CANCEL)

Group [start] references the following raster file
-----
                craterluft in start
-----
Look ok? (y/n)y
```

```
GRASS:/spare/gisdata/grass/bin > i.target

Please select the target LOCATION and MAPSET for group <start>

CURRENT LOCATION: start
CURRENT MAPSET:   start

TARGET LOCATION: craterlake
TARGET MAPSET:   tutorial

(enter list for a list of locations or mapsets within a location)

AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
                (OR <Ctrl-C> TO CANCEL)
```

**Passpunkte** Um die Transformationsparameter, in diesem Fall für eine Rotation und ggf. eine Maßstabsänderung, bestimmen zu können sind sogenannte Passpunkte zu bestimmen. Im Idealfall nimmt man hier die Eckpunkte der Rasterdatei z.B. einer Karte, für die natürlich Koordinaten im Start- und Zielsystem vorliegen müssen. Es können aber auch Punkte innerhalb der Rasterdatei, wie etwa Gitterkreuze oder andere bekannte Punkte benutzt werden. Diese sollten nach Möglichkeit ein Rechteck aufspannen und flächenhaft d.h. gleichmäßig über den zu transformierenden Bereich verteilt sein. In diesen Fällen reichen 4 Punkte aus um die Transformation mit einer Kontrolle durchzuführen. Sollten keine Punkte bekannt sein, kann man die Transformationsparameter durch Bestimmung identischer Punkte in der Startdatei und einer schon im Zielsystem vorhandenen geokodierten

Datei erfolgen. Als Beispiel könnten in einem Luftbild anhand einer schon vorhandenen georeferenzierten Karte so entsprechende Passpunkte erzeugt werden. Die Bestimmung der Passpunkte erfolgt mit Unterstützung eines Monitores, der mit *d.mon* geöffnet wird. Nach dem Starten des Programmes *i.points* wird die Start - *GROUP* abgefragt. Im Monitor kann jetzt per doppeltem Mausklick im Menü *Mapset import* die zu transformierende Datei ausgewählt werden. Links unten im Monitor befindet sich eine Befehlszeile mit den durch Anklicken auswählbaren Befehlen *Quit* zum Beenden, *ZOOM* für eine Ausschnittsvergrößerung, *PLOT RASTER* um weitere Rasterdateien anzeigen zu lassen und schließlich kann mit *ANALYZE* eine Aufstellung der Einzel- und Gesamtfehler der Passpunkte ausgegeben werden. In dieser Aufstellung können einzelne Passpunkte durch Doppelklicken aus der Transformation entfernt oder hinzugefügt werden. Als Minimum braucht GRASS 4 Punkte um eine Analyse zu erstellen. Nach [3] sollte der Gesamtfehler nicht größer sein als die Auflösung (*GRID RESOLUTION*) des Zielsystems.

```
GRASS:/spare/gisdata/grass/bin >d.mon  
  
GRASS:/spare/gisdata/grass/bin >i.points
```

**Numerische Passpunkteingabe** Als Passpunkt wird per Mauszeiger gegebenenfalls nach Vergrößerung mit *ZOOM* ein Punkt ausgewählt. Im Terminalfenster erscheinen nun die sogenannten "Tisch- oder Bildschirmkoordinaten". Die Aufforderung *Enter coordinates as east north:* beantworten wir mit der Eingabe der Koordinaten des Punktes im Zielsystem (Rechtswert Hochwert getrennt durch ein Leerzeichen). Sind per *ANALYZE* Passpunkte entfernt worden so werden sie durch einen roten Punkt symbolisiert. Die für die Transformation benutzten Punkte werden als grüner Punkt angezeigt. Sind die ausgegebenen Differenzen weiterhin zu groß sollten ggf. die Passpunkte in einem stärker vergrößertem Ausschnitt ausgewählt werden.

**Passpunktidentifikation** Sollen anstatt vorhandener Passpunkte geokodierte Rasterdateien zur Identifikation von Passpunkten benutzt werden, ruft man die fragliche Datei mit *PLOT RASTER* auf (Abb. 9.1). Die Aufforderung *Indicate which side should be plotted* wird mit einem Mausklick auf die rechte Monitorhälfte beantwortet. Im folgenden Menü werden alle im Zielmapset vorhandenen Dateien angezeigt. Die Auswahl geschieht durch einen Doppelklick. Auch hier können mit *ANALYZE* die Passpunkte überprüft werden (Abb. 9.2).

**Transformation** Die eigentliche Transformation geschieht durch den Befehl *i.rectify*. Nach der Angabe der zu transformierenden Dateigruppe fragt GRASS nach dem Grad der Transformation. Mögliche Eingaben sind hier 1, 2 oder 3 für den Grad des von GRASS aufzustellenden Transformationspolynomes. In unserem Fall einer Translation, Rotation und Skalierung gleichen Faktors in x und y-Richtung ist eine 1 einzugeben.

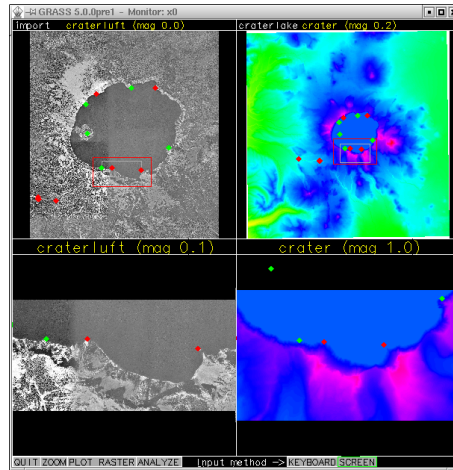


Abbildung 9.1.: GRASS-Monitor während einer Georeferenzierung

#### 9.1.4. Georeferenzierung einer verzerrten unausgerichteten Datei

Soll beispielsweise eine Rasterdatei, deren Scanvorlage von geringer geometrischer Qualität ist georeferenziert werden, so kann GRASS dies durch eine höhergradige Transformation ausführen. Hierbei werden nicht nur Rotationen, Verschiebungen (Translationen) und gemeinsame Maßstabsänderungen für alle Koordinatenrichtungen berücksichtigt, sondern auch Veränderungen in den Winkelbeziehungen der Achsen, verschieden große Maßstabsänderungen, Verschiebungen der einzelnen Achsen, sowie verschiedene Bezugssysteme für die Höhe in verschiedenen Koordinatensystemen. In diesen Fällen müssen mindestens 6 (2. Grad) oder 10 (3. Grad) Passpunkte eingegeben oder identifiziert werden und an *i.rectify2* wird als Grad der Transformation 3 übergeben.

```
GRASS:/spare/gisdata/grass/bin > i.rectify2
```

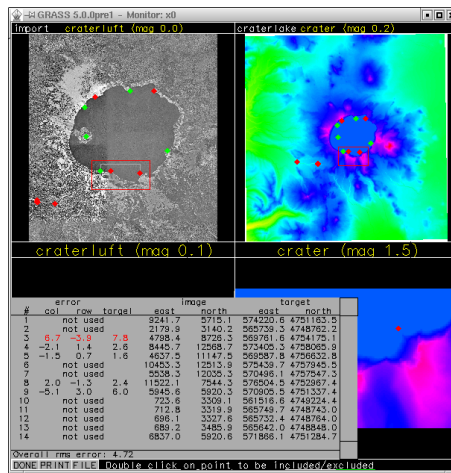


Abbildung 9.2.: Analyse der Passpunkte

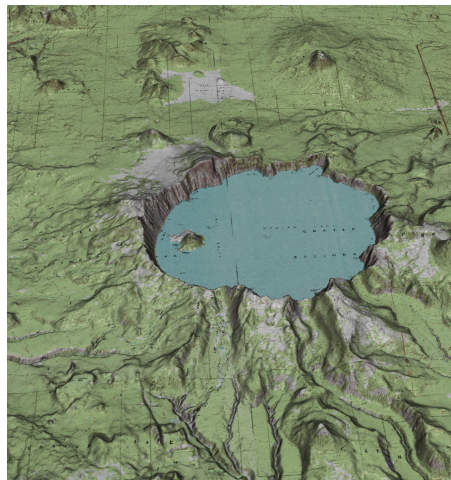


Abbildung 9.3.: Ergebnis einer Georeferenzierung Karte und DEM im GRASS-Modul *Nviz2.2*

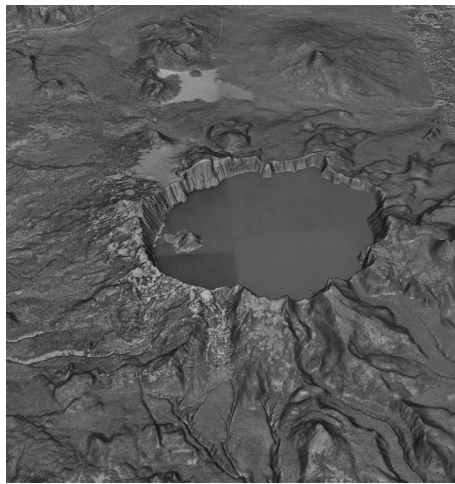


Abbildung 9.4.: Ergebnis einer Georeferenzierung Luftbild und DEM im GRASS-Modul  
*Nviz2.2*



# 10. Volumenberechnung aus Rasterdaten

## Aufgabenstellung

Ein schneller und für den Anwender gut nachzuvollziehender Weg Volumen aus Rasterdaten zu ermitteln eröffnet sich wenn eine Grundfläche des zu berechnenden Körpers eben ist. Sollte dies nicht gegeben sein, teilt man den Körper durch einen ebenen Schnitt und führt 2 Berechnungen aus, die man zum Ergebnis addiert. Im gegebenen Fall „Craterlake“ gib es 2 Dateien, eine mit Angaben über den Kraterboden, die aus Echolotdaten zusammengestellt wurden und eine Datei, die bei der Wasseroberfläche abschließt. Da die Wasseroberfläche mit dem Grund des Kraters einen Körper bildet, kann das Volumen ermittelt werden.

## Beispieldaten

Für die Bearbeitung der gestellten Aufgabe wird ein Höhenmodell des Craterlakes mit eingearbeiteten Echolotdaten des Seebodens (10-m DEM with bathymetry)<sup>1</sup> benutzt, das wiederum vom USGS Crater Lake Data Clearinghouse <sup>2</sup> zur Verfügung gestellt wird.

## 10.1. Lösung mit GRASS

Es wurde GRASS 5.0pre1 unter einer Debian GNU/Linux Installation (woody) verwendet.

Grass bedient sich zur Lösung dieser Aufgabe eines integrativen Verfahrens, d.h. es „zählt“ die Zellen in jeder Ebene über bzw. unter jedem einzelnen Rasterelement. Die Genauigkeit kann begrenzt durch die Ausgangsdaten über die in *g.region* angegebene Auflösung beeinflusst werden. Der Wasserstand im Krater ist natürlich schwankend wird aber auf <sup>3</sup> mit 1883 m über dem Meeresspiegel angegeben. Zusammen mit den Echolotdaten des Kratergrundes kann somit eine Volumenberechnung erfolgen.

### 10.1.1. Maskierung der Wasseroberfläche

Um die Volumenberechnung auf den Krater unterhalb der Wasseroberfläche zu beschränken, wird der übrige Teil der Rasterdatei durch eine Maske ausgeblendet. Zum Erstellen

---

<sup>1</sup>[http://craterlake.wr.usgs.gov/products/dem/crla\\_bathy.zip](http://craterlake.wr.usgs.gov/products/dem/crla_bathy.zip)

<sup>2</sup><http://craterlake.wr.usgs.gov/>

<sup>3</sup><http://craterlake.wr.usgs.gov/facts.html>



Abbildung 10.1.: Maske aller Gebiete mit Höhen kleiner oder gleich 1883 m

verwendet man den Befehl *mapcalc* und beschränkt die Anzeige auf Höhen kleiner oder gleich 1883m.

```
GRASS:~ > r.mapcalc 'mask=if((crater1 <= 1883),1,null())'
EXECUTING mask = ... 100%
CREATING SUPPORT FILES FOR ergebnis
range: 1 1
```

Hierdurch wird eine neue Rasterdatei mit dem Rasterwert 1 für jede Rasterzelle der Datei *crater1* angelegt deren, Wert kleiner oder gleich 1883 ist. Die so deutlich hervortretenden Umrise des mit Wasser gefüllten Kraters müssen aber noch einmal per Hand maskiert werden, um alle anderen unter 1883 m liegenden Gebiete auszublenden. Dies geschieht, indem erst ein Vektorpolygon um den zu maskierenden Teil gelegt wird, und dieser in einem zweiten Schritt in eine Rasterdatei umgewandelt wird, die dann als Maske benutzt werden kann.

### 10.1.2. Digitalisierung am Bildschirm

Im folgenden, durch *v.digit* aufgerufene Bildschirmmenü wird 3 ausgewählt, da wir eine sogenannte Bildschirmdigitalisierung vornehmen wollen. Die neu zu erstellende Vektordatei wird in diesem Beispiel *mask* benannt.



```

Available Digitizers

Name          Description
----          -
[1] Calcomp   Calcomp digitizer, format 23 (binary)
[2] Altek     Altek digitizer, model AC30, format 8 (binary)
[3] none      Run digit without the digitizer.

Hit return to use digitizer in brackets
or type in number or name of other digitizer.

Select digitizer [none] : 3

Selected digitizer is: none

Enter the name of a map to work with.
If name is entered that does not already exist, it
will be created at this time.

DIGIT FILENAME
Enter 'list' for a list of existing digit files
Hit RETURN to cancel request
> mask
<mask>
You requested to create new file: 'mask'. Is this correct? (y/n) [n] y

Current mapset is UTM.
Is this map in UTM meters? (y/n) [y]

```

Um Informationen zu der erzeugten Vektordatei speichern zu können, müssen im folgenden Menü Angaben gemacht werden.

```

Provide the following information:

Your organization _____
Todays date (mon,yr) 09,2001_____
Your name            heiko_____
Map's name           mask_____
Map's date           _____
Map's scale          1:1_____
Other info           _____
Zone                 10____
West edge of area    550840_____
South edge of area   4733028_____
East edge of area    592070_____
North edge of area   4775095_____

AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE

```

Die Frage *Shall we continue?* [y] wird bestätigt.

**Das Modul *v.digit*** Das Modul *v.digit* besteht aus mehreren Menüs, die durch die Eingabe von Buchstaben angewählt werden. Hierbei ist auf Gross- und Kleinschreibung zu achten. Das Digitalisieren geschieht mit der Maus am Bildschirm. Hierzu ist natürlich wieder ein Monitor nötig der mit *d.mon* zu starten ist. Das Startmenü sieht wie folgt aus:

```

l-----k
| GRASS-DIGIT Modified  4.10                               Main menu
|-----|
| MAP INFORMATION          | AMOUNT DIGITIZED
| Name:      mask         | # Lines:      0
| Scale:     1            | # Area edges: 0
| Person:    heiko        | # Sites:      0
| Dig. Thresh.: 0.0300 in. | - - - - -
| Map Thresh.: 0.0008 meters | Total points: 0
|-----|
| OPTIONS:
| Digitizer: Disabled
|-----|
m-----j
| Digitize Edit Label Customize Toolbox Window
|      Help Zoom Quit                               * ! ^
|-----|
| GLOBAL MENU: Press first letter of desired
|               command.[Upper Case Only]
m-----j

```

Um die Bildschirmdigitalisierung durchzuführen, bringen wir die schon erzeugte Rasterdatei *mask* auf den Monitor. Dies geschieht im *Customize Menü*, das durch ein großes *C* aufgerufen wird.

```

l-----k
| GRASS-DIGIT Modified  4.10                               Customize Menu |
|-----|
| Customize options:           Current:                    |
| d - Set digitizing threshold  0.0300                   |
| s - Set snapping threshold    0.0300                   |
| F - Toggle flexible grey scale OFF                     |
| b - Toggle BEEP                ON                       |
| a - Toggle Auto Window         OFF                     |
|   - Toggle Windowing device    MOUSE                  |
|   - Toggle Point device        MOUSE                  |
|   - Toggle Digitizing device   MOUSE                  |
| O - Select An Overlay vector Map None                  |
| B - Select A Backdrop CELL Map None                    |
| D - Enter Display Options Menu REMOVE                 |
| C - Enter Color Options Menu                                     |
| m - Toggle Auto-smoothing     OFF                     |
| q - Return from whence we came                               |
m-----j
| Help  Zoom  *  !  ^                                       |
| GLOBAL MENU: Press first letter of desired command.     |
| [[Upper Case Only]                                       |
m-----j

```

Das Hinterlegen einer Rasterdatei geschieht durch den Menüpunkt *B*. Nachdem die entsprechende Datei ausgewählt wurde, wird sie im Monitor dargestellt. Die Eingabe eines kleinen *q* bringt das Anfangsmenü wieder auf den Bildschirm.

```

Enter the name of an existing cell file
Enter 'list' for a list of existing cell files
Hit RETURN to cancel request
> mask
<mask>
Do you want to automatically redraw backdrop on re-window? (y/n) [y]

```

Das eigentliche Digitalisieren wird im Menü *D* (*Digitize*) durch Drücken der *Spacetaste* gestartet. In dem nun gestarteten Menü wählt man mit *t* den Ergebnistyp des Digitalisiervorganges aus, in unserem Fall *AREA EDGE*. Der Linie kann Automatisch ein Attributswert zugewiesen werden *Auto Label*, Taste *l*, bei diesem Beispiel lassen wir die Ausgangsstellung *DISABLED* bestehen.



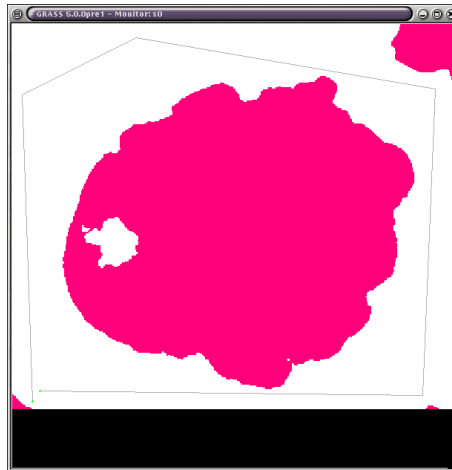


Abbildung 10.2.: Polygon zur Erstellung einer Maske

```
Line to snap FROM:

Buttons:
  Left:  Choose line
  Middle: Accept chosen line
  Right: Abort/Quit

Node to snap TO:

Buttons:
  Left:  Choose node
  Middle: Accept chosen node
  Right: Abort/Quit

node#: 2

EAST:  575524.65
NORTH: 4750334.93

Attempting to snap line to itself. Proceed? [y]
```

Ist die Linie ausgewählt und mit mittlerer Maustaste bestätigt wird auf dieselbe Weise ein „offener“ Endknoten angewählt und wiederum bestätigt. Die Eingabe von *Proceed? [y]* löst die Fangfunktion aus. Als Ergebniss erhält man ein geschlossenes Polygon (Abb. 10.4). Um das Polygon erfolgreich in eine Maske umwandeln zu können, muß seinem Inhalt ein Attribut zugewiesen werden. Unter dem Menüpunkt *Label* des *Main menu* findet sich die Option *Label Areas* die durch *a* angewählt wird.

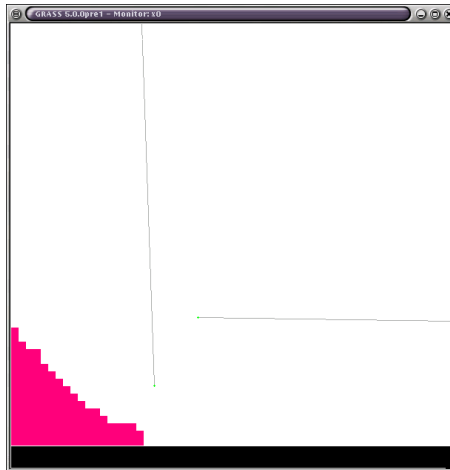


Abbildung 10.3.: Vergrößerung des noch ungeschlossenen Polygons

```

l-----k
| GRASS-DIGIT Modified  4.10                               Label Menu |
|-----|
| Label options:                                          |
| a - Label Areas          m - Label Multiple Lines     |
| l - Label Lines         M - Un-Label Multiple Lines   |
| s - Label Sites                                                |
| A - Un-Label Areas      c - Label Contours            |
| L - Un-Label Lines     i - Contour interval:         < 5> |
| S - Un-Label Sites                                           |
| B - Bulk Label Remaining Lines                             |
| h - Highlight Lines of category #                       |
| d - Display Areas of category #                         |
| q - Return to main menu                                   |
|-----|
m-----j
| Digitize Edit Customize Toolbox Window Help Zoom * ! ^  |
|-----|
| GLOBAL MENU: Press first letter of desired command.     |
| [Upper Case Only]                                       |
m-----j
  
```

Die folgende Aufforderung *Do you wish to enter area names?* [y] kann mit y beantwortet werden. Als *Category Number* sollte eine Zahl gewählt werden, die nicht mit den vorkommenden Höhen verwechselt wird, z.B. 5, als Label *mask*. Dieses *Label* kann aber auch ein Attribut für die *Areas* sein.

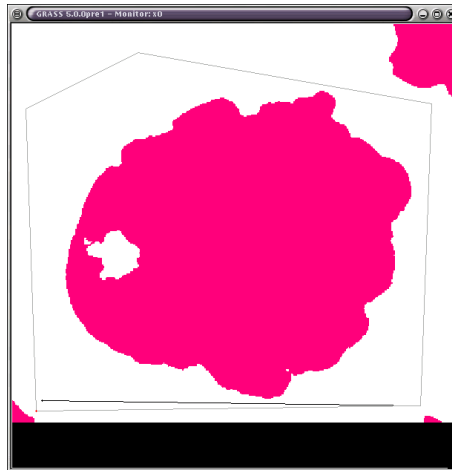


Abbildung 10.4.: Geschlossenes Polygon für die Maskenerstellung

```

l-----k
|
|
|   Enter Category Number (0 to quit):[0] 5
|-----j
l-----k
|
|   Enter a label (<CR> to quit): mask
|-----j
l-----k
|
|   Add new category <5>, named <mask> ? [y] y
|-----j
m-----j

```

Zum Definieren der *AREA* wird ein Punkt innerhalb des Polygons und die Polygonkante mit der linken Maustaste ausgewählt und mit der rechten bestätigt.

Das Digitalisierungs-Modul wird durch ein *Q* (Quit) im *Main menü* beendet. Die Frage *Leave digit [n]* wird mit Ja [*y*] und *compress [n]* mit Nein [*n*] beantwortet.

### 10.1.3. Erzeugung einer Rastermaske aus der Vektordatei

Die Umwandlung der Vektordatei in eine Rastermaske geschieht durch das Kommando *v.to.rast*. Um die Ausgangsmaske *mask* nicht zu überschreiben wird als Name für die Rasterdatei *mask2* gewählt.

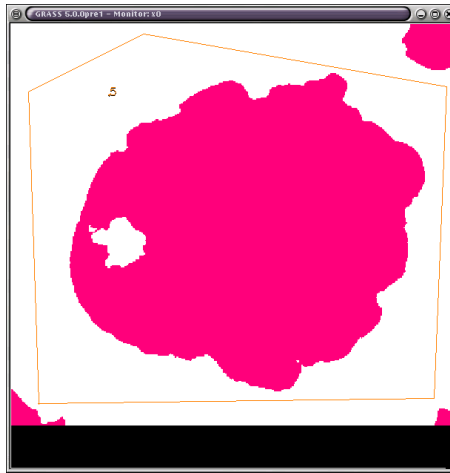


Abbildung 10.5.: Erstellte AREA in *v.digit*

```
GRASS:~ > v.to.rast

OPTION:   vector input file
         key: input
         required: YES

Enter the name of an existing vector file
Enter 'list' for a list of existing vector files
Hit RETURN to cancel request
> mask
<mask>

OPTION:   raster output file
         key: output
         required: YES

Enter a new raster file name
Enter 'list' for a list of existing raster files
Hit RETURN to cancel request
> mask2
<mask2>
```



```

OPTION:  number of rows to hold in memory
         key: rows
         default: 512
         required: NO
enter option >
Loading vector information ...           0 mins 00 secs
Sorting areas by size ... 1 areas       0 mins 00 secs
Pass #1 (of 3)
  Processing areas ... 1 areas           0 mins 00 secs
  Processing lines ... 0 lines           0 mins 00 secs
  Processing sites ... 0 sites           0 mins 00 secs
  Writing raster map ...                 0 mins 01 sec
Pass #2 (of 3)
  Processing areas ... 1 areas           0 mins 00 secs
  Writing raster map ...                 0 mins 00 secs
Pass #3 (of 3)
  Processing areas ... 1 areas           0 mins 00 secs
  Writing raster map ...                 0 mins 00 secs
Creating support files for raster map ... 0 mins 00 secs

Raster map <mask2> done.
Total processing time:      0 mins 01 sec

```

Jetzt kehren wir zum Anfang der Aufgabe zurück und erstellen eine neue Maske, die nur die Wasseroberfläche in 1883 m Höhe über dem Meeresspiegel bedeckt. Zuerst benutzen wir die gerade erzeugte *mask2* um in *mapcalc* nur die Wasseroberfläche zu berücksichtigen. Dazu rufen wir *r.mask* auf.

```

GRASS:~ > r.mask

MASK:  Program for managing current GIS mask

current mask:          none

Options:
  1      Remove the current mask
  2      Identify a new mask
RETURN  Exit program

> 2

Enter name of data layer to be used for mask
Enter 'list' for a list of existing raster files
Enter 'list -f' for a list with titles
Hit RETURN to cancel request
> mask2

```

Im folgenden Menü können den verschiedenen Ebenen der Rasterdatei numerische Werte zugewiesen werden. Der Einfachheit halber wird der 5. Ebene *mask* der Wert 5 zugeteilt.

```

IDENTIFY THOSE CATEGORIES TO BE INCLUDED IN THE MASK

OLD CATEGORY NAME                                CAT
                                                NUM
. . . . .                                0    0___
. . . . .                                1    0___
. . . . .                                2    0___
. . . . .                                3    0___
. . . . .                                4    0___
mask . . . . .                            5    5___
. . . . .                                6    0___

Next category: end__ (0 thru 6)

```

Wird solchermaßen „geschützt“ nocheinmal der *mapcalc* Befehl für die Rasterdatei *crater1* aufgerufen erhält man die gewünschte Wasseroberfläche als Rasterdatei *surface*.

```

GRASS:~ > r.mapcalc 'surface=if((crater1 <= 1883),1,null())'

EXECUTING mask = ... 100%
CREATING SUPPORT FILES FOR ergebnis
range: 1 1

```

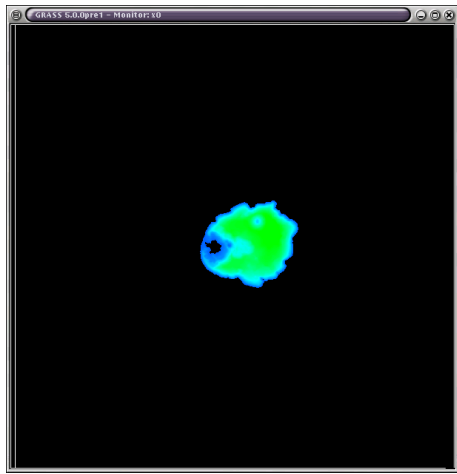


Abbildung 10.6.: Maskierte Wasseroberfläche des Kraters

Die Datei *surface* wird nun als neue Maske eingesetzt, nachdem die alte *mask2* verworfen wurde.

```

GRASS:~ > r.mask

MASK:  Program for managing current GIS mask

current mask:          <mask2@tutorial> in mapset <tutorial>
masking category(ies): 5

Options:
  1      Remove the current mask
  2      Identify a new mask
RETURN  Exit program

> 1

MASK:  Program for managing current GIS mask

current mask:          none

Options:
  1      Remove the current mask
  2      Identify a new mask
RETURN  Exit program

> 2

Enter name of data layer to be used for mask
Enter 'list' for a list of existing raster files
Enter 'list -f' for a list with titles
Hit RETURN to cancel request
> surface

IDENTIFY THOSE CATEGORIES TO BE INCLUDED IN THE MASK

OLD CATEGORY NAME                                     CAT
                                                    NUM
. . . . . 0  0____
. . . . . 1  1____

Next category: end__ (0 thru 1)

AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
                (OR <Ctrl-C> TO CANCEL)

```

Ein Aufruf der gesamten so maskierten Rasterdatei *d.rast -o crater1* erzeugt nur noch die Wasserfläche im GRASS-Monitor (Abb. 10.6). Um die in *crater1* gespeicherten Höheninformationen auf die Wasseroberfläche zurückzuführen, ziehen wir von der Höhe der Wasseroberfläche (1883 m) über dem Meeresspiegel jede Höhenangabe in jeder Rasterzelle

der Datei *crater1* ab. Sollten sich die Höhenangaben in Bereichen unter dem Meeresspiegel bewegen oder benutzt man eine andere lokale Referenzhöhe so zieht man natürlich die Höhenangaben des Seegrundes von der des Wasserspiegels ab. Das Ergebniss bleibt gleich nur das Vorzeichen ändert sich.

```
r.mapcalc volumen="1883 - crater1"

EXECUTING volumen = ... 100%
CREATING SUPPORT FILES FOR volumen
range: 0 585
```

Das eigentliche Volumen wird mit dem Befehl *r.volume* ermittelt.

```
GRASS:~ > r.volume volumen
Complete ... 100%
Volume report on data from volumen using clumps on MASK map

  Cat   Average Data # Cells      Centroid      Total
Number in clump Total in clump Easting  Northing  Volume

  1     329.32 19615517 59564 573090.40 4754376.55 17661187006.99
                                     Total Volume = 17661187006.99
```

## Ergebniss

Das Ergebniss ist in diesem Fall 17.661.187.007 Kubikeinheiten, also Kubikmeter Wasser. Aufgrund der Lagekoordinaten mit 30m Auflösung, der jahreszeitlichen Schwankungen des Wasserspiegels und der gänzlich unbekanntem Genauigkeit der Tiefenmessung ist dieser Wert nur mit Einschränkungen weiterzuverwenden.

# 11. Mosaik aus Luftbildern erstellen

## Aufgabenstellung

Da Luftbildflüge streifenweise mit großer Überlappung der einzelnen Aufnahmen durchgeführt werden ist die Erstellung eines Mosaiks aus den einzelnen Aufnahmen eine interessante Aufgabe. In diesem Fall haben wir 10 Luftbilder, die durch vorhandene Passpunkte direkt geokodiert werden können. Obwohl durch den Einsatz moderner Kamera- und Navigationstechnik die mittleren Standardabweichungen der Fluglage und Richtung unter 0,01 Gon liegen, lassen sich die unbearbeiteten Aufnahmen nicht ohne Ungenauigkeiten zusammenfügen. Dies liegt daran, daß die Luftaufnahmen noch nicht in eine Ebene projiziert sind. Durch diese sogenannte Orthophotoherstellung lassen sich Luftbilder so überarbeiten, daß sie wie eine Karte behandelt werden können. Es kann in ihnen gemessen werden und durch die hohe Informationsdichte des Orthophotos ergeben sich vielfältige Möglichkeiten der Datengewinnung. Die hier beschriebene Erstellung eines Luftbildmosaiks soll ein Gefühl für die Arbeit mit Luftbildern vermitteln, besonders dem Identifizieren von identischen Punkten aus verschiedenen Blickwinkeln sollte besondere Aufmerksamkeit gewidmet werden.

## Beispieldaten

Die verwendeten Luftbilder liegen als Tiff Dateien 174.tif - 183.tif vor. Jede Datei hat eine Größe von 360 MB. Diese Daten wurden von der RWE Rheinbraun AG Abteilung BT3 - Photogrammetrie zur Verfügung gestellt. Diese Daten wurden den Vorgaben dieses Tutorials entsprechend unter die GPL (General Public License) gestellt.

## 11.1. Lösung mit Grass

Es wurde GRASS 5.0pre2 unter einer Debian GNU/Linux Installation (woody) verwendet.

Um mit GRASS das gewünschte Luftbildmosaik zu erstellen, können mehrer Wege gewählt werden. Es könnte jedes Bild für sich erst georeferenziert werden um zum Schluß alle zusammenzufügen. Da sich bei dem vorliegenden Datenmaterial nicht in jedem Luftbild ausreichend markierte Passpunkte in der Örtlichkeit befinden, erstellt man alternativ eine *x-y Location* mit einem *Mapset* für jedes Luftbild. Anschließend werden die untereinander liegenden Bilder je in ein gemeinsames *Mapset* transformiert, so das sich die 5 Einzelteile für die erste Transformationsreihe ergeben. Von diesen 5 Einzelteilen wird das mittlere, Nummer 3 sozusagen, als Zielsystem für die zweite Transformationsreihe bestimmt. Die 4

übrigen Mosaik werden von innen also 2,4 nach 3 und 1,5 nach 3 in das Zielsystem transformiert. Auch bei genauestem Markieren der identischen Passpunkte in den Luftbildern werden sich wegen der fehlenden Entzerrung Spannungen, die sogenannte „Klaffen“ bei der Ermittlung der Transformationsparameter nicht vermeiden lassen. Als Grad der Transformation wird 1 gewählt, also eine Translation, Rotation und Skalierung gleichen Faktors in x und y-Richtung.

## 11.2. Vorbereitungen

Um alle 10 Luftbilder zu einem Mosaik zusammenzufügen, wird nach dem Start von GRASS eine *Location* mit einem *x-y* Koordinatensystem erstellt z.B. *import*. Die Ausdehnung sollte so großzügig gewählt werden, daß das gesamte endgültige Mosaik Platz findet. Ein nachträgliches Ändern der Regionsparameter ist jedoch jederzeit möglich. Die Bildgröße eines Luftbildes beträgt ca. 12000 \* 12000 Pixel, ausgehend hiervon sollte eine Nord-Süd Ausdehnung von +13000 bis -15000 mehr als ausreichend sein, da die Photos sich ja zu einem guten Teil überdecken. In Ost-West Richtung ist man mit -25000 bis +25000 auf der sicheren Seite. Innerhalb dieser *Location* werden nun 10 *Mapsets* erzeugt. Der Einfachheit halber können diese z.B. nach der Numerierung der Luftbilder (175-182) benannt werden. Nach erfolgter Transformation sollten diese Grenzen natürlich angepasst werden, um die Dateigröße zu begrenzen. Bei der Benutzung der Photos in voller Auflösung ergeben sich je nach Rechner pro Transformation schon mal Rechenzeiten im Bereich von Stunden. Es ist also eine Überlegung wert, die Transformationsparameter im Voraus für mehrere Bilder zu bestimmen und die eigentlichen Transformationen *i.rectify* über Nacht ablaufen zu lassen. In jede der neu erstellten *Mapsets* wird nun die zugehörige Luftbilddatei importiert. Dazu wird beispielsweise das *Mapset 176* geöffnet und bei dem ersten Aufruf angelegt.

GRASS 5.0.0pre2

LOCATION: This is the name of an available geographic location.  
-spearfish- is the sample data base for which all  
tutorials are written.

MAPSET: Every GRASS session runs under the name of a MAPSET.  
Associated with each MAPSET is a rectangular  
COORDINATE REGION and a list of any new maps created.

DATABASE: This is the unix directory containing the geographic  
databases

The REGION defaults to the entire area of the chosen  
LOCATION. You may change it later with the command:  
g.region

-----  
LOCATION: import\_\_\_\_\_ (enter list for a list of locations)  
MAPSET: 176\_\_\_\_\_ (or mapsets within a location)

DATABASE: /spare/gisdata/rheinbraun/\_\_\_\_\_

AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE  
(OR <Ctrl-C> TO CANCEL)

Mapset <<176>> is not available

Mapsets in location <import>

-----  
(+)PERMANENT

note: you only have access to mapsets marked with (+)  
-----

Would you like to create < 176 > as a new mapset? (y/n) y

Ist so das *Mapset* angelegt, kann durch den Befehl *r.in.tiff -v input=/spare/gisdata/rheinbraun/176.tif output=176* das Luftbild importiert werden.

Welcome to GRASS 5.0.0pre2 (September 2001)

Geographic Resources Analysis Support System (GRASS) is Copyright, 1999-2001 by the GRASS Development Team, and licensed under terms of the GNU General Public License (GPL).

This GRASS 5.0.0pre2 release is coordinated and produced by the GRASS Development Team headquartered at ITC-irst (Trento, Italy) with worldwide support and further development sites located at Baylor University and the University of Illinois (U.S.A.).

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

This version running thru the Bash Shell (/bin/bash)  
Help is available with the command: g.help  
See the licence terms with: g.version -c  
Start the graphical user interface with: tcltkgrass&  
When ready to quit enter: exit

```
GRASS:> r.in.tiff -v input=176.tif output=176
```

TIFF Directory at offset 0x16678646

Image Width: 11194 Image Length: 11193 Image Depth: 1

Bits/Sample: 8

Compression Scheme: None

Photometric Interpretation: RGB color

Software: "TLD2TIFF, Z/I Imaging GmbH, Oberkochen, Germany"

Orientation: row 0 top, col 0 lhs

Samples/Pixel: 3

Rows/Strip: 5

Planar Configuration: single image plane

Looking for TIFF World file ... Not Found!

11194x11193x24 image

8 bits/sample, 3 samples/pixel

Total colors = 948082

WARNING: Color levels quantization...

Total used colors = 8000

Creating SUPPORT Files for 176

done.

Dieses Importieren wird in allen *Mapsets* für die jeweiligen Luftbilder ausgeführt.

### 11.3. Transformationsparameter bestimmen

Beispielhaft wird jetzt die Bestimmung der Transformationsparameter für den ersten „Block“ bestimmt. Um die Spannungen zwischen den einzelnen Blöcken so klein wie möglich zu halten, wird als Ursprung und endgültiges Zielsystem aller folgenden Transformationen ein Luftbild gewählt, das in der Mitte einer der beiden Befliegungstreifen (175, 176, 177, 178) und (179, 180, 181, 182) liegt. Ist das Zielsystem also das Photo 181.tif so wird GRASS für die *Location: import* mit dem *Mapset: 176* aufgerufen. Für das zu transformierende Photo



wird nun eine Gruppe mit *i.group* gebildet. Bei der Benennung der Gruppe hat man freie Wahl, doch bei umfangreichen Transformationen ist es hilfreich, eindeutige Bezeichnungen zu wählen, in unserem Fall wird die Gruppe nach dem Bild 176 genannt. Die dieser *Group* beigefügte Datei ist natürlich 176. Nach der abschließenden Bestätigung ist die *Group 176* erstellt.

```
GRASS: > i.group

LOCATION: import          i.group          MAPSET: 176

This program edits imagery groups. You may add data layers to,
or remove data layers from an imagery group.
You may also create new groups

Please enter the group to be created/modified

GROUP:    176_____ (list will show available groups)

          AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
          (OR <Ctrl-C> TO EXIT)

176-does not exist, do you wish to create a new group?(y/n) [n] y

LOCATION: import          GROUP: 176          MAPSET: 176

Please mark an 'x' by the files to be added in group [176]

MAPSET: 176

x_ 176

          AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
          (OR <Ctrl-C> TO CANCEL)

Group [176] references the following raster file
-----
          176 in 176
-----
Look ok? (y/n) y

Group 176 created!
```

Dieser Gruppe muß jetzt ein Zielsystem für die Transformation zugewiesen werden. Dazu dient der Befehl *i.target*. Als Ziel *Mapset* wird 181 ausgewählt da dieses Luftbild in der Reihe über 176 liegt. Das Bild 181 wird damit das Zielsystem dieser Transformation.

```

GRASS:> i.target
This program targets an imagery group to a GRASS database

Enter group that needs a target
Enter 'list' for a list of existing imagery groups
Enter 'list -f' for a verbose listing
Hit RETURN to cancel request
>176

Please select the target LOCATION and MAPSET for group <176>

CURRENT LOCATION: import
CURRENT MAPSET: 176

TARGET LOCATION: import_____
TARGET MAPSET: 181_____

(enter list for a list of locations or mapsets within a location)

AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

group [176] targeted for location [import], mapset [181]

```

Zur Bestimmung der Transformationsparameter wird das schon in Kapitel 9.1.3 Seite 77 besprochene Modul *i.points* aufgerufen, nachdem vorher mit *d.mon* ein GRASS-Monitor gestartet wurde.

```

GRASS:/spare/gisdata/grass/bin > i.points

Enter imagery group to be registered
Enter 'list' for a list of existing imagery groups
Enter 'list -f' for a verbose listing
Hit RETURN to cancel request
>176

```

Im Grass-Monitor kann jetzt die im *Mapset* vorhandene Datei 176 mit einem Doppelklick aufgerufen werden. Anschließend wird mit einem Mausklick auf *PLOT RASTER* die im *Ziel-Mapset* liegende Datei 181 in dem vorher angewählten Viertel des GRASS-Monitors angezeigt. Durch Vergrößern identischer Bildausschnitte in beiden Luftbildern durch die *ZOOM*-Funktion können identische Punkte bestimmt werden. Es ist darauf zu achten, daß diese Punkte direkt auf der Erdoberfläche liegen, es bieten sich also Straßenmarkierungen Gullydeckel und ähnliches an (Abb. 11.1). Dachfirste, Strommasten, Kirchtürme u.s.w. können nicht benutzt werden. Auch bei scheinbar parkenden Autos ist Vorsicht geboten, da man nur in den seltensten Fällen einen identischen Punkt auf dem Bo-

den, sprich Reifenunterkante in zwei Bildern genau identifizieren kann. Bedingt durch die kurze Zeitspanne zwischen zwei Aufnahmen können sich scheinbar stillstehende Objekte unbemerkt bewegt haben. Nachdem identifizieren von mindestens 4 Punkten kann mit einem Mausklick auf *ANALYZE* der momentane Fehler der Transformationsparameter mit den aktuell identifizierten Punkte berechnet werden. Bei einer flächenhaften Verteilung der Passpunkte in der Überlappungsfläche der beiden Bilder und gewissenhafter Bestimmung der Passpunkte ist es möglich, den Fehler im Bereich unter 10 Einheiten, in diesem Falle Pixel zu halten. Bei einer ungefähren Kantenlänge eines Pixels von 15 cm hätte man so eine Ungenauigkeit von 1,5 m. Dies erscheint bei einer Bildausdehnung von 1,8 \* 1,8 km und der Tatsache, daß die Luftbilder nicht entzerrt sind, annehmbar.



Abbildung 11.1.: Nutzung von *i.points* zur Identifizierung von Passpunkten in Luftbildern

## 11.4. Transformieren des Luftbildes

Mit dem Befehl *i.rectify* wird die gewünschte Transformation mit den bestimmten Parametern ausgeführt. Abschließend wird noch der neue Name in dem Ziel-*mapset* angegeben *Please select the file(s) you wish to rectify by naming an output file.*

```
GRASS:> i.rectify

Enter the group containing files to be rectified
Enter 'list' for a list of existing imagery groups
Enter 'list -f' for a verbose listing
Hit RETURN to cancel request
> 176
<176>

Please enter the order of the transformation you want: 1

Please select the file(s) you wish to rectify
by naming an output file

176@176 . . . . . 176_____

(enter list by any name to get a list of existing raster files)

      AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
      (OR <Ctrl-C> TO CANCEL)

Please select one of the following options
1. Use the current region in the target location
2. Determine the smallest region which covers the image
>2
```

Die angezeigte kleinste Ausdehnung der Startdatei in der Ziel-Region kann nun im Regionsbildschirm mit *ESC ENTER* bestätigt werden. Die Frage *Do you accept this region? (y/n) [n]* y wird bejaht und *Would you like this region saved as the region in the target location? (y/n)* verneint.

```
Do you accept this region? (y/n) [n] y
Would you like this region saved as the region
in the target location?
(y/n) n

You will receive mail when i.rectify is complete
GRASS:/spare/gisdata/grass/bin >
```

Wie schon geschrieben kann die aufgerufene Transformation einige Zeit dauern. Mit *exit* kann GRASS nun beendet werden. Nach der Meldung per e-mail, daß die Datei transformiert wurde, wechselt man in den Ziel - *Mapset 181* und führt hier *r.support* aus, um die von GRASS benötigten Hilfsdateien auf den neuesten Stand zu bringen oder überhaupt erst erzeugt. In einem gestarteten GRASS-Monitor können nun nacheinander beide vorhandene Rasterdateien (176 und 181) aufgerufen und durch die Option *-o* für Overlay überlagert werden (Abb. 11.2). Zur besseren Unterscheidung der einzelnen Luftbilder sind hier die Bildränder mit den Rahmenmarken nicht entfernt worden, bei der Arbeit mit Orthophotos würde natürlich vor der Transformation eine entsprechende Maske wie in 10.1.1 auf Seite

84 beschrieben erstellt und angewendet werden.

```
GRASS:/spare/gisdata/grass/bin > d.rast -o map=181  
GRASS:/spare/gisdata/grass/bin > d.rast -o map=176
```

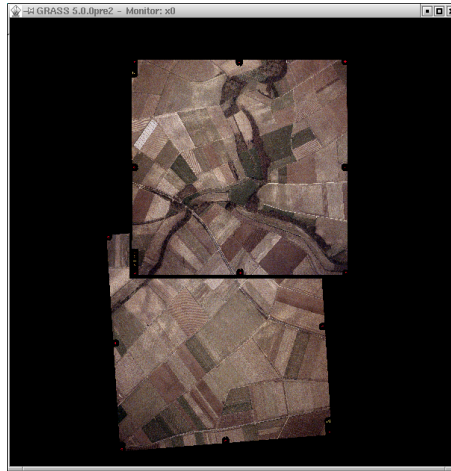


Abbildung 11.2.: Überlagerte Ausgabe von 2 Luftbildern

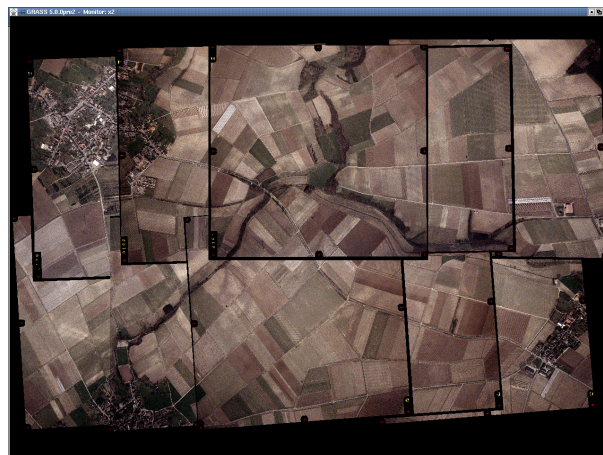


Abbildung 11.3.: Überlagerte Ausgabe von 10 Luftbildern

Um die beiden Luftbilder zu einer Rasterdatei zusammenzufügen, gibt es den interaktiven Befehl *r.patch*, der übergebene Rasterdateien so überlagert das die zuerst angegebene Datei als oberste „Folie“ erscheint und die später aufgerufenen Dateien in den Überlappungsgebieten verdeckt. Dem Programm werden an der Kommandozeile alle Rasterdateien übergeben, ist dies geschehen wird die folgende Aufforderung ohne Eingabe mit *ENTER* bestätigt, daraufhin fragt der Befehl den Namen der neuen Rasterdatei ab. Anschließend wird wieder *r.support* für die neuerstellte Datei aufgerufen.

## **11.5. Abschließendes Vorgehen**

Diese Art der Transformation wird für alle verbliebenen „Luftbildpaare“ ausgeführt. Anschließend können die 4 äußeren neugebildeten „Patch-Dateien“ wiederum in das System der „mittleren“ transformiert und „gepatcht“ werden. Trotz der in Kauf genommenen Ungenauigkeiten kann sich das Ergebnis sehen lassen und ist für Übersichtszwecke auf jeden Fall geeignet (Abb. 11.3).

## **11.6. Alternative Vorgehensweisen**

Sollten auf diese Weise die Ungenauigkeiten zwischen den einzelnen Bildblöcken zu groß werden, kann man versuchen, jedes Luftbild einzeln an das Ursprungsbild anzuhängen. So könnten die Spannungen zwischen den einzelnen Bildern unter Umständen besser verteilt werden.

# 12. Verarbeiten von GPS - Daten

## Aufgabenstellung

GPS steht für Global Positioning System. GPS ist ein System zur 3-dimensionalen Positionsbestimmung mittels räumlicher Bogenschläge zu mindestens 4 Satelliten mit bekannten Positionen. Dahinter stehen die Satelliten-Systeme navstar-gps (USA) und glonass(GUS) sowie in Zukunft auch gallileo(EU). Es gibt mehrere Anbieter für sogenannte Handheld bzw. Handy - GPS-Empfänger am Markt, darunter Casio, Garmin, Magellan, Rand Mc Nelly. Die folgenden Programme wurden in Verbindung mit einem Garmin GPS II betrieben. Dieser Empfänger wurde mir freundlicherweise von der Fachhochschule Oldenburg/Ostfriesland/Wilhelmshaven zur Verfügung gestellt.

## Beispieldaten

Die in diesem Kapitel benutzten Daten werden durch die GPS Empfänger während des Gebrauches erzeugt.

## 12.1. Programm gpspoint

Das Programm gpspoint wurde in der Version 1.010708 unter einer Debian GNU/Linux Installation (woody) verwendet.

Das Programm gpspoint wird direkt von der Kommandozeile aus betrieben. Es ermöglicht das Ein- und Auslesen von Waypoints (vorgegebene oder im Feld erzeugte koordinierte Punkte), Routes (aus Waypoints zusammengesetzte Routen, deren einzelne Waypoints nacheinander vor- oder rückwärts abgelaufen werden) oder Tracks (automatische Wegprotokollierung durch Routen). Weiter können im NEMA - Modus die aktuell empfangenen GPS-Daten im Roh-Zustand laufend ausgegeben werden und so von anderen Anwendungen oder Geräten verwendet werden.

**Vorbereitungen** Der Gps-Empfänger wird per seriellem Kabel (RS232) mit dem Rechner verbunden. Üblicherweise gibt es die sogenannten COM-Schnittstellen oder COM-Ports 1 und 2. In einem Linux-System können diese Schnittstellen direkt über `/dev/ttyS0` für COM 1 und `/dev/ttyS1` für COM 2 angesprochen werden. Am Garmin wird als Interface die Option GRMIN/GRMIN (GPS II) gewählt.

## Tipps

- Um die Kommunikation zwischen Garmin und Rechner auf grundlegendste Art und Weise zu überprüfen kann man wie folgt vorgehen. Das Garmin Interface ist auf NMEA/NMEA zu stellen. Hierdurch werden die Empfangenen Daten im NMEA Protokoll an die Schnittstelle weitergereicht, auch wenn der Garmin keine Satelliten empfängt, werden weiterhin „leere“ Protokollpakete an die Schnittstelle gesendet. Um sich diese Pakete am Bildschirm anzeigen zu lassen, gibt man an der Kommandozeile den Befehl `cat /dev/ttyS1` für einen an dem COM2-Port angeschlossenen Garmin ein. Die auf dem Bildschirm ausgegebenen Datensätze sollten ähnlich den folgenden sein. Hierdurch lässt sich sozusagen die elektrische Verbindung zwischen Garmin und Computer testen.

```
bash-2.05$ cat /dev/ttyS1
0,334,00*7B
$PGRME,,M,,M,,M*00
$GPGLL,,,,,082435,*58
$PGRMZ,,, *7E
$PGRMM,WGS 84*06
$GPBOD,,T,,M,, *47
$GPRTE,1,1,c,0*07
$GPRMC,082436,V,,,,,,051001,, *3F
$GPRMB,V,,,,,,V*66
$GPGGA,082436,,,,,0,00,,,M,,M,, *6D
$GPGSA,A,1,,,,,, *1E
$GPGSV,3,1,12,03,00,087,00,04,58,252,00,05,00,307,00,06,00,356,00*7F
$GPGSV,3,2,12,07,00,224,00,08,03,190,00,09,00,283,00,22,13,065,00*7A
$GPGSV,3,3,12,23,00,288,00,24,38,304,46,27,28,176,00,30,00,334,00*79
$PGRME,,M,,M,,M*00
$GPGLL,,,,,082437,*5A
$PGRMZ,,, *7E
$PGRMM,WGS 84*06
$GPBOD,,T,,M,, *47
$GPRTE,1,1,c,0*07
$GPRMC,082438,V,,,,,,051001,, *31
$GPRMB,V,,,,,,V*66
$GPGGA,082438,,,,,0,00,,,M,,M,, *63
$GPGSA,A,1,,,,,, *1E
$GPGSV,3,1,10,03,00,087,00,04,58,252,00,06,00,356,00,07,00,224,00*7F
$GPGSV,3,2,10,08,03,190,00,09,00,283,00,10,01,292,00,23,00,288,00*70
$GPGSV,3,3,10,24,38,304,46,30,00,334,00,,,,, *77
$PGRME,,M,,M,,M*00
$GPGLL,,,,,082439,*54
$PGRMZ,,, *7E
```

- Sollte der Aufruf `cat /dev/ttyS1/` mit einem `bash: /dev/ttyS1: Keine Berechtigung` beschieden werden, so sind die Zugriffsrechte für den Benutzer nicht ausreichend um auf die serielle Schnittstelle zuzugreifen.



**Programmaufruf** Der einfache Aufruf `./gpspoint` an der Kommandozeile gibt den Hilfe-Bildschirm aus, der auch über `./gpspoint -h` erzeugt werden kann.

```
bash$ ./gpspoint

*****
usage :

-g          Garmin mode
-n          NMEA mode
-d device   Device eg: -d /dev/ttyS0
-s bauds    line Speed; it is save not to use this option,
            there are defaults

-dw filename Download Waypoints from gps and
            write them to filename
-dr filename Download Routes
-dt filename Download Tracks
-da filename Download All, Waypoints/Routes/Tracks
-uw filename Upload Waypoint to gps
-ur filename Upload Routes
-ut filename Upload Tracks
-ua filename Upload All, Waypoints/Routes/Tracks

-p          display current position
-o          Turn Off Device
-h          Help
-a          about

please have a look at the man page, available in german and
            english !
*****
```

### 12.1.1. Verwalten von GPS-Daten

Das Auslesen von Waypoints geschieht durch den Kommandozeilenaufruf `./gpspoint -g -d /dev/ttyS1 -dw /tmp/gpspoint/heiko`. Dabei bezeichnen die Flaggen (Optionen) `-g`, wie aus dem Startbildschirm ersichtlich, den Programmablauf im sogenannten Garmin - Modus, `-d` gibt die COM Schnittstelle an, `-dw` liest die waypoints aus dem Speicher des Garmin aus und sichert sie in der Datei `/tmp/gpspoint/heiko`.

```

bash$ ./gpspoint -g -d /dev/ttyS1 -dw /tmp/gpspoint/heiko
***** Garmin Device Info *****
      Garmin Product ID: 73
      Garmin software version: 207
      Garmin Product description: GPS II+ SOFTWARE 2.07
      Link Protocol: 1
      Command Protocol: 10
      Waypoint Protocol: 100
      Waypoint Data: 103
      Route Protocol: 200
      Route Header: 201
      Route Data: 103
      Track Protocol: 300
      Track Header: 0
      Track Data: 300
      Prx Protocol: 0
      Prx Data: 0
      Alm Protocol: 500
      Alm Data : 501
*****
      date: 2001-10-04
      time: 14:56:39
*****
serial port: /dev/ttyS1
serial port speed: default
mode : garmin
Getting Waypoints, 10 packets 100% done
Writing Waypoints to /tmp/gpspoint/heiko, done
*****
good bye

```

Die so ausgelesenen 10 Punkte werden in einer Datei mit folgender Form gespeichert. Der „Lattenzaun“ zu Beginn der Headerzeilen kommentiert diese aus, so das sie nicht beachtet werden. Zum „Hochladen“ von Waypoints in den Garmin muss die Koordinatenliste in der gleichen Form vorliegen.

```

#NOTE: the gpd-filestructure has slightly changed
#if you have gpspoint version < 1.010622 upgrade
#
#GPSPOINT DATA FILE
#gpspoint version: 1.010708
#http://scampi.physik.uni-konstanz.de/~tschank/gpspoint
#GPS Device : GPS II+ SOFTWARE 2.07
#download time (local) : Thursday 04 October 2001 15:12
#waypoints

type="waypointlist"
type="waypoint" latitude="52.2749" longitude="8.04706" name="001  "
comment="02-OCT-01 10:54" symbol="dot" display_option="symbol+name"

type="waypoint" latitude="52.27457" longitude="8.04646" name="002  "
comment="02-OCT-01 10:55" symbol="dot" display_option="symbol+name"

type="waypoint" latitude="52.2749" longitude="8.04555" name="003  "
comment="02-OCT-01 10:56" symbol="dot" display_option="symbol+name"

type="waypoint" latitude="52.27522" longitude="8.04531" name="004  "
comment="02-OCT-01 10:56" symbol="dot" display_option="symbol+name"

type="waypoint" latitude="52.2759" longitude="8.04475" name="005  "
comment="02-OCT-01 10:57" symbol="dot" display_option="symbol+name"

type="waypoint" latitude="52.27597" longitude="8.0441" name="006  "
comment="02-OCT-01 10:58" symbol="dot" display_option="symbol+name"

type="waypoint" latitude="52.27621" longitude="8.03812" name="007  "
comment="02-OCT-01 11:12" symbol="dot" display_option="symbol+name"

type="waypoint" latitude="52.27575" longitude="8.03741" name="008  "
comment="02-OCT-01 11:13" symbol="dot" display_option="symbol+name"

type="waypoint" latitude="52.27517" longitude="8.0361" name="009  "
comment="02-OCT-01 11:14" symbol="dot" display_option="symbol+name"

type="waypoint" latitude="52.27501" longitude="8.03234" name="010  "
comment="02-OCT-01 11:17" symbol="dot" display_option="symbol+name"

```

Auf die gleiche Weise verfährt man beim Verwalten der Tracks und Routes.

### 12.1.2. Ausgabe der aktuellen Position

Für diese Option muß der Gramin in das Interface NMEA/NMEA geschaltet werden, da sie momentan nur hier für dieses Protokoll implementiert ist. Der Aufruf an der Kommandozeile ist `./gpspoint -n -d /dev/ttyS1 -p`. Die Ausgabe erfolgt in geographischen Koordinaten.

```
bash-2.05$ ./gpspoint -n -d /dev/ttyS1 -p

***** NMEA Interface *****
serial port: /dev/ttyS1
serial port speed: default
mode : nmea
latitude 52.2747 longitude 8.0475
*****
good bye
```

## 12.2. Programm GPSMan

Dieses Programm zur Verwaltung und Bearbeitung von GPS-Daten verfügt über eine GUI und unterstützt laut Homepage <sup>1</sup> Empfänger von Garmin und Lowrance. Das Tutorial stützt sich auf Erfahrungen, die mit dem Garmin-Empfänger GPS II+ gemacht wurden.

Es wurde GPSMan Version 5.4 unter einer Debian GNU/Linux Installation (woody) verwendet.

Ist eine Verbindung mit dem Programm `gpspoint` zustande gekommen, so sollte GPSMan (GPSManager) relativ problemlos zum Laufen zu bringen sein. Nachdem man GPSMan an einen Platz seiner Wahl entpackt hat, muß das Startskript `gpsman.tcl` angepasst werden. Dies erfolgt mit einem Editor nach Wahl. Die folgenden Zeilen sind auf die eigene Installation abzustimmen:

```
set LANG deutsch

# otherwise the default is
set DEFSPORT /dev/ttyS1

# path to directory containing user data
set USERDIR /spare/gisdata/gps/gpsman/gpsman-5.4/.gpsman-dir

# path to directory containing program files
set SRCDIR /spare/gisdata/gps/gpsman/gpsman-5.4/gmsrc
```

Eine deutschsprachige Nutzerumgebung wählt man mit `set LANG deutsch`. Die Zeile `set DEFSPORT /dev/ttyS1` bezeichnet die vorgewählte serielle Schnittstelle, in diesem Fall COM2. Diese Einstellung kann im Programm geändert werden. Durch `set USERDIR /spare/gisdata/gps/gpsman/gpsman-5.4/.gpsman-dir` wird der Pfad deklariert, in dem GPSMan ein Verzeichniss mit Nutzerdaten anlegen kann. In `set SRCDIR /spare/gisdata/gps/gpsman/gpsman-5.4/gmsrc` schließlich wird der Pfad angegeben, in dem sich das Verzeichniss `gmsrc` befindet. Dies ist üblicherweise das Verzeichniss, in das GPSMan entpackt wurde. Sind diese

---

<sup>1</sup><http://www.ncc.up.pt/mig/gpsman.html>

Einstellungen gemacht, so kann das Startskript *gpsman.tcl* im GPSMan Installationsverzeichnis durch Eingabe von *./gpsman.tcl* an der Kommandozeile aufgerufen werden. Jetzt sollten 3 Fenster erzeugt werden, mit denen GPSMan im folgenden bedient wird.

**Prüfen der Verbindung** Im Fenster *GPS Empfänger* (Abb. 12.1) kann die Verbindung zwischen Rechner und Garmin überprüft werden, indem man zuerst das Protokoll *Garmin* im Protokollmenü auswählt. Ein weiterer Mausklick auf die Schaltfläche *offline* überprüft nun die Verbindung und sollte als Ergebnis ein kleines Fenster mit der Meldung *Verbindung ok* erzeugen. Die gerade benutzte Schaltfläche ist nun mit einem grünen Schriftzug *online* belegt.

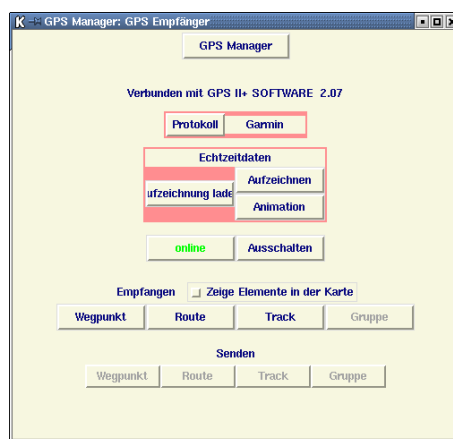


Abbildung 12.1.: GPSMan GPS Empfänger mit überprüfter Verbindung zum Garmin GPS II+

### 12.2.1. Verwalten von Daten

Sind im Garmin Wegpunkte, Routen, Tracks oder Gruppen gespeichert, so können sie im Bildschirm *GPS Empfänger* unter der Überschrift *Empfangen* aus dem Garmin abgerufen werden. Hat man den Checkbutton *Zeige Elemente in der Karte* gesetzt, so werden die importierten Elemente in dem Hauptfenster angezeigt (Abb. 12.3). Eine Listenansicht wird in *GPSManager: Listen* ausgegeben (Abb. 12.2).

### 12.2.2. Anzeigen und Speichern von Echtzeitdaten

Mit GPSMan hat man die Möglichkeit in Echtzeit die ermittelten Koordinaten in einem proprietären Garmin-Format (Abb. 12.4) oder im NMEA-GPS-Datenformat (Abb. 12.5) direkt in den Computer zu übernehmen. Nachdem am Garmin und im GPS Manager ein übereinstimmendes Protokoll ausgewählt wurde, wird unter der Überschrift *Echtzeitdaten* die Aufzeichnung begonnen. Mit *Aufzeichnung laden* wird die Visualisierung gestartet. In dem Fenster *Echtzeitdaten* kann per Schieberegler das Aufzeichnungsintervall verändert

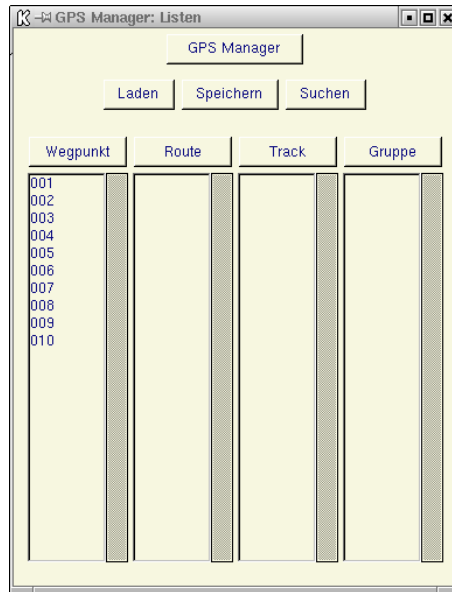


Abbildung 12.2.: GPSMan GPS Listen mit importierten Wegpunkten

werden. Ist der Empfänger in Bewegung, können mit einem Mausklick auf Animation die bisher aufgezeichneten Datensätze der Reihe nach im Hauptfenster angezeigt werden.

### 12.2.3. Erstellen von Routen

Im Fenster *GPS Manager: Listen* können Wegpunkte, Routen, Tracks und Gruppen geladen und gesichert werden. Weiter können aus Wegpunkten Routen zusammengestellt werden oder aus Tracks Routen ausgeschnitten werden. Wird im Hauptfenster eine georeferenzierte Karte angezeigt, so können in ihr Wegpunkte markiert und diese im *GPS Manager: Listen* zu einer Route zusammengestellt werden die dann in den Garmin importiert werden kann.

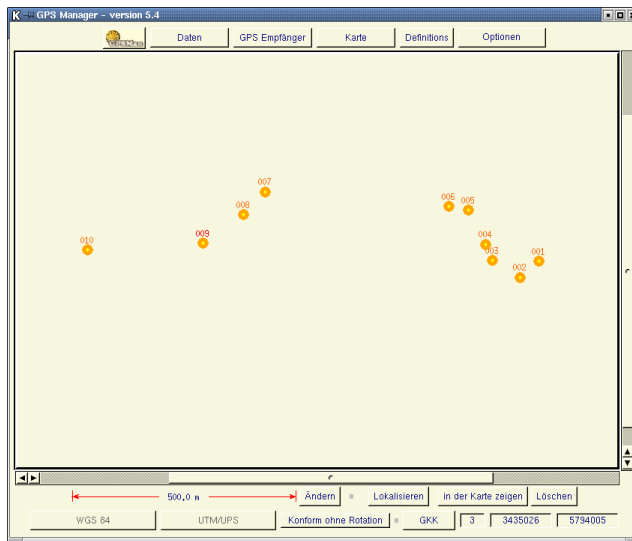


Abbildung 12.3.: GPSMan GPS Hauptfenster mit importierten Wegpunkten

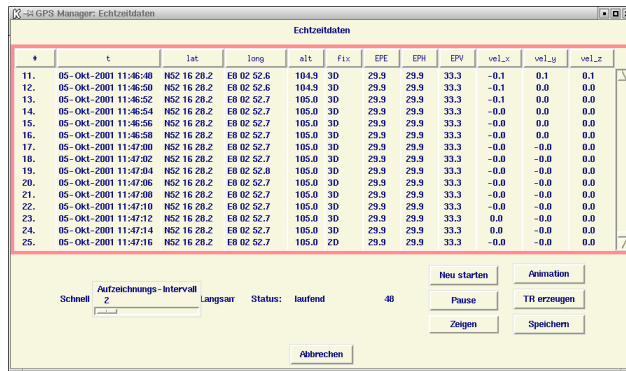


Abbildung 12.4.: GPSMan Echtzeitdatenerfassung im Garmin Format

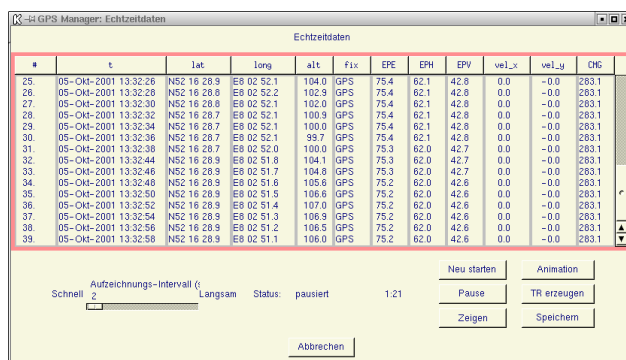


Abbildung 12.5.: GPSMan Echtzeitdatenerfassung im NMEA Format





# 13. Differentielle GPS - Messung

## Aufgabenstellung

Um die Genauigkeit bei der Positionsbestimmung mittels GPS in den Submeterbereich zu bringen, wurde die differentielle GPS-Messung entwickelt. Hierbei wird auf einer sogenannten Referenzstation, deren Position genau bekannt ist, von einem GPS-Empfänger kontinuierlich seine Position bestimmt und mit der bekannten verglichen. In der Anfangszeit wurde diese so ermittelte Differenz einfach an die zu referenzierenden Empfänger also ,beispielsweise dem im Tutorial benutzten Garmin, übermittelt und damit die seinerseits bestimmte Position verbessert. Diese Methode funktioniert aber nur zufriedenstellend, wenn von Referenzstation und Garmin dieselben Satelliten benutzt werden. Aufgrund der oftmals unterschiedlichen Satellitenkonstellationen bei Referenzstation und Empfänger ist diese Methodik nur eingeschränkt anwendbar. Viel flexibler wird man durch die Übermittlung von Korrekturdaten für jeden einzelnen Satelliten, den die Referenzstation empfängt. Diese Korrekturdaten werden vom Empfänger während der Messung auf jeden einzelnen Satelliten angewendet.

## Beispieldaten

Die benötigten Korrekturdaten werden in Echtzeit über das Internet übertragen und dem Programm zur Verfügung gestellt.

## 13.1. Lösung mit DGPSIP

Es wurde DGPSIP 1.32 unter einer Debian GNU/Linux Installation (woody) verwendet.

Ein Problem bei der Übermittlung der Korrekturdaten ist, die Verbindung zur Referenzstation auch über weitere Entfernungen aufbauen zu können. Ist ein Rechner mit Internetzugang und serieller Schnittstelle vorhanden, können solche Korrekturdaten im RTCM-Format direkt an den Garmin übermittelt werden. Laut Homepage des Initiators dieses Services Wolfgang Rupprecht<sup>1</sup>, gibt es zur Zeit 3 Referenzstationen in den USA, die solche Korrekturdaten im Internet zur Verfügung stellen. Trotz dieser Entfernung ist es bereits gelungen in den USA und Deutschland gleichzeitig dieselben Satelliten zu empfangen und so eine differentielle GPS-Messung durchzuführen, die zu einer Verbesserung der im Garmin-display angezeigten Punktgenauigkeit führte. Während der Erstellung des Tutorials ist die

<sup>1</sup><http://www.wscc.com/wolfgang/gps/dgps-ip.html>

Verbindung mit einer Referenzstation in Kalifornien und New York gelungen, aber durch die denkbar schlechte Satellitenkonstellation (Empfang von 4 Satelliten durch ein Bürofenster) hat sich die angezeigte Genauigkeit eher verschlechtert. Dessen ungeachtet wurde vom Garmin eine differentielle Messung angezeigt, was nach Rücksprache mit dem Entwickler bestätigt das wenigstens 1 identischer Satellit von Garmin und Referenzstation empfangen wurde.

### 13.1.1. Durchführung der Messung

**Einstellungen am Garmin** Am Garmin wird unter dem Menu Interface die Verbindungsart „RTCM/NONE“ mit der Baudrate 4800 eingestellt. Die Meldung „BEACON RCVR: RECEIVING“ im Garmin Display zeigt an das tatsächlich Daten empfangen werden. Die „FREQ“ Frequenz wird zusammen mit der aktuellen Dämpfung des Signales an der Empfangsantenne automatisch übermittelt. Eine Berechnung der Entfernung Empfänger-Referenzstation wird bei „DIST“ ausgegeben. Die Übertragungsrate der Korrekturdaten ist im Moment bei 200 bps (bits per second), diese Einstellung kann beim Garmin II + in dem Verbindungsprotokoll „RTCM/NMEA“ verändert werden.

**Einstellungen am Rechner** Grundlegende Voraussetzung ist eine funktionierende Verbindung zum Internet. Ist die Datenübertragung Rechner-Computer etabliert braucht nur das Programm *DGPSIP* gestartet werden. Im Betrieb unter GNU/LINUX arbeitet man an der Kommandozeile für Windows gbt es ein GUI. Die bei jedem Aufruf angezeigte Warnung das das kann durch setzt des Schalters *-w* im Programmaufruf verhindert werden.

```
bash-2.05$ ./dgpsip --help
dgpsip receives DGPS over IP and sends to serial port
Usage: dgpsip [OPTIONS]
Options are:
  -c capture_file  saves GPS NMEA data to file
  -d               decode RTCM and print to stdout
                  this turns off tty output, use -o to turn it on again
  -h DGPS_host    sets name of server to connect to
                  default is dgps.wsrcc.com
  -p DGPS_port    sets IP port #/name to connect to
                  default is IANA assigned port rtcm-sc104 (2101)
  -o serial_port  sets name of serial output device
                  default is /dev/gps - a link to /dev/tty??
  -s serial_speed sets serial output baud rate
                  default is 4800 baud
  -v verblevel    turns on extra output
                  see README for details
  -w              turns off startup warranty message
```

Mit der Option *-o* wird der serielle Anschluss angegeben an dem der Garmin-Empfänger angeschlossen ist. Die Internet-Adresse kann mit *h* angepasst werden, die Portnummer mit *-p*. Als Grundeinstellung sind hier jeweils die passenden Angaben für die Referenzstation in Point Blunt, Kalifornien angegeben. Um den Empfang der Korrekturdaten aus dem In-

ternet zu überprüfen kann beim Aufruf von *dgpsip* der Schalter *-d* gesetzt werden der die empfangenen Korrekturdaten dekodiert und an die Kommandozeile übergibt. Nach ein paar Zeilen ohne Daten die zur Synchronisation des Empfangs mit der Ausgabe dienen, werden die Korrekturdaten an der Kommandozeile ausgegeben und könnten von hieraus durchaus weiterverwendet werden.

```

bash-2.05$ ./dgpsip -o /dev/ttyS1 -w -d

H      9      268      2983.2  1      4      2
S     26      0       30      2983.2 -1.760  0.002
S     21      0      151      2983.2 -0.880  0.004
H      9      268      2984.4  2      5      3
S     17      0       76      2984.4  2.940  0.000
S      6      0       84      2984.4 -0.100  0.002
S     15      0      190      2984.4  2.740  0.000
H      9      268      2985.0  3      5      2
S     23      0       77      2985.0  2.740  0.000
S     18      0       22      2985.0  2.660  0.002
S     22      0      223      2985.0  0.600 -0.002
f    3100     200
s    321      36
H      9      268      2986.2  4      4      2
S     26      0       30      2986.2 -1.780 -0.004
S     21      0      151      2986.2 -0.900  0.004
H      9      268      2987.4  5      5      3
S     17      0       76      2987.4  2.940  0.000
S      6      0       84      2987.4 -0.120  0.002
S     15      0      190      2987.4  2.780  0.000
H      9      268      2988.0  6      5      2
S     23      0       77      2988.0  2.740  0.000
S     18      0       22      2988.0  2.700  0.004
S     22      0      223      2988.0  0.600  0.002
H      9      268      2989.2  7      4      2
S     26      0       30      2989.2 -1.700  0.000
S     21      0      151      2989.2 -0.900  0.002
H      9      268      2990.4  0      5      3
S     17      0       76      2990.4  2.980  0.002
S      6      0       84      2990.4 -0.060  0.002
S     15      0      190      2990.4  2.840  0.002
f    3100     200
s    321      36

```

Bekommt man ein Ausgabe ähnlich der ausgegebenen, kann man diese Daten nun an den Garmin umleiten, oder besser gesagt die Umleitung an die Kommandozeile, die sogenannte Standardausgabe (stdout) aufheben. Dieser Aufruf an der Kommandozeile ist denkbar einfach. Nach ein paar Sekunden sollten nun im Garmin-Display Angaben über Frequenz Dämpfung und falls eine Positionsbestimmung möglich war auch die Entfernung zur Referenzstation abzulesen sein.

```

bash-2.05$ ./dgpsip -o /dev/ttyS1 -w

```



# 14. Flug über das Höhenmodell

## Aufgabenstellung

Im folgenden werden zwei verschiedene Arten von Flügen über ein Höhen bzw. Landschaftsmodell besprochen. Einmal die direkte interaktive Bewegung über das Modell in „Echtzeit“ und das Berechnen einer im voraus festgelegten Route über ein Modell, um z.B. eine Animation daraus zu erzeugen.

## Beispieldaten

Für die Bearbeitung mit GRASS werden die in Kapitel 9 Seite 71 importierten Daten verwendet.

VTP Enviro kann die DEM Daten des USGS Crater Lake Data Clearinghouse<sup>1</sup> direkt importieren.

## 14.1. Lösung mit VTP Enviro

Es wurde VTP Enviro auf einer Windows 2000 Installation verwendet.

Das Programm VTP Enviro gehört wie VTBuilder und BExtractor zu dem Virtual Terrain Projekt (VTP). VTP Enviro bietet die Möglichkeit, die mit VTBuilder und BExtractor bearbeiteten Projekte anzuzeigen und sich virtuell in ihnen zu bewegen. Wie schon in Abschnitt 6 Seite 56 ff beschrieben wird ein DEM in VTBuilder mit *Layer/Import Data* und der Auswahl *Elevation* eingelesen. Um diese Daten nun in VTP Enviro zu übernehmen, werden sie als .bt (Binary Terrain)<sup>2</sup> Datei durch *Elevation/Merge and Export Elevation* in dem Ordner „Elevation“ im Installationspfad von VTP Enviro gespeichert. Um den „Bildausschnitt“ und die Auflösung zu justieren, kann mit *Export Area/Set to Extents* das gesamte DEM ausgewählt werden, mit *Numeric Values* gibt man die Ausdehnung durch entsprechende Koordinatenwerte ein. Die Schaltfläche *Set Export Area* ermöglicht das Aufziehen eines Ausgaberechteckes direkt in der Bildschirmanzeige des DEMs. In dem nächsten Auswahlmenü können die Auflösung und Ausdehnung der BT-Datei gewählt werden. Um sie aufeinander anzupassen sollten die Schaltflächen  $\ll$  und  $\gg$  genutzt werden.

---

<sup>1</sup><http://craterlake.wr.usgs.gov/dem.html>

<sup>2</sup>[www.vterrain.org/Implementation/BT.html](http://www.vterrain.org/Implementation/BT.html)

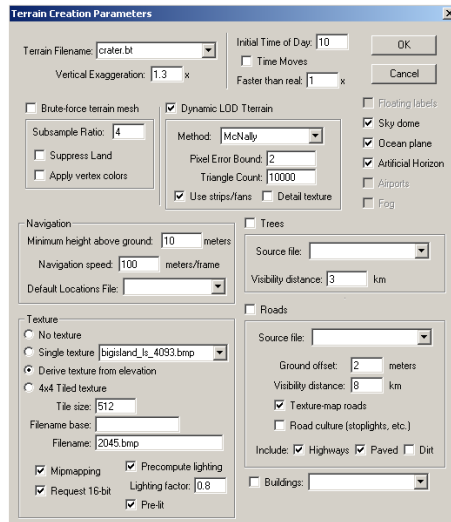


Abbildung 14.1.: Auswahl der Anzeigeparameter in VTP Enviro

**Aufruf von VTP Enviro** Nach dem Aufruf von VTP Enviro wird das *Enviro Startup* Menü gestartet. Mit *Select/Generic Terrain* erhält man die Möglichkeit, nicht vordefinierte DEMs anzuzeigen. Die Auswahl des DEM geschieht durch *Edit Properties/Terrain Filename*. weiter sollten als Anzeigeparameter gesetzt sein: *Dynamic LOD Terrain* und *Texture/Derive texture from elevation*. Nach der Bestätigung *OK* dieser Einstellungen startet VTP Enviro. Grundsätzlich gibt es zwei Anzeigemodi: *Earth* (Abb. 14.3) und *Terrain* (Abb. 14.2). *Earth* öffnet eine Globusansicht der Erde mit Markierungen für vorhandene anwählbare *Terrain*-Ansichten wie z.B. das importierte DEM. Mit gedrückter mittlerer Maustaste lässt sich der Globus rotieren und ein Linksklick auf eine Markierung öffnet die entsprechende *Terrain*-Ansicht.

**Steuerung des Fluges** Die Navigation erfolgt mittels der Maus. Bei gedrückter linker Maustaste können Bewegungen nach vorne und rückwärts sowie seitliche Drehungen ausgeführt werden. Die linke Maustaste kontrolliert die Flughöhe und Bewegungen nach Rechts und Links. Werden beide Maustasten gedrückt kann eine Loopingartige Bewegung ausgeführt werden. Das farbige Achsenkreuz dient zur Orientierung und als Cursor um Bildinformationen abzufragen. Die „Zickzacklinie“ im linken unteren Bildrand ist eine Visualisierung der momentan ausgegebenen frames per second (fps), also die Bildwiederholfrequenz von VTP Enviro. Die Möglichkeit einer Aufzeichnung des Fluges besteht momentan noch nicht (Abb. 14.3).

## 14.2. Lösung mit GRASS

Es wurde GRASS 5.0pre1 unter einer Red Hat GNU/Linux Installation verwendet.

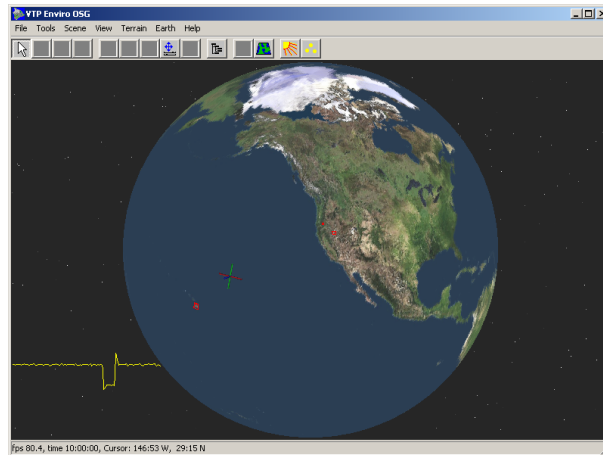


Abbildung 14.2.: VTP Enviro im Earth-Modus

Der Befehl *nviz2.2* mit den Parametern *el=* für die Rasterdatei mit Höheninformationen, *ve=* für Vektordateien, z.B. Höhenlinien und *si=* für Punktdaten öffnet eine GUI „Graphische Benutzer Oberfläche“ mit der das Programm bedient wird.

Diese GUI ermöglicht es dem Benutzer, den Beobachtungspunkt „Projektionszentrum“, den Überhöhungsfaktor *zexag* und das Blickfeld *perspective* dynamisch zu verändern. Mit dem Menüpunkt *Panel/Surface* wird das „Oberflächen Schalterpult“ aufgerufen, in welchem mit der Schaltfläche *color* georeferenzierte Rasterdateien als Texturen auf die 3D-Oberfläche gelegt werden können. Nach dem Auswählen und Akzeptieren einer Rasterdatei wird das 3D-Modell durch Anklicken der *Surface* Schaltfläche aktualisiert.

Vektordaten, zum Beispiel Höhenlinien können mittels *Panel/Vectors* in *nviz2.2* übernommen werden. Mit der Schaltfläche *New* wird eine Vektordatei ausgewählt und mit *Accept* eingelesen. Ein Klicken auf die Schaltfläche *Vectors* zeichnet die Monitorausgabe neu.

Einen Überflug definiert man mit dem Menüpunkt *Panel/Animation*. Hier sollten unter *total frames* für eine 10 Sekunden lange Animation ca. 300 frames „Bilder, Rahmen“ eingestellt werden. Man wählt nun Projektionszentrum, Blickfeld u.s.w. für den Anfangspunkt des Fluges aus. Mit dem Schieberegler *Key Frames* legt man den Zeitpunkt für weitere Stationen des Fluges fest. Die Anzahl der Frames zwischen den Key Frames ergibt die Dauer der einzelnen „Einstellungen“. Die „Kamerafahrten“ von einem Key Frame zum nächsten können *linear*, d.h. auf einer Geraden liegen, oder als *spline*, also als Aneinanderreihung von Kurvensegmenten, die die Projektionszentren bestmöglichst verbinden, definiert werden. Mit dem *tension-Schieber* kann der Grad der „Anschmiegung“ an die definierten Projektionszentren des Fluges beeinflusst werden.

Die Schaltfläche *run and save Images* öffnet ein Fenster in dem Pfad und Dateiname der Flugsequenz *Enter a base name* angegeben werden, weiter wird die Art der Datei *IrisRGB*, *PPM* oder *TIFF* definiert und es besteht die Möglichkeit, sich den Flug als Drahtgittermodell *Wireframe* oder mit berechneter Oberfläche *Full Rendering* erstellen zu lassen. Die gespeicherten Bilder müssen jetzt noch mit einem geeigneten Programm z.B. *ppmtompeg* oder *sampeg* zu einer MPEG-Datei zusammengefasst werden.

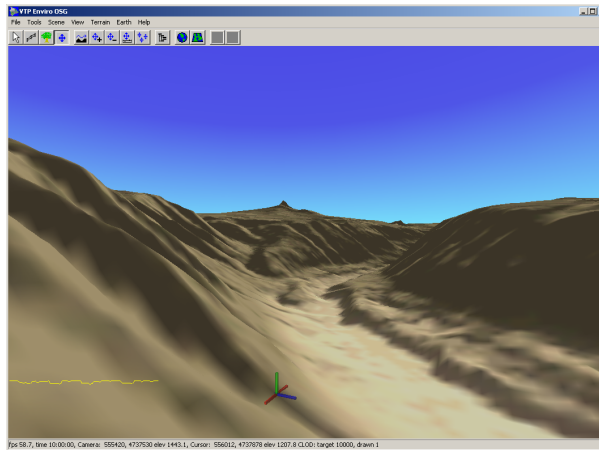


Abbildung 14.3.: VTP Enviro im Terrain-Modus beim Flug über das USGS-DEM Craterlake

```
GRASS:~ > nviz2.2 el=rasterdata ve=hlinien si=data
```

### Tipps:

- Während der Berechnung der einzelnen Frames sollte der Rechner nicht für andere Arbeiten benutzt werden, da ansonsten über den `nviz2.2` -Monitor gelegte Fenster in den Flug übernommen werden oder beim Arbeiten an anderen Konsolen nur noch ein schwarzes Bild produziert wird.
- Mit *File/Image Dump* können Einzelbilder gespeichert werden.

### 14.2.1. Zusammenfassen der Flugeinzelbilder zu einem Film

#### Lösung mit Programm `ppmtompeg`

**Programmaufruf** Dem Programmaufruf `ppmtompeg` wird eine Parameterdatei übergeben die folgende Informationen enthält:



```
bash-2.05$ cat parameterdatei
```

```
PATTERN IBBPBBPBBPBB
OUTPUT /home/heiko/flug/flympeg1
INPUT\_DIR /tmp/flug/
INPUT
fly*.ppm [00001-00349]
END\_INPUT
BASE\_FILE\_FORMAT PPM
INPUT\_CONVERT *
GOP\_SIZE 16
SLICES\_PER\_FRAME 1
PIXEL HALF
RANGE 10
PSEARCH\_ALG EXHAUSTIVE
BSEARCH\_ALG CROSS2
IQSCALE 8
PQSCALE 10
BQSCALE 25
REFERENCE\_FRAME DECODED
```

Wichtig sind hierbei die korrekten Pfadangaben für Eingabe- *INPUT\_DIR /tmp/flug/* und Ausgabedateien *OUTPUT /home/heiko/flug/flympeg1*. Die in eckigen Klammern stehenden Zahlen geben die an *fly* anzuhängenden Dateibezeichnungen an.

```
bash-2.05$ ppmtompeg parameterdatei
```

## Lösung mit Programm sampeg

**Programmaufruf** Ein beispielhafter Programmaufruf zur Erzeugung einer MPEG-1 Datei wäre:

```
bash-2.05$ sampeg-2 -1 -Y /tmp/flug/newfly\%05d.ppm
/home/test/flug/fly-2.1.mpg
```

Dabei bestimmt *-1* das Format MPEG-1, *-Y* setzt als Eingabeformat PPM-Dateien. */tmp/flug/newfly%05d.ppm* bezeichnet den Eingangspfad, wobei *%05d* ein Platzhalter für eine 5-stellige Dezimalzahl ist von der aus aufsteigend alle vorliegenden Frames abgearbeitet werden. Der Ausgabepfad ist in diesem Beispiel */flug/fly-2.1.mpg* also im Heimatverzeichnis der Ordner *flug* und die Dateibezeichnung *fly-2.1.mpg*.



# 15. Dynamische Karten im WWW (Web Mapping)

## Aufgabenstellung

Web Mapping bietet die Möglichkeit, Rasterkarten und/oder Vektorkarten interaktiv in einem Datennetz, meist dem Internet, zu präsentieren. Typische Grundfunktionen sind zoomen und verschieben der Karte (Navigation auf der Karte), und das Verbinden von Hyperlinks (URLs) mit Orten die in der Karte dargestellt sind.

## Beispieldaten

Bei der Anwendung MapIt! werden die im Programmpaket enthaltenen Beispieldaten verwendet.

Für Mapserver werden Daten des Craterlakes aus den Kapiteln 7 Seite 63 und 9 Seite 71 sowie weitere Datensätze (Shapefiles über Straßen und Höhenlinien) des USGS Crater Lake Data Clearinghouse<sup>1</sup> verwendet.

## 15.1. Lösung mit MapIt!

Es wurde MapIt! 1.0.1-1 unter einer Debian GNU/Linux Installation (woody) und unter Windows 2000 verwendet.

Das Programm MapIt! bietet die Möglichkeit via Web-Browser auf Rasterkarten zu navigieren, hinein- und herauszuzoomen sowie darzustellende Objekte/Objektklassen auszuwählen und auf der Karte zu identifizieren. Es können vorhandenen Webserver durch cgi, fastcgi oder Python-Modul benutzt werden oder um schnell Ergebnisse zu erzielen, d.h. eine Karte im Netz, kann der gleich mitgelieferte in Python programmierte Webserver benutzt werden. Da Mapit durch die Systemunabhängige Programmierung in Python sowohl unter GNU/Linux als auch unter Windows läuft können die aufbereiteten Daten problemlos auf diesen Systemen benutzt werden.

---

<sup>1</sup><http://craterlake.wr.usgs.gov/dlg.html>

### 15.1.1. Vorbereitung

Um auf schnellem Wege zu Ergebnissen zu kommen bietet es sich sowohl unter Windows wie auch unter GNU/Linux an die unter <sup>2</sup> vorhandene Schnellinstallation auszuführen die gleichzeitig einen Beispieldatensatz enthält. Diese Beispieldaten ersetzt man Schritt für Schritt mit den eigenen und hat so sehr schnell und einfach beispielsweise eine Karte oder ein Luftbild/Orthophoto über ein Netzwerk präsentiert.

**GNU/Linux** Nachdem sowohl die Mapit- und python-imaging rpms mit *rpm -i Dateiname* installiert wurden kann das Skript *mapitdemo* unter */opt/mapit* ausgeführt werden.

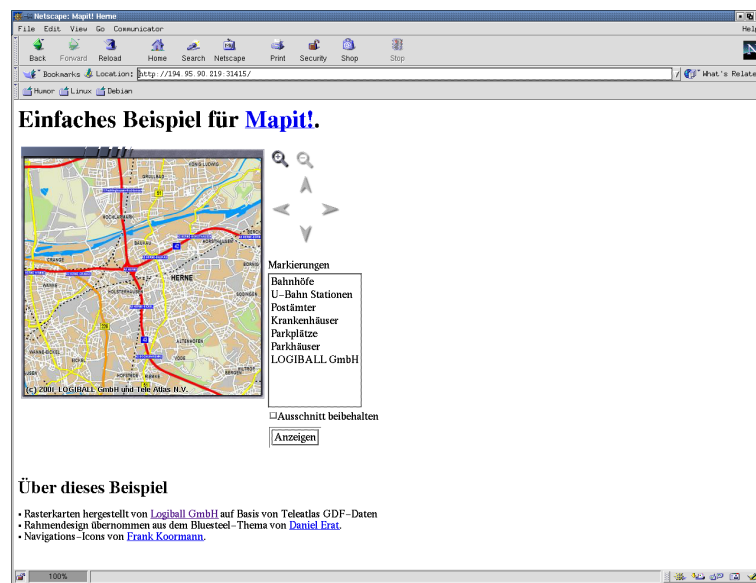


Abbildung 15.1.: Vom MapIt! Webserver generierte Seite mit präsentierter Karte „Herne“

---

<sup>2</sup>[www.mapit.de/download.de.html](http://www.mapit.de/download.de.html)

```

grassuser@xml-2:/opt/mapit > mapitdemo

Open http://127.0.0.1:31415/index.html in your WebBrowser to start.
Hit Ctrl-C twice in this Window to stop.

You will see detailed log information in this window.
If you get an error message, check whether there is an
old mapitdemo still running.
Serving HTTP on port 31415 ...
intevation.de - - [18/Oct/2001 13:57:42] "GET / HTTP/1.1" 200 -
intevation.de - - [18/Oct/2001 13:57:43] code 404, message
File not found
intevation.de - - [18/Oct/2001 13:57:43] "GET /favicon.ico
HTTP/1.1" 404 -
intevation.de - - [18/Oct/2001 13:57:44] "GET /simple/Icons/
top.fancy.png HTTP/1.1" 200 -
intevation.de - - [18/Oct/2001 13:57:44] "GET /simple/Icons/
left_top.fancy.png HTTP/1.1" 200 -
intevation.de - - [18/Oct/2001 13:57:44] "GET /simple/Icons/
right_top.fancy.png HTTP/1.1" 200 -

```

Jetzt kann ausgehend von dem lokalen als auch von jedem Rechner zu dem eine http - Verbindung besteht, die vom Webserver generierte Seite mit der präsentierten Karte aufgerufen werden (Abb. 15.1). Dies geschieht indem man in einem Webbrowser den MapIt! - Webserver auf dem Port 31415 anwählt. Laufen Webserver und Webbrowser auf einem Rechner so ist der Aufruf *http://localhost:31415* oder *http://127.0.0.1:31415* oder auch *http://IP-Adresse des Rechners:31415*. Für einen Rechner im Netzwerk wird die IP-Adresse des Rechners mit angehängtem Port aufgerufen *http://IP-Adresse des Rechners:31415*. Die an der Kommandozeile des Webserver ausgegebenen folgenden Zeilen, zum Beispiel *intevation.de - - [18/Oct/2001 11:29:53] "GET / HTTP/1.0"200 -* geben Auskunft darüber das von einem Rechner der intevation.de Domain aus der MapIt! - Webserver angesprochen wird.

**Windows** Ist die Installationsroutine unter Windows erfolgreich abgeschlossen worden kann aus dem *Startmenu* heraus über den Menüpunkt *MapIt!* der Webserver aufgerufen werden. Gleichzeitig wird der als Grundeinstellung in Windows konfigurierte Webbrowser gestartet. Auch hier gibt der Webserver Informationen über die ankommenden Anfragen an die Kommandozeile aus (Abb. 15.2).

### 15.1.2. Struktur des Beispieldatensatzes

Ein erstellter Datensatz für MapIt! kann sowohl unter Windows als auch unter GNU/Linux verwendet werden. Die Steuerung von MapIt! geschieht durch Konfigurationsdateien.

**Die Konfigurationsdatei *mapconfig.py*** In der Datei *mapconfig.py* wird die grundlegende Konfiguration von MapIt! festgelegt. In der Sektion *The Tile Data* definiert man unter *\_base\_dir = '../examples/herne/'* den Ort des Verzeichnisses in dem die *Tiles* ge-

```

Auswählen MapIt! Starten
Servering HTTP on port 31415
pasithea.intevation.de - - [17/Oct/2001 14:05:17] "GET /index.html HTTP/1.1" 200
-
pasithea.intevation.de - - [17/Oct/2001 14:05:18] "GET /simple/Icons/left_top.fancy.png HTTP/1.1" 200
pasithea.intevation.de - - [17/Oct/2001 14:05:18] "GET /simple/Icons/top.fancy.png HTTP/1.1" 200
pasithea.intevation.de - - [17/Oct/2001 14:05:18] "GET /simple/Icons/right_top.fancy.png HTTP/1.1" 200
pasithea.intevation.de - - [17/Oct/2001 14:05:18] "GET /simple/Icons/border_left.fancy.png HTTP/1.1" 200
pasithea.intevation.de - - [17/Oct/2001 14:05:18] "GET /genmap?x=416764911&y=572286196&width=400&height=400&scale=250000&saneregion=0&autocenter=0 HTTP/1.1" 200
-
pasithea.intevation.de - - [17/Oct/2001 14:05:19] "GET /simple/Icons/border_right.fancy.png HTTP/1.1" 200
pasithea.intevation.de - - [17/Oct/2001 14:05:19] "GET /simple/Icons/left_bottom.fancy.png HTTP/1.1" 200
pasithea.intevation.de - - [17/Oct/2001 14:05:19] "GET /simple/Icons/bottom.fancy.png HTTP/1.1" 200
pasithea.intevation.de - - [17/Oct/2001 14:05:19] "GET /simple/Icons/right_bottom.fancy.png HTTP/1.1" 200
pasithea.intevation.de - - [17/Oct/2001 14:05:19] "GET /simple/Icons/zoomin.fancy.png HTTP/1.1" 200
pasithea.intevation.de - - [17/Oct/2001 14:05:19] "GET /simple/Icons/zoomout.off

```

Abbildung 15.2.: Vom MapIt! Webserver ausgegebene Statusinformationen unter Windows

speichert werden. *scale = 250000* gibt die Auflösungsstufe an die als Grundeinstellung bei jedem Aufruf der Webseite geladen wird. Soll das Zentrum der Darstellung nicht in der Mitte der Bilddatei liegen kann über *default\_x = , default\_y =* ein abweichendes Darstellungszentrum definiert werden. Die Art der Bilddateien wird in der Zeile *tile\_ext = '.png'* in diesem Fall auf .png - Dateien festgelegt.

```

#
# The Tile Data
#
# _base_dir is only used within this file
# this path points to the directory containing the data
# (tiles, markers, etc.)
_base_dir = '../..examples/herne/'

# The default scale to use
scale = 250000

# The default location to use. None means the center of the map
default_x = None
default_y = None

# data_dir is the name of the directory where the map tiles can be
# found.
tile_dir = _base_dir + '/tiles'

# The file extension of the tile files
tile_ext = '.png'

```

Die Sektion *# Output format and size* steuert die Ausgabe des Webservers, unter anderem wird mit *output\_pil\_format = 'JPEG'* die Python Imaging Library(pil) angewiesen als Ausgabeformat ein .jpeg zu erzeugen. *# Default image size in pixels* legt die Größe des Kartenfensters fest das vom Webserver an den Browser geliefert wird.

```

# Output format and size

# The MIME type of the format
output_mime = 'image/jpg'

# The PIL name of the format. This is passed to a PIL Image's save
# method. Obviously the values of output_pil_format and
# output_mime have to match
output_pil_format = 'JPEG'

# The values needed for PNG output
#output_mime = 'image/png'
#output_pil_format = 'PNG'

# Default image size in pixels
image_width = 400
image_height = 400

```

**Das Verzeichnis *tiles*** Die darzustellenden Bilddaten befinden sich in dem Beispieldatensatz in dem Verzeichnis *examples/herne/tiles*. Hier gibt es nun 4 Unterverzeichnisse in denen sich die Bilddaten für jeweils eine Vergrößerungsstufe befinden. Diese Vergrößerungsstufen werden von MapIt! dem numerischen Wert nach geordnet. Die Benennung ist muß nicht dem tatsächlichen Maßstab der Daten folgen, dies hätte wenig Sinn da sich je nach eingestellter Bildschirmauflösung die gröÙe der am Browser ausgegebenen Bilddaten ändert. In diesen „Tiles“ (engl. für Kacheln) - Verzeichnissen befinden sich die Bilddaten in Form von einer oder mehreren Kacheln gleicher GröÙe. Das Auseinanderschneiden in Kacheln ist hilfreich um den Bildaufbau durch den Webserver zu beschleunigen. Es muß nicht mehr das ganze Bild einer Zoomstufe geladen werden um den angeforderten Teilbereich an den Webbrowser zu liefern sondern nur die betroffenen Kacheln. Dieses wird im Ordner */opt/mapit/examples/herne/tiles/10000* deutlich. Die Kacheln werden ausgehend von der linken unteren Ecke aufsteigend in der Form *horizontale Nummerierung x vertikale Nummerierung* . *Dateierweiterung* benannt. *1x1.png* ist also die linke untere Kachel in der das Bildmaterial im PNG - Format gespeichert ist. *5x3.png* wäre somit die Kachel in der dritten Zeile von unten an gezählt die fünfte Kachel vom linken Rand aus gesehen.

**Informationsdatei zu den Kacheln** In jedem Ordner Unterhalb des *tiles* - Verzeichnisses in dem Bilddaten einer Auflösungsstufe gespeichert sind ist eine Informationsdatei vorhanden in der Daten zu den Kacheln gespeichert sind. Der Aufbau dieser *info* - Dateien kann mit jedem Editor betrachtet werden oder auch mit dem Befehl *cat Dateiname*.

```
bash-2.05$ cat info
```

```
2000
2000
# [1,1]
416341578.400
571862863.000
416510911.733
572032196.333
# [1,2]
416341578.400
572032196.333
416510911.733
572201529.667
# [1,3]
416341578.400
572201529.667
416510911.733
572370863.000
```

Die ersten beiden Zahlen bezeichnen die Ausdehnung der einzelnen Kacheln angegeben in Pixel, in diesem Fall also 2000 \* 2000 Pixel. Beginnend mit jedem # wird in den eckigen Klammern eine Kachel benannt für die im folgenden die linke untere und rechte obere Bildecke mit Koordinatenwerten bezeichnet wird. Hierdurch können im weiteren Verlauf Symbole direkt über Koordinatenangaben (in diesem Fall Gauss-Krüger) in der Karte in allen Auflösungsstufen plaziert werden. Da die Auflösungsstufe 100000 Beispielsweise nur durch eine Kachel repräsentiert wird hat die info - Datei die folgende Form. Die Größe der Kachel Nummer  $ixj$  ist 1000 \* 1000 Pixel, in den nächsten 4 Zeilen sind jeweils wieder die Koordinatenpaare der linken unteren und rechten oberen Ecke aufgeführt.

```
1000
1000
# [1,1]
416341578.400
571862863.000
417188245.067
572709529.667
```

**Das Verzeichnis *markers*** In diesem Verzeichnis werden die Symbole im .png - Format gespeichert die in der dargestellten Bilddatei einen Ort kennzeichnen.

**Das Verzeichnis *icons*** Das Verzeichnis Icons beinhaltet alle zur Webseitengenerierung notwendigen Elemente wie Schalter, Rahmen, Pfeile u.s.w. wiederum im .png - Format.



**Die Konfigurationsdatei *markerdefs*** In dem Beispieldatensatz ist diese Datei einmal in deutsch (*matkerdefs.de*) und in englisch (*markerdefs.en*) vorhanden. In Ihnen werden die auszugebenen Objekte definiert und es wird bestimmt wo und wie sie dargestellt werden. Die Datei ist nach Gruppen gegliedert aufgebaut. Leere Zeilen und Zeilen, die mit einem Doppelkreuz (#) anfangen, werden ignoriert. Eine Gruppe wird durch eine Zeile definiert, die mit dem Schlüsselwort "group:" anfängt, gefolgt von vier durch Komma getrennten Werten. Der erste Wert *db* ist der Name des zu verwendenden Symbols aus dem Verzeichnis *markers*. Der zweite Wert bezeichnet die Gruppe mit einem eindeutigen Namen *bahnhof*. in dem Leerraum zwischen den beiden folgenden Kommatas kann eine URL in der Form *http://www.mapit.de* eingegeben werden die beim anklicken eines Objektes der Gruppe aufgerufen wird für das keine spezifische URL angegeben ist. Der vierte Eintrag schließlich ist der Name unter dem die Gruppe auf der generierten Webseite erscheint.

```
group: db,bahnhof,,Bahnhöfe
416837551.50,572293916.00,db,bahnhof,,Herne
416702461.80,572512765.30,db,bahnhof,,Recklinghausen Süd
416738098.40,572018523.00,db,bahnhof,,Bochum-Nokia
416479774.90,572187957.70,db,bahnhof,,WANNE-EICKEL HBF
group: ubahn,ubahn,,U-Bahn Stationen
416851273.00,572306182.00,ubahn,ubahn,,U Herne
416869341.00,572269123.00,ubahn,ubahn,,U Herne Mitte
416823616.00,572396147.00,ubahn,ubahn,,U Schloß Strünkede
416853251.00,572139861.00,ubahn,ubahn,,U Hölkeskampring
416812730.00,572074791.00,ubahn,ubahn,,U Herne-Berninghausstrasse
416895683.00,572206515.00,ubahn,ubahn,,U An der Kreuzkirche
416775106.00,571969359.00,ubahn,ubahn,,U Bochum-Rensingstrasse
416771838.00,571920835.00,ubahn,ubahn,,U Riemke Markt
```

Unterhalb des Gruppensdefinition werden Zeilenweise die Objekte jeder Gruppe aufgeführt. Begonnen wird dabei mit dem jeweiligen Koordinatenpaar das die Position des Objektes angibt. Weiter wird wieder das Symbol aus dem Verzeichnis *markers* angegeben, hier kann durchaus für jedes Objekt einer Gruppe ein unterschiedliches Symbol definiert werden, der Gruppenname *bahnhof*, Platz für einen URL - Aufruf, und als letzte Angabe ein Name für das Objekt der in der Webseite eingeblendet wird wenn der Cursor über dem Objekt steht.

**Die Datei *simple.de.template*** Auch diese Datei gibt es im Beispieldatensatz in einer deutschen *simple.de.template* und englischen *simple.en.template* Version. In ihr wird der Aufbau der HTML - Seite festgelegt die von dem Webserver an den Browser übermittelt wird.

### 15.1.3. Bildkacheln selbst erzeugen

Um aus einer vorhandenen Bilddatei ein *tiles* - Verzeichnis mit beliebig vielen Vergrößerungsstufen zu erzeugen kann man das Bildbearbeitungsprogramm seines Vertrauens bemühen oder man benutzt das Python - Skript *epsct* das sich unter */opt/mapit/src/tools* befindet.

Dieses Skript erzeugt aus einer Postskript (.ps) - Bilddatei in eine angegebene Anzahl von Auflösungsstufen mit vorher zu definierender Kachelanzahl. Gleichzeitig wird eine passende Info - Datei erzeugt, so das anstatt des ursprünglichen *tiles* - Verzeichnisbaumes einfach der neu erzeugte verwendet werden kann.

**Anpassen des *epscut* Skriptes** In Zeile 275 des Skriptes wird die Anzahl der Auflösungsstufen und die jeweilig zu erzeugende Anzahl von Kacheln bestimmt. Bei der Angabe von *factors* = (1, 2, 4, 8) würden also 4 Auflösungsstufen mit jeweils 1, 4, 16 und 64 Kacheln erzeugt. Die Anzahl der Kacheln ergibt sich aus dem Quadrat der angegebenen Faktoren. Im folgenden Beispiel sind als Faktoren *factors* = (1, 2) angegeben, möglich wären aber auch *factors* = (1, 2, 3, 5) was natürlich 1, 4, 9 und 25 Kacheln erzeugen würde. Zum Aufruf wird die zu bearbeitende *179.ps* - Datei einfach mit in das Verzeichnis kopiert indem sich *epscut* befindet. Mit dem Aufruf *./epscut 179.ps* wird das Skript gestartet und liefert eine Folge von Textausgaben an der Kommandozeile die mit der Ausgabe der benutzten Transformationsparameter abschließt. Die erzeugten Kacheln liegen in dem */opt/mapit/src/tools* - Verzeichnis in einem Ordner *179*.

```
bash-2.05$ ./epscut 179.ps
gs -sDEVICE=ppmraw -r72 -dNOPAUSE -dSAFER -q -sOutputFile=/tmp/@821.1
-g400x400 -c "0.035730 0.035733 scale -14.000000 -14.000000
translate /oldshowpage /showpage load def /showpage {}
def" -f179.ps -c oldshowpage quit
writing 179/2/1x1.png
gs -sDEVICE=ppmraw -r72 -dNOPAUSE -dSAFER -q -sOutputFile=/tmp/@821.2
-g400x400 -c "0.071460 0.071467 scale -14.000000 -14.000000
translate /oldshowpage /showpage load def /showpage {}
def" -f179.ps -c oldshowpage quit
writing 179/1/1x1.png
gs -sDEVICE=ppmraw -r72 -dNOPAUSE -dSAFER -q -sOutputFile=/tmp/@821.3
-g400x400 -c "0.071460 0.071467 scale -5611.500000 -14.000000
translate /oldshowpage /showpage load def /showpage {}
def" -f179.ps -c oldshowpage quit
writing 179/1/2x1.png
gs -sDEVICE=ppmraw -r72 -dNOPAUSE -dSAFER -q -sOutputFile=/tmp/@821.4
-g400x400 -c "0.071460 0.071467 scale -14.000000 -5611.000000
translate /oldshowpage /showpage load def /showpage {}
def" -f179.ps -c oldshowpage quit
writing 179/1/1x2.png
gs -sDEVICE=ppmraw -r72 -dNOPAUSE -dSAFER -q -sOutputFile=/tmp/@821.5
-g400x400 -c "0.071460 0.071467 scale -5611.500000 -5611.000000
translate /oldshowpage /showpage load def /showpage {}
def" -f179.ps -c oldshowpage quit
writing 179/1/2x2.png
Transformation from EPS-coordinates (px, py)
to MapIt! coordinates (mx, my):
mx = 0.071460473425636439 * (px - 14)
my = 0.07146685724495265 * (py - 14)
bash-2.05$
```

### 15.1.4. Navigation im Browser

Wird eine von *MapIt!* erzeugte Seite in einem Webbrowser aufgerufen ist die weitere Navigation und Benutzung denkbar einfach. Das Zoomen geschieht über die „Lupen-Icons“ in einer vergrößerten Ansicht kann der Bildausschnitt mit den „Pfeil-Icons“ oder durch anklicken des zu Zentrierenden Punktes in der Karte gewählt werden. Die in der „Markierungsliste“ angewählten Objektklassen (*groups*) werden durch den Button „Anzeigen“ in aufgerufen (Abb. 15.3).

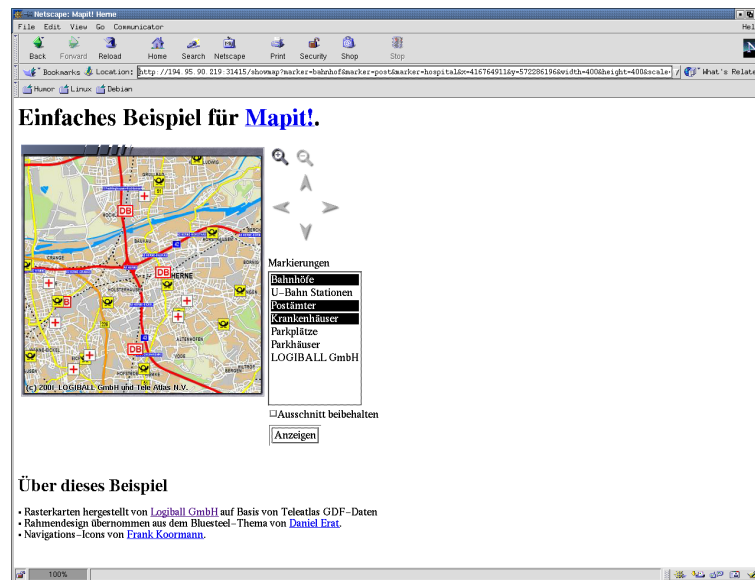


Abbildung 15.3.: Vom MapIt! Webserver generierte Seite mit präsentierter Karte „Herne“ und aufgerufenen Markierungen für „Bahnhöfe“ , „Krankenhäuser“ und „Postämter“

## 15.2. Lösung mit MapServer

Es wurde MapServer 3.5 (nightly build vom 30.Oktober 2001) unter einer SuSE GNU/Linux Installation (7.0) und GRASS 5.0pre2 unter einer Debian GNU/Linux Installation (woody) verwendet.

Sollen nicht nur reine Bilddateien in einem Netzwerk zugänglich gemacht werden sondern auch Vektor- oder Sachdaten so kann man MapServer benutzen. Einen kleinen Einblick in die Vielzahl der mit MapServer möglichen Anwendungen gewinnt man unter <sup>3</sup>. Für das FreeGIS Tutorial greifen wir auf die bekannten Daten des „Craterlake“ zurück <sup>4</sup>.

<sup>3</sup><http://mapserver.gis.umn.edu/gallery.html>

<sup>4</sup><http://craterlake.wr.usgs.gov/>

Hier können neben den Bilddateien auch Vektordaten Beispielsweise in Form von Höhenlinien „Hypsography“ Verwaltungsgrenzen „Boundary“ oder Straßen „Road“ als Shapefiles heruntergeladen werden.

### 15.2.1. Vorbereitung

Zur Installation auf einem GNU/Linux System entpackt man die MapServer Distribution mit dem Kommando `tar -zxfs ms_3.5.tar.gz`, wechselt mit `cd mapserver` in das erzeugte Verzeichnis, erstellt mittels `./configure` ein Makefile. Die dem `configure` angehängten Optionen ergänzen das eigentlich erzeugte Programm um weitere Funktionen wie zum Beispiel das Erkennen und Erzeugen von `.tiff` oder `.png` - Dateien. Eine Liste mit den möglichen Optionen erhält man durch den Befehl `configure --help`.

```
freegis@xml-2:~/tmp/mapserver > configure --with-png --with-jpeg
--with-tiff
loading cache ./config.cache
checking for gcc... (cached) gcc
checking whether the C compiler (gcc ) works... yes
checking whether the C compiler (gcc ) is a cross-compiler... no
checking whether we are using GNU C... (cached) yes
checking whether gcc accepts -g... (cached) yes
checking for c++... (cached) c++
checking whether the C++ compiler (c++ ) works... yes
checking whether the C++ compiler (c++ ) is a cross-compiler... no
checking whether we are using GNU C++... (cached) yes
checking whether c++ accepts -g... (cached) yes
checking for ranlib... (cached) ranlib
checking for flex... (cached) lex
checking for yywrap in -ll... (cached) no
checking for bison... no
checking for byacc... no
checking if compiler supports -R... (cached) no
checking if compiler supports -Wl,-rpath,... (cached) yes
checking for exp in -lm... (cached) yes
checking how to run the C preprocessor... (cached) gcc -E
checking for ANSI C header files... (cached) yes
checking for strcasecmp... (cached) yes
checking for strncasecmp... (cached) yes
checking for strdup... (cached) yes
checking whether we should include JPEG support...
checking for jpeg_read_header in -ljpeg... yes
        using libjpeg from system libs.
.
.
.
checking for PHP/MapScript module options...
        PHP/MapScript module not configured.
creating ./config.status
creating Makefile
```

Nachdem erfolgreichen Erzeugen eines Makefiles wird die Übersetzung des Quelltextes

in ausführbaren Code mit *make* gestartet.

```
freegis@xml-2:~/tmp/mapserver > make
gcc -c -O2 -Wall -DIGNORE_MISSING_DATA -DUSE_EPPL -DUSE_TIFF
-DUSE_JPEG -DUSE_GD_PNG -DUSE_GD_JPEG -DUSE_GD_WBMP -DUSE_GD_FT
-I/usr/local/include mapbits.c -o mapbits.o
gcc -c -O2 -Wall -DIGNORE_MISSING_DATA -DUSE_EPPL
-DUSE_TIFF -DUSE_JPEG -DUSE_GD_PNG -DUSE_GD_JPEG -DUSE_GD_WBMP
-DUSE_GD_FT-I/usr/local/include
und sollte die Ausgabe {\em No query information to decode.

.
.
.
USE_GD_WBMP -DUSE_GD_FT -I/usr/local/include
tile4ms.c -o tile4ms.o
gcc -O2 -Wall -DIGNORE_MISSING_DATA -DUSE_EPPL -DUSE_TIFF
-DUSE_JPEG -DUSE_GD_PNG -DUSE_GD_JPEG -DUSE_GD_WBMP
-DUSE_GD_FT -I/usr/local/include tile4ms.o -L.
-lmap -lgd -L/usr/local/lib -lgd -ljpeg
-lfreetype -lpng -lz -ltiff -ljpeg -lfreetype
-lpng -lz -ljpeg -lm -o tile4ms
freegis@xml-2:~/tmp/mapserver > cp mapserv /usr/local/httpd/cgi-bin/
```

Die so erzeugte binäre "cgi-Datei" *mapserv* wird in das entsprechende Verzeichnis des Web-Servers für „cgi-skripte" kopiert, der hier angegebene Verzeichnispfad */usr/local/httpd/cgi-bin/* ist ein möglicher Ort. Es ist darauf zu Achten, daß dieses "cgi-Skript" von jedermann ausgeführt werden kann. Nötigenfalls können mit *chmod 755 mapserv* die nötige Berechtigung erteilt werden, diese muss natürlich auch für das Verzeichnis */usr/local/httpd/cgi-bin/* gelten.

**Test des cgi-Skriptes** Um das erstellte cgi-Skript und dessen Installation zu testen kann man mit einem beliebigen Browser das Skript direkt aufrufen und sollte die Ausgabe *No query information to decode. QUERY\_STRING is set, but empty.* bekommen (Abb. 15.4).

**Bildmaterial** Als grundlegende Bilddateien können die im Kapitel 9 georeferenzierten Orthophotos und Karten des Gebietes „Craterlake" genutzt werden. alternativ liegen die entsprechenden Dateien dem Tutorial bei oder können von der Webseite heruntergeladen werden. Mapserver unterstützt hierbei zum Beispiel *.tiff* - Dateien mit dazugehörigen *.tfw* - Dateien (Tiff World File) deren Dateiendung allerdings in *.wld* geändert werden muß. Beispielhaft wird nun der Export einer Rasterdatei „karte" mit zugehöriger *.tfw* - Datei aus dem GIS - Programm GRASS beschrieben. Nachdem wie im Kapitel 9 Seiten 71 ff beschrieben die Bilddateien in GRASS importiert und georeferenziert wurden, geschieht der Export als *.tiff* - Datei mit dem Kommando *r.out.tiff -t input=karte output=karte* wobei die Option *-t* das gewünschte *.tfw* Worldfile erzeugt.

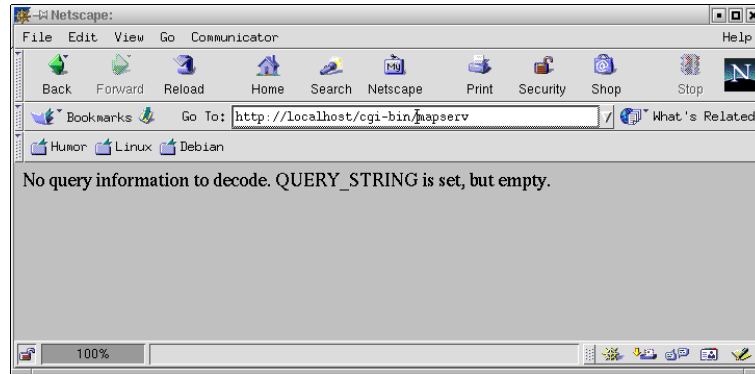


Abbildung 15.4.: Test des Mapserver cgi-Skriptes

```
GRASS:/spare/gisdata/grass/bin > r.out.tiff -t input=karte output=karte
```

Der Aufbau des Worldfile wird deutlich wenn man es betrachtet. Dazu wechselt man mittels *cd* in das entsprechende Verzeichnis und ruft das Worldfile in einem Editor oder einem Dateibetrachter auf.

```
GRASS:/spare/gisdata/grass/bin > r.out.tiff -t input=karte output=karte
GRASS:~ > cd /spare/gisdata/craterlake
GRASS:/spare/gisdata/craterlake > cat karte.tfw

          30.007278020378457
          0.0000000000000000
          0.0000000000000000
        -30.004992867332383
          550855.003639010246843
          4775079.997503566555679

GRASS:/spare/gisdata/craterlake >
```

Die erste Zeile *30.0072* gibt die horizontale Pixelgröße in Metern an, die nächste 2 Zeilen sind Rotations - Koeffizienten die im Normalfall gleich Null sind. In der vierten Zeile ist die vertikale Ausdehnung eines Pixels angegeben, da bei Bilddateien im allgemeinen der Ursprung in der oberen linken Ecke ist hat diese Angabe ein negatives Vorzeichen. Die beiden letzten Zeilen enthalten die Koordinaten des Zentrums des obersten linken Pixels, in diesem Fall UTM Koordinaten der Zone 10.

### 15.2.2. Erstellung der MapServer - Anwendung

In einem für den Webserver zugänglichen Verzeichnis, Beispielsweise */usr/local/httpd/htdocs/mapserver/ms\_de* wird eine Verzeichnisstruktur mit den Verzeichnissen *data*, *graphics* und *symbols* erzeugt. Dies geschieht durch den Befehl *mkdir*. In das Verzeichnis *data* werden die Shapefiles ab-

gelegt, in *graphics* die *.tiff* - Files und *.wld* - Files. Das Umbenennen der *.twf* in *.wld* - Dateien geschieht mittels des Move Kommandos *mv karte.twf karte.wld*. Das Verzeichnis *symbols* enthält die nötige Symboldateien. Weiter werden 3 *.html* Dateien vom cgi - Skript benötigt um mit ihrer Hilfe die Webseiten zu generieren und sie vom Webserver an den Browser zu liefern. Eine mit dem Kommando *dir* erzeugte Dateübersicht sollte ähnlich der folgenden sein.

```
freegis@xml-2:/usr/local/httpd/htdocs/mapserver/ms_demo > dir
insgesamt 40
drwxr-xr-x   4 freegis  users          4096 Okt 30 12:35 data
-rwxr-xr-x   1 freegis  users          1875 Okt 30 15:13 demo.html
-rwxr-xr-x   1 freegis  users          4021 Okt 30 16:44 demo.map
-rwxr-xr-x   1 freegis  users           83 Okt 30 09:46 demo_header.html
-rwxr-xr-x   1 freegis  users          1276 Okt 30 10:49 demo_init.html
drwxr-xr-x   3 freegis  users          4096 Okt 30 15:29 graphics
drwxr-xr-x   2 freegis  users          4096 Okt 30 09:46 symbols
```

**demo\_init.html** Da führende *d* in einigen Zeilen sagt aus das es sich um Verzeichnisse (directories) und nicht um Dateien handelt. Beginnen wir mit der Datei *demo\_init.html*, um sie zu erstellen öffnet man den Editor der Wahl und erzeugt folgenden Text, oder man kopiert die Datei von der dem Buch anliegenden CD oder von der Webseite des Free GIS Tutorials.

```
freegis@xml-2:/usr/local/httpd/htdocs/mapserver/ms_demo > cat demo_init.html

<html>
<head><title>MapServer Demo Interface</title></head>
<body bgcolor="#FFFFFF">

<center><h2>MapServer Demo Interface</h2></center>
<p><hr><p>
Click on the initialize button to fire up the demo.

<p>

<form method=GET action="/cgi-bin/mapserv">

<input type="hidden" name="map" value="/usr/local/httpd
/htdocs/mapserver/demo.map">
<input type="hidden" name="zoomsize" value=2>

<center><input type="submit" value="Initialize"></center>

</form>

<p><hr><p>

</body></html>
```

Wird diese in einem Webbrowser mit `http://localhost/msdemo/mapserver/demo_init.html` aufgerufen sollte sich ein Bild ähnlich (Abb 15.5) ergeben.

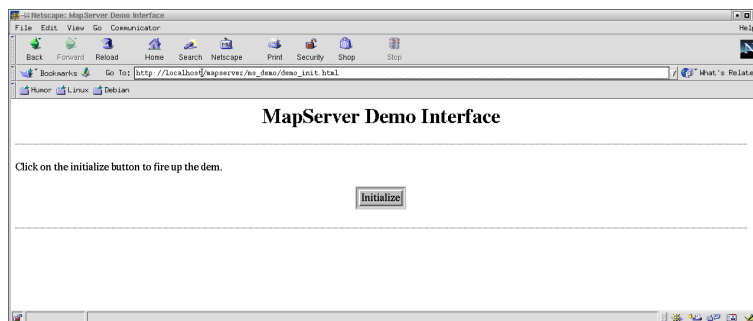


Abbildung 15.5.: Vom Webserver generierte startseite `demo_init.html`

**demo\_header.html** Weiter wird eine Datei `demo_header.html` benötigt, die aber nur den Text der Kopfzeile und die Grundfarbe zu generierenden Webseite enthält. Entsprechend kurz ist der eigentliche Text der Datei.

```
freegis@xml-2:/usr/local/httpd/htdocs/mapserver/ms_demo > cat demo_header.html
<html>
<head><title>MapServer Demo Interface</title></head>
<body bgcolor=#FFFFFF>
```

Die wichtigeren Dateien sind die `demo.html` in der das eigentliche Aussehen der Webseite beschrieben wird und die `demo.map` die die Verbindungen zu den darzustellenden Daten definiert. Zuerst die `demo.html` die ein Template also ein Muster oder eine Schablone der Webseite beinhaltet.



```
freegis@xml-2:/usr/local/httpd/htdocs/mapserver/ms_demo > cat demo.html
```

```
<html>
<head><title>MapServer Demo Interface</title></head>
<body bgcolor=#FFFFFF>

<center><h1>MapServer Demo Interface</h1></center>
<hr>
<form method=GET action="/cgi-bin/mapserver/mapserv">

<center>
<table border=2 cellpadding=10 width=100%>
<tr>
<td align=center>
<INPUT NAME="img" TYPE="image" SRC="[img]" width=600 height=600</A><p>

</td>
<td>

<input type="submit" value="Refresh/Query"><p>

<input type="radio" name="mode" value="browse" checked>
<b>Browse map</b><br>
<input type="radio" name="mode" value="query"> <b>Query feature</b><br>
<input type="radio" name="mode" value="nquery">
<b>Query multiple features</b>

<hr>
<center><a href="/scripts/mapserv?map=[map]&mapext=[minx]+[miny]+[maxx]+[maxy]
&mode=nquery&[get_layers]">summarize</a></center>
<hr>

<p>
<b>Select Layers to Display: </b><br>
<select multiple name="layer" size=3>
<option value="crater" [crater_select]> Craterlake
<option value="craterluft" [craterluft_select]> Craterluft
<option value="contour" [contour_select]> Cratercontour
<option value="road" [road_select]> Road
<option value="boundaryline" [boundaryline_select]> Boundaryline
<option value="boundarypolyline" [boundarypolyline_select]>
Boundarypolyline
</select>
```

```

<p>
Zoom In <input type=radio name=zoomdir value=1 [zoomdir_1_check]>
Pan <input type=radio name=zoomdir value=0 [zoomdir_0_check]>
Zoom Out <input type=radio name=zoomdir value=-1 [zoomdir_-1_check]>
<p>
Zoom Size <input type=text name=zoomsize size=4 value=[zoomsize]>
<p>

<font size=+1><b>Legend</b></font><br><hr>
<hr>

<p>
<INPUT NAME="ref" TYPE="image" SRC="[ref]" border="0"></A><p>

</td>
</tr>
</table>
</center>

<input type="hidden" name="imgxy" value="299.5 299.5">
<input type="hidden" name="imgext" value="[mapext]">
<input type="hidden" name="map" value="[map]">

</form>

<p><hr><p>

</body></html>

freegis@xml-2:/usr/local/httpd/htdocs/mapserver/ms_demo >

```

Jede einzelne Zeile erklären zu Wollen würde den Rahmen diese Kapitels sicherlich sprengen, deshalb gehe ich nur auf die Zeilen ein deren Editierung mir für diese MapServer Anwendung wichtig erscheinen. Zu nennen wären hier:

```

<option value="crater" [crater_select]> Craterlake

```

Mit dieser Zeile wird ein "Layer" aus der *demo.map* Datei angegeben dessen Darstellung in Mapserver angewählt werden kann. Das „crater“ ist hierbei der Name des „Layers“ während „Craterlake“ der in MapServer erscheinende Name im Auswahlmenü ist. Betrachtet man die Datei *demo.html* im Browser ohne Bearbeitung durch das cgi - Skript sähe sie so aus (Abb. 15.6). Um diese leere Schablone nun mit Daten zu füllen wird vom Webserver, oder besser vom cgi - Skript, das der Webserver ausführt, die Datei *demo.map* gelesen und die dort angegebenen Daten in die *demo.html* eingetragen. Die *demo.html* für dieses Beispiel sieht folgendermaßen aus.

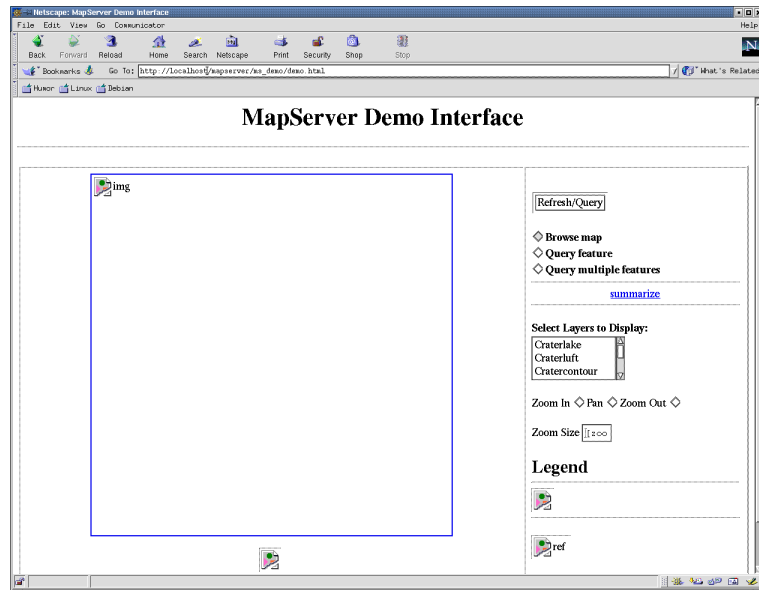


Abbildung 15.6.: Ausgabe der demo.html Datei im Webserver

```

freegis@xml-2:/usr/local/httpd/htdocs/mapserver/ms_demo > cat demo.map

# Start of map file
NAME DEMO
STATUS ON
SIZE 600 600
IMAGETYPE PNG
EXTENT 550840.000000 4733028.000000 592070.000000 4775095.000000
UNITS METERS
SHAPEPATH "/usr/local/httpd/htdocs/mapserver/ms_demo/data/"
IMAGECOLOR 255 255 255

#LABELOVERLAP FALSE

# Start of web interface definition
WEB
  HEADER demo_header.html
  TEMPLATE demo.html
  MINSCALE 100
  MAXSCALE 1550000
  IMAGEPATH "/usr/local/httpd/htdocs/mapserver/tmp/"
  IMAGEURL "/mapserver/tmp/"
  LOG "/tmp/mapserver.log"
END

```

```

# Start of reference map
REFERENCE
  IMAGE graphics/karte.png
EXTENT 550840.000000 4733028.000000 592070.000000 4775095.000000
  SIZE 169 172
  STATUS ON
  COLOR -1 -1 -1
  OUTLINECOLOR 255 0 0
END

# Start of legend
LEGEND
  KEYSIZE 18 12
  LABEL
    TYPE BITMAP
    SIZE MEDIUM
    COLOR 0 0 89
  END
  STATUS ON
END

# Start of scalebar
SCALEBAR
  IMAGECOLOR 255 255 255
  LABEL
    COLOR 0 0 0
    SIZE SMALL
  END
  SIZE 350 5
  COLOR 255 255 255
  BACKGROUNDCOLOR 0 0 0
  OUTLINECOLOR 0 0 0
  UNITS KILOMETERS
  INTERVALS 5
  STATUS ON
END

```

```

# Start of layer definitions

LAYER
  NAME karte
  TYPE RASTER
  STATUS ON
  DATA karte.tif
  OFFSITE 0
END#LAYER

LAYER
  NAME road
  TYPE LINE
  STATUS ON
  DATA roads1

  CLASS
    NAME "Road"
    COLOR 255 0 0
  END
END # Layer

END # Map File

freegis@xml-2:/usr/local/httpd/htdocs/mapserver/ms_demo >

```

**„Layer“ für Rasterdaten** Ein „Layer“ für eine Rasterdatei also zum Beispiel die ins *.tiff* - Format exportierte Karte des USGS wird durch die *LAYER* - Zeile begonnen und mit einem *END* abgeschlossen. *NAME karte* bezeichnet den Namen des „Layers“ *TYPE RASTER* gibt an das es sich in diesem Fall um eine Rasterdatei handelt. Mit der Zeile *STATUS ON* bzw. *STATUS OFF* kann die Darstellung des Layers ein oder auch abgeschaltet werden. *DATA karte.tif* gibt den Dateinamen der darzustellenden Bilddatei an.

**„Layer“ für Shapefiles** Um die Shapefiles des Craterlake zusammen mit der georeferenzierten Rasterdatei verarbeiten zu können werden für sie ebenfalls Layer angelegt. Als Beispiel sei hier der „Layer“ *road* angeführt. Im Unterschied zu einem Rasterlayer hier *TYPE LINE* angegeben da es sich um eine Vektordatei handelt. Die Zeile *DATA roads1* gibt den entsprechenden Dateinamen des Shapefiles an. Um in der Legende einen Entsprechenden Eintrag für die dargestellten Vektoren zu erzeugen, die in diesem Fall Straßen darstellen ist dem „Layer“ noch mindestens eine *CLASS* zuzuordnen. Das Schlüsselwort *CLASS* eröffnet einen entsprechenden Eintrag in der *demo.map*, anschließend wird der Bezeichner des Legendeneintrages mit der Zeile *NAME „Road“* bestimmt. Die Farbe der Einträge wird durch

die Zeile `COLOR 255 0 0` bestimmt.

**Anzeige im Browser** Als Ergebnis kann im Browser nun eine Karte optional mit den Straßen, die im Shapefile gespeichert sind überlagert werden. Dazu werden im Menü *Select Layers to Display*: die Layer „Karte“ und „Road“ ausgewählt. Mit einem Klick auf *Refresh/Query* wird die neu Auswahl an den Webserver übermittelt und die entsprechende Antwort wird im Browser angezeigt (Abb. 15.6).

### 15.2.3. Einbinden der erzeugten Höhenlinien aus Kapitel 7 Seite 63

Um die mit GRASS erzeugten Höhenlinien in Mapserver einzubinden werden die im GRASS-Vektorformat vorliegenden Höhenlinien in Shapefiles konvertiert. Nachdem GRASS wie in Kapitel 6.4 beschrieben geöffnet wurde wird das Kommando `v.out.shape` zum Datenexport benutzt. Übergeben werden dem Kommando dabei der Name der GRASS-Vektordatei `map=` und der Name des auszugebenden Shapefiles mit dem Verzeichnis in das geschrieben werden soll `prefix=`.

```
GRASS:/spare/gisdata/grass/bin > v.out.shape map=hlinien prefix=/spare/gisdata/test/craterlake
```

Die erzeugten Dateien `hoehenlinien.shp`, `hoehenlinien.shx`, `hoehenlinien.dbf` werden in das `data`-Verzeichnis von MapServer kopiert

```
GRASS:/spare/gisdata/test/craterlake/tutorial/ > cp hoehenlinien.shp/  
/usr/local/httpd/htdocs/mapserver/ms_demo/
```

In die `demo.map` datei wird folgende Zeile eingefügt um die Höhenlinien für MapServer zugänglich zu machen.

```
LAYER  
  NAME hlinien  
  TYPE LINE  
  STATUS ON  
  DATA hoehenlinien  
  
  CLASS  
    NAME "Höhenlinien"  
    COLOR 255 0 0  
  END  
END # Layer
```

Damit in dem Auswahlmenü der MapServer-Webseite der Eintrag für diese Höhenlinien auftaucht wird die `demo.html` durch die Zeile `<option value="hlinien"[hlinien_select]> Höhenlinien` folgt ergänzt.

```

<b>Select Layers to Display: </b><br>
<select multiple name="layer" size=3>
<option value="crater" [crater_select]> Craterlake
<option value="craterluft" [craterluft_select]> Craterluft
<option value="contour" [contour_select]> Cratercontour
<option value="road" [road_select]> Road
<option value="boundaryline" [boundaryline_select]> Boundaryline
<option value="boundarypolyline"
[boundarypolyline_select]> Boundarypolyline
<option value="hlinien" [hlinien_select]> Höhenlinien
</select>

```

Anschließend können die „GRASS-Höhenlinien“ zusammen mit der Karte, Luftbild u.s.w. in MapServer angezeigt werden. Interessant ist auch ein Vergleich der „GRASS-Höhenlinien“ mit den vom USGS gelieferten „Cratercontour“-Linien.

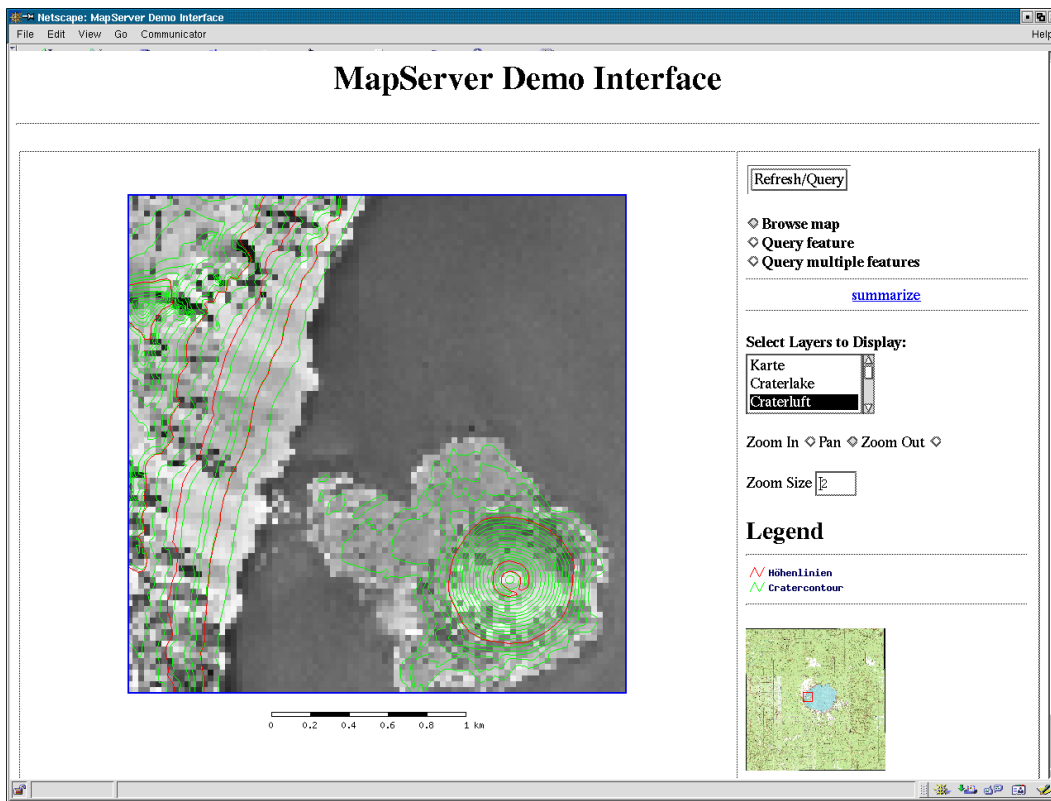


Abbildung 15.7.: Vergleich der GRASS-Höhenlinien (rot) mit den USGS-Conturlines (grün) in MapServer

## 15.2.4. Erweiterung der Anwendung

Zu dem eigentlichen Shapefile gehört grundsätzlich auch eine Datei im *.dbf* Datenbankformat, die beliebig viele Daten zu dem Shapefile beinhalten kann. Diese Daten können von Mapserver direkt ausgelesen und verarbeitet d.h. auf Anfrage im Netz dargestellt werden. Gesteuert wird diese Funktionalität durch Einträge in der *demo.map* und drei weitere Templatedateien, in unserem Beispiel die *streetspy.html*, die *streetspyheader.html* und die *streetspyfooter.html*. Um aus den *.dbf* - Dateien Daten verwenden zu können ist es natürlich Notwendig zu wissen was für Daten in ihnen gespeichert werden und vor allem wie sie gespeichert sind. Da heutzutage Tabellenkalkulationsprogramme solche DBASE (*.dbf*) - Dateien auslesen können kann man sie sich auf diese Weise anzeigen lassen. Die als Beispiel verwendete Datei *roadsl.dbf* enthält den Datensatz Nr. 44 mit folgenden Feldern.

```
Record: 44
FNODE_:      80
TNODE_:      72
LPOLY_:      20
RPOLY_:      22
LENGTH:      964.58496
ROADS_:      45
ROADS_ID:    44
ENTITY_LAB:  1700209
ENTITY_DEF:  CLASS 3
PHOTOREVIS:
RELATION_T:
VERTICAL_R:
BANK:
OPERATIONA:
SALT:
UNSURVEYED:
INTERMITTE:
SUBMERGED:
DRY:
MINERAL_OR:
NAVIGABLE:
EARTHEN:
INTERPOLAT:
ELEVATION:      0.000
ROTATION_A:    0
RIVER_MILE:      0.000
BEST_ESTIM:
COMPOSITIO:
TIDE:
UNDREDEDGED:
```

**streetspyheader.html** In dieser Datei werden die darzustellenden Überschriften der Tabellenspalten definiert.



```

<font size+1><b>Layer: Road</b></font><p>
<table cellpadding=5 cellspacing=2 border=0>
<tr bgcolor=#CCCCCC><th>Straßen Identifizierer</th>
<th>Zustandsstufe/Klassifizierung der Straße</th>
<th>Straßenlänge</th></tr>

```

**streetspy.html** Hier werden den Tabellenspalten die jeweiligen Inhalte aus der *.dbf* - Datei zugewiesen indem das entsprechende Feld aufgerufen wird. Zusätzlich können hier wie in jeder anderen HTML - Seite z.B. Hyperlinks gesetzt werden.

```

<tr><td>[ROADS_ID]</td><td>[ENTITY_DEF]</td><td>[LENGTH]</td></tr>

```

**streetspyfooter.html** Mit dieser Datei wird die Tabelle beendet und zur besseren Übersicht werden noch zwei Kartenfenster generiert.

```

</table><p>
<font size+1 face=arial,Helvetica><b>Query Map Examples</b></font>
<p>

<table cellpadding="3" cellspacing="0" border="0">
  <tr><td></td>
  <td></td></tr>
  <tr><th align="center">standard querymap</th>
  <th align="center">cached query</th></tr>
</table>

</body>
</html>

```

Abschließend muß nun noch die *demo.map* modifiziert werden. So werden dem „Layer“ *Road* die Namen der neu erzeugten Templatedateien hinzugefügt, die Zeile *TOLERANCE* 3 gibt an, daß der User bis zu 3 Pixel daneben Klicken kann und trotzdem die gewünschte Information angezeigt bekommt.

```

LAYER
  NAME road
  TYPE LINE
  STATUS ON
  DATA roads1

  CLASS
    NAME "Road"
    TEMPLATE "streetspy.html"
    COLOR 255 0 0
  END

  HEADER "streetspyheader.html"
  FOOTER "streetspyfooter.html"
  TOLERANCE 3
END # Layer

```

Um die Übersichtskarten in der Abfrageseite generieren zu können werden ihre Ausmaße durch folgende Zeilen festgelegt, die beispielsweise in dem Abschnitt "# Start of web interface definition" gespeichert werden kann. Die Zeile *COLOR 255 255 0* beschreibt die Farbe des ausgewählten Objektes in der Übersichtskarte, diese sollte sich von der eigentlichen Darstellungsfarbe des "Layers" *roads* unterscheiden. Die Größe der Fenster kann durch die Zeile *SIZE 200 200* gesteuert werden.

```

QUERYMAP
  SIZE 200 200
  STATUS ON
  STYLE HILITE
  COLOR 0 0 255
END

```

Nachdem MapServer neu in einem Webbrowser gestartet wurde kann eine Abfrage durch auswählen des Knopfes *Query feature* und anschließendem Mausklick auf ein entsprechendes Objekt im Kartenfenster von MapServer ausgelöst werden. Im besprochenen Fall können also die drei ausgewählten Einträge angezeigt werden wenn man die gewünschte Straße in einem Abstand von höchstens 3 Pixeln anklickt. Die erzeugte Webseite sollte ähnlich der Folgenden sein (Abb. 15.8).

Die so erstellten Seiten können natürlich mit allen Möglichkeiten die HTML bietet weiter ausgebaut werden. Auch sind die Möglichkeiten von WEBSERVER hier nur ansatzweise benutzt worden, wie man an den am Anfang angegebenen Beispielen sieht. Ausgehend von dem hier besprochenen Beispiel können die vielfältigsten Anwendungen erstellt werden. Auf der MapServer - Homepage<sup>5</sup> finden sich eine Vielzahl von Dokumentationen.

---

<sup>5</sup><http://mapserver.gis.umn.edu/>

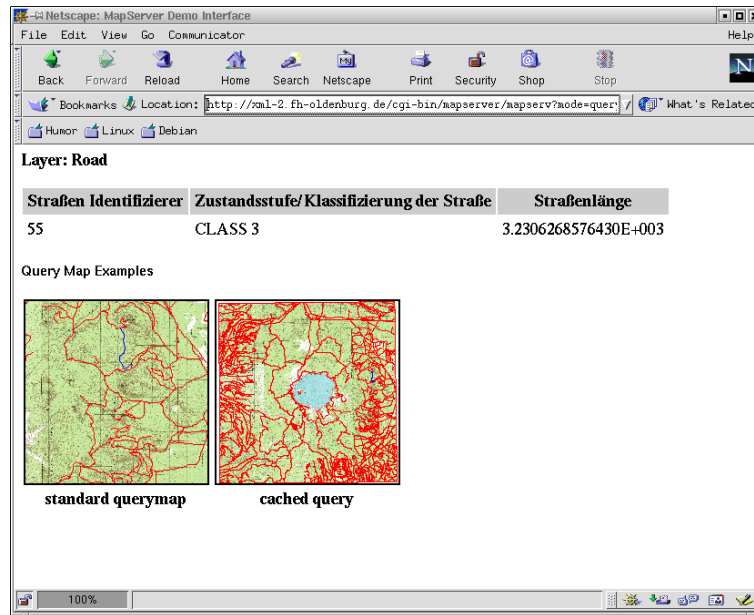


Abbildung 15.8.: Ausgabeseite der Abfrage in MapServer

**Tipps** Durch die Definition von *IMAGEPATH* `"/usr/local/httpd/htdocs/mapserver/tmp/"` als Verzeichnis in das Mapserver die erzeugten Bilddateien für den Webserver schreibt, wird sich dieses Verzeichnis im Laufe der Zeit mit Bilddaten füllen die nicht mehr benötigt werden. Um die zu verhindern, kann auf Rechnern mit dem Betriebssystem GNU/Linux ein einfaches Shell-Skript geschrieben werden, das in diesem Verzeichnis alle Dateien löscht die älter als beispielsweise 60 min sind. Erstellt wird ein solches Skript mit einem beliebigen Editor.

```
freegis@xml-2:/usr/local/httpd/htdocs/mapserver > vim ex.sh
```

Dieses Kommando ruft den Editor *vim* an der Kommandozeile auf. Das eigentliche Skript besteht hier in aus einem Kommando, dem verschiedene Parameter mitgegeben werden und die es veranlassen alle Dateien deren Name mit Demo beginnt und die älter als 60 min sind zu löschen.

```
#!/bin/sh
# löscht die von mapserver erzeugten dateien in
# /usr/local/httpd/htdocs/mapserver/tmp/ die älter als 60 min sind

find /usr/local/httpd/htdocs/mapserver/tmp/ -name 'DEMO*.*' -type f/
-cmin +60 -exec rm -f -- '{}' ';'
```

Die erste Zeile gibt an, daß als Kommandointerpreter die sh - Shell benutzt wird. Die nächsten zwei Zeilen sind Kommentare die die Funktion dieses Skriptes beschreiben. Der



# 16. EDBS Daten verarbeiten

von Otto Dassau

## Aufgabenstellung

An dieser Stelle soll der Import von ATKIS/EDBS Daten vorgestellt werden. Aufbauend auf den schon seit längeren existierenden EDBS\_extra Reader von Claus Rinner gibt es seit ein paar Wochen ein Import Modul, dass einzelne oder auch Gruppen von EDBS-Dateien direkt in eine Location im GRASS Vektor- bzw. Sites-Format importieren kann.

## 16.1. Lösung mit GRASS

Es wurde GRASS 5.0pre2 mit dem Paket *grass\_v.in.edbs.tar.gz* unter einer SuSE GNU/Linux Installation (7.1) verwendet.

### 16.1.1. Das EDBS-Format

Das Datenformat EDBS ist entwickelt worden, um eine einheitliche Datenbankschnittstelle für Daten des ALK "Automatisierte Liegenschaftskarte" und des ATKIS "Amtliches Topographisch-Kartographisches Informationssystem" bereitzustellen. Vergleichbar sind diese Bemühungen mit dem SDTS Format des U.S. Geological Survey für den nordamerikanischen Raum.

Da sich im GIS-Bereich bis heute keine für Deutschland allgemein gültigen Schnittstellenstandards durchsetzen konnten, sind die Standards hierzulande allerdings oft länderspezifisch.

Im Bereich ALK wird das EDBS-Format in Nordrhein-Westfalen, Hessen, Niedersachsen, Schleswig-Holstein, Berlin, Brandenburg, Bremen, Rheinland-Pfalz, Sachsen-Anhalt und Schleswig-Holstein verwendet

ATKIS-Daten werden von allen Bundesländern (bis auf das Saarland) in EDBS-Form angeboten. In Baden-Württemberg werden Daten des Landesvermessungsamtes bislang im SICAD-SQD-Format oder BGRUND-Format [Baden 1990] bereitgestellt. In Bayern steht neben dem SQD-Format das DFK- Schnittstellenformat zur Verfügung [Bayern 1993]; mittlerweile wird auch dort das EDBS-Format unterstützt.

Die Definition der EDBS erfolgte im Rahmen des Gesamtsystems "Automatisierung der

Liegenschaftskarte" (ALK-System). Die Struktur ist textbasiert, vektororientiert und beinhaltet die vollständige Topologie.

Inhalte sind u.a. :

- Objektnummer A999525
- Objektteilnummer 1
- Objektart 3102 (= Weg)
- Folie 104
- Objekttyp L(inie)
- Labelpunkt (3418315.94,5765981.04)
- Entstehungsdatum 11.11.1992 sowie Anfangspunkt (3418583.34,5765292.92)
- Zwischenpunkt (3418644.90,5765115.08)
- Endpunkt (3418761.27,5765110.19)

### 16.1.2. Der EDBS\_extra Reader

Das Programm EDBS\_extra wurde am Institut für Umweltsystemforschung der Universität Osnabrück von Claus Rinner entwickelt.

Dabei handelt es sich um ein ANSI-C-Programm, das EDBS-Aufträge mit ALK- oder ATKIS-Daten abarbeitet, und die darin enthaltenen Geometrie-, Topologie- und Sachinformationen 'lesbar' abspeichert, wobei die Geometriedaten für den Import mit der Funktion GENERATE von Arc/Info vorbereitet sind. Weitere Erläuterungen hierzu und Informationen zum EDBS-Format sind unter <sup>1</sup> nachzulesen.

### 16.1.3. Import von EDBS Daten nach GRASS

Im Folgenden soll der Import von EDBS-Daten nach GRASS näher beschrieben werden. Aufbauend auf den EDBS\_extra Reader gibt es ein Import Modul **v.in.edbs**, das diesen mit einbezieht und die im ARC/Info (Ungenerate) Format vorliegende Datei(en) ins GRASS Vektor- bzw. Punktformat importiert.

Dieses Import Modul ist momentan nicht in GRASS integriert, sondern liegt als Zusatzpaket im "Add-ons" Bereich der GRASS-Homepage. Es befindet sich noch in der Testphase, und wurde bislang nur an wenigen Daten getestet. Auf der Internetseite <sup>2</sup> ist ein Link auf das Programm mit einer kurzen Beschreibung vorhanden.

Das Modul v.in.edbs mit seinen Unterprogrammen kann auf der oben genannten Internetseite heruntergeladen werden.

In dem Paket **grass\_v.in.edbs.tar.gz** sind die EDBS\_extra Programme *edx*, *dom*, *extra\_run* mit den Verzeichnissen *edx\_files* und *dom\_files*. In diesen werden die von EDBS\_extra ausgelesenen Daten im ARC/Info Format abgelegt. Desweiteren enthalten sind ein Perl-Skript *edbs\_split*, das den direkten Import der einzelnen Objektarten ermöglicht, und das eigentliche import Modul *v.in.edbs*.

---

<sup>1</sup><http://www.riners.de/edbs/>

<sup>2</sup>[http://grass.itc.it/grass\\_addons.html](http://grass.itc.it/grass_addons.html)

**Installation von v.in.edbs** Nach dem Download sind folgende Schritte als admin durchzuführen:

```
cp grass_v.in.edbs.tar.gz $GISBASE
cd $GISBASE
gunzip grass_v.in.edbs.tar.gz
tar xfv grass_v.in.edbs.tar.gz
```

Das Verzeichnis \$GISBASE entspricht dem GRASS Programmverzeichnis (meist ist dies /usr/local/grass5/). Danach steht das Programm unter GRASS zu Verfügung. Getestet wurde es bislang nur auf einem Linux Rechner, zur Lauffähigkeit auf anderen Systemen ist noch nichts bekannt.

Während des Imports der EDBS Datei wird zwischen linien-, flächen- und punkthaften Daten unterschieden. Der Import ist daher in 3 Bereiche unterteilt:

1. Import von linienhaften Strukturen
2. Import von flächenhaften Strukturen
3. Import von punkthaften Strukturen

Diese werden während des Imports nacheinander durchlaufen.

**Import von linienhaften Daten** Gestartet wird das Modul nach der Installation mit dem Befehl:

```
GRASS:~ > v.in.edbs atkis.bsp
```

Die hier verwendete Beispieldatei 'atkis.bsp' ist eine frei im Internet verfügbare Musterdatei, und kann u.a. über die EDBS\_extra Seite von Claus Rinner erreicht werden. Zu Beginn des Imports wird gefragt, ob eine einzelne oder mehrere EDBS-Dateien in die GRASS Location importiert werden sollen.

```
Does the importfile contain more than one EDBS-file? (y/n)
```

Wenn mehrere EDBS-Dateien gleichzeitig in eine Location importiert werden sollen, muss vor dem Import eine Datei erstellt werden, in der die kompletten Pfade der einzelnen zu importierenden Dateien aufgelistet sind.

Der Inhalt einer solchen Datei kann z.B. folgendermaßen aussehen:

```
/home/user/edbsdaten/beispiel.ebs
/home/user/edbs/edbs_import/atkis.bsp
.....
```

Je nach Angabe wird nun die einzelne EDBS-Datei oder die aus einer Gruppe von EDBS-Dateien bestehende Importdatei eingelesen. Danach werden die in den EDBS-Daten vorhandenen linienhaften Objekte aufgelistet. Diese sind unterteilt in Objektbereiche (Siedlung, Verkehr, Vegetation, ...) und die dazugehörigen Objektarten (Strasse, Grenze,...).

```
EXISTING LINE OBJECT GROUPS:

SIEDLUNGEN:
VERKEHR:
3101 Strasse
3102 Weg
3105 Strassenkoerper
3106 Fahrbahn
3201 Schienenbahn
3514 Bruecke_Unterfuehrung_Ueberfuehrung
3531 Kabelleitung
VEGETATION:
4203 Hecke_Knick
GEWAESSER:
5101 Strom_Fluss_Bach
5103 Graben_Kanal
RELIEF:
GEBIETE:
7299 Grenze

import OBJEKTART or OBJEKTBEREICH? [OBJART]
[BEREICH or OBJART]
```

Dabei erfolgt eine Abfrage, ob die zu importierenden Daten als “Objektbereichs-” oder “Objektartkarten” ins GRASS Vektorformat überführt werden sollen.

### Objektbereich

Entscheidet man sich für den Import als Objektbereichskarten, so erhalten die importierten Vektorkarten den Namen des jeweiligen Bereichs mit der Endung “\_l”.

- siedlung\_l
- verkehr\_l
- vegetation\_l
- gewaesser\_l
- relief\_l
- gebiete\_l

In der Karte ‘gewaesser\_l’ sind dann z.B. die Objektarten Strom\_Fluss\_Bach und Graben\_Kanal enthalten.

### Objektart



Entscheidet man sich für den Import der Linienzüge als Objektartkarten, so werden die einzelnen Objektarten (Strasse, Weg,...) jeweils als einzelne Vektorkarten importiert. Da hier keine Überschneidung der linien- und flächenhaften Kartennamen entsteht, erfolgt keine weitere Kennzeichnung, ob es sich um Karten mit Linien oder Polygonen handelt.

**Import von flächenhaften Daten** Bei den flächenhaften Daten läuft der Import entsprechend ab. Nach einer Bestätigung werden die in der EDBS-Datei vorhandenen Polygone aufgelistet.

Auch hier wird wieder nach Objektbereiche und den darin enthaltenen Objektarten unterschieden.

```
existing POLYGON GROUPS:

SIEDLUNGEN:
2101 Ortslage
2111 Wohnbauflaeche
2113 Flaeche_gemischter_Nutzung
2114 Flaeche_besonderer_funktionaler_Praegung
2201 Sportanlage
2222 Sportplatz
2228 Campingplatz
VERKEHR:
3514 Bruecke_Unterfuehrung_Ueberfuehrung
VEGETATION:
4101 Ackerland
4102 Gruenland
4107 Wald_Forst
4108 Gehoelz
GEWAESSER:
5112 Binnensee_Stausee_Teich
RELIEF:
GEBIETE:
7101 Verwaltungseinheit

import OBJEKTART or OBJEKTBEREICH? [OBJART]
[BEREICH or OBJART]
```

Es erfolgt wieder eine Abfrage, ob die zu importierenden Daten als “Objektbereichs-” oder “Objektartkarten” ins GRASS Vektorformat überführt werden sollen.

### Objektbereich

Entscheidet man sich für den Import der Polygone als Objektbereichskarten, so erhalten die importierten Vektorkarten den Namen des jeweiligen Bereichs mit der Endung “\_f”.

- siedlung\_f
- verkehr\_f
- vegetation\_f
- gewaesser\_f

- relief\_f
- gebiete\_f

In der Karte 'vegetation\_f' sind die Objektarten Ackerland, Gruenland, Wald\_Forst und Gehoelz enthalten.

### Objektart

Entscheidet man sich für den Import als Objektartkarten, so werden diese (Ortslage, Gehoelz,...) jeweils als einzelne Vektorkarten importiert. Da hier keine Überschneidung der linien- und flächenhaften Kartennamen entsteht, erfolgt keine weitere Kennzeichnung, ob es sich um Linien oder Polygone handelt.

**Import von punkthaften Daten** Der Import der Punktdaten findet am Ende des Imports statt. Hierbei werden wie im vorigen Verlauf die in der EDBS -Datei vorkommenden Punktdaten angezeigt.

```
existing POINT DATA:

3541 Mast

import point data? [Return]

Mast imported to ~/grassdata/sh_edbs/sh_edbs/site_lists/...
```

Hier gibt es keine Unterscheidung zwischen Objektbereich und dazugehörigen Objektarten. Der Import findet automatisch als Objektart statt.

Wie oben zu sehen, werden die Punktdaten, die unter GRASS als 'Sites' bezeichnet werden in das Verzeichnis \$LOCATION/site\_lists importiert. Dieses wird, wenn es nicht schon vorhanden ist erzeugt.

### **16.1.4. Ergebnis**

Wenn der Import der Daten abgeschlossen ist, kann man sich die Liste der nun in GRASS befindlichen Vektorkarten mit dem Befehl 'g.list.vect' anschauen.

```

GRASS:~ > g.list vect
-----
vector files available in mapset sh_edbs:
Ackerland                               Graben_Kanal
Binnensee_Stausee_Teich                 Grenze
Bruecke_Unterfuehrung_Ueberfuehrung    Gruenland
Campingplatz                            Hecke_Knick
Fahrbahn                                Kabelleitung
Flaeche_besonderer_funktionaler_Praegung  Ortslage
Flaeche_gemischter_Nutzung             relief_f
gebiete_f                                relief_l
gebiete_l                                Schienenbahn
Gehoelz                                  siedlung_f
gewaesser_f                              siedlung_l
gewaesser_l                              Sportanlage
Sportplatz                              vegetation_l
Strasse                                   verkehr_f
Strassenkoerper                         verkehr_l
Strom_Fluss_Bach                        Verwaltungseinheit
vegetation_f                             Wald_Forst
Weg                                       Wohnbauflaeche
-----

```

Die Punktdaten liegen im Verzeichnis \$LOCATION/site\_lists. Eine Liste der vorhandenen Sites-Daten kann mit dem Befehl 'g.list.sites' ausgegeben werden.

```

GRASS:~ > g.list sites
-----
site list files available in mapset sh_edbs:
Mast  Verkehrsknoten  Verwaltungseinheit
-----

```

Zur Veranschaulichung sind hier 2 Karten dargestellt, in denen alle 3 Bereiche der importierten EDBS-Daten (Linien, Polygone und Punkte) als Vektor- und als Rasterkarte dargestellt sind. Durch den Befehl 'd.what.vect' bzw. 'd.what.rast' können die Attribute der Punkte, Linien und Flächen abgefragt werden.

In der Vektorkarte ist der Objektbereich 'gewaesser\_f', die Objektart 'Strasse' und die Objektart 'Mast' dargestellt. Der X-term enthält die Abfrageergebnisse.

In der Rasterkarte ist der Objektbereich 'gewaesser\_f', die Objektart 'Weg' und die Objektart 'Mast' dargestellt. Der X-Term enthält die Abfrageergebnisse. Die Umwandlung der Vektordaten ins Rasterformat ist mit dem Befehl 'v.to.rast' geschehen, und bietet durch die automatische Farbzuzuweisung eine bessere Visualisierung.

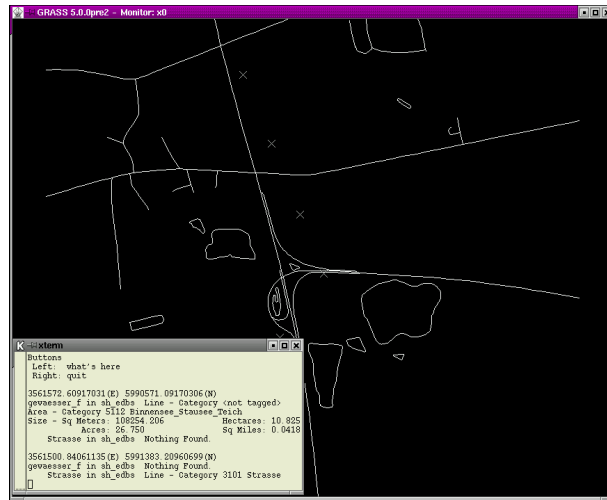


Abbildung 16.1.: GRASS Monitor mit EDBS Vektorkarte

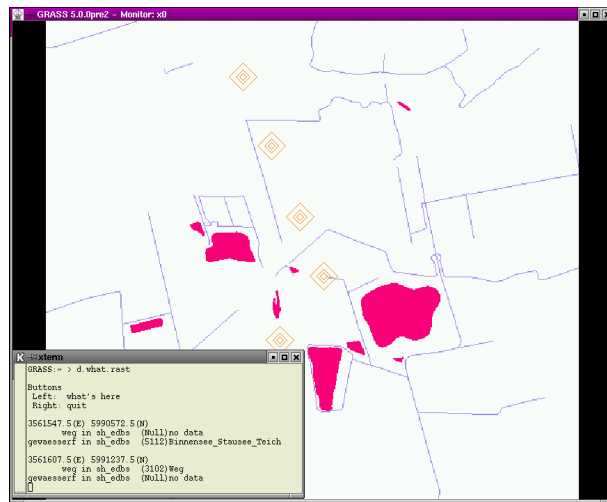


Abbildung 16.2.: GRASS Monitor mit EDBS Rasterkarte

# 17. Daten konvertieren

Um die erzeugten Daten, seien es Koordinaten, Höhenmodelle, Triangulationen, Orthophotos oder auch Bilddateien in möglichst vielen Programmen weiterverwenden zu können, oder z.B. die Arbeitsergebnisse einem Auftraggeber in dem von ihm gewünschten Format übergeben zu können, ist die Konvertierung von Daten in jedwede Form von zentraler Bedeutung in der Datenverarbeitung und besonders in den Geowissenschaften, in denen es für die verschiedensten Anwendungen oftmals eigene Datenformate und Koordinatensysteme gibt.

## 17.1. Höhenmodelle

Obwohl sich Höhenmodelle grundsätzlich aus einem 2-dimensionalen Raster zusammensetzen, dessen Rasterzellen als Information eine Höhenangabe tragen, gibt es doch eine Vielzahl von Formaten. Oft unterscheiden sich die Formate nur in den Informationen und ihrer Anordnung in dem Header - Teil der Dateien. Einige Formate haben auch eine zusätzliche Informationsdatei die zusammen mit den Rasterdaten das Höhenmodell ergibt, als Beispiel sei hier das TIFF - Dateiformat(.tif , .tiff) genannt, dem man ein TFW (Tiff World File) anfügen kann. Darüber hinaus gibt es sogenannte „Container Dateien, die viele Arten von Geodaten in sich tragen können und gleichzeitig Informationen über Orientierung, zugrundeliegende Koordinatensysteme, Alter u.s.w. speichern. Ein Beispiel hierfür ist das SDTS - Format des United States Geological Surveys (USGS).



## A. Ausblicke

Das Free GIS Tutorial soll genau wie die freie Software die, es behandelt kontinuierlich erweitert und überarbeitet werden und so quasi der Software folgen, die es beschreibt. Dazu ist es essentiell, daß genau wie bei freier Software jeder seinen Teil dazu beitragen kann. Beiträge, Themenvorschläge, Kommentare, Fehlermeldungen und Kritiken in jeder Form sind jederzeit willkommen und werden ihren Weg in folgende Versionen des Free GIS Tutorials finden. Einige Themen- und Aufgabenvorschläge, die bei der bisherigen Arbeit entstanden, aber noch nicht umgesetzt wurden sind zum Beispiel:

- Analytische Auswertung von Luftbildern. In den vorhandenen Luftbilder können die „Überlappungsstreifen“ dazu genutzt werden Höheninformationen aus ihnen abzuleiten und so das Höhenmodell zu erzeugen, das für eine Orthophotoherstellung nötig ist.
- Orthophotos aus Luftbildern erstellen. Diese Aufgabe ist zum Beispiel mit GRASS lösbar, erfordert aber ein detailliertes Höhenmodell des beflogenen Gebietes.
- In Orthophotos digitalisieren. Durch das Projizieren der Luftbilder in eine Ebene kann in ihnen wie in einer Karte gemessen werden. So können auf vielfältige Weise die verschiedensten Daten erzeugt werden.
- Verschneidungen der verschiedenen Datenarten (Vektordaten, Rasterdaten, Sachdaten) miteinander und anschließende Auswertung.
- Erweiterung des Glossars. Hier sind vor allem kurze, präzise Erklärungen der verwendeten Fachbegriffe gefragt.





## B. GNU Free Documentation License

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The purpose of this License is to make a manual, textbook, or other written document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

### B.1. Applicability and Definitions

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to

the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format,  $\LaTeX$  input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

## **B.2. Verbatim Copying**

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly

display copies.

### **B.3. Copying in Quantity**

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

### **B.4. Modifications**

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of

the principal authors of the Document (all of its principal authors, if it has less than five).

- State on the Title page the name of the publisher of the Modified Version, as the publisher.
- Preserve all the copyright notices of the Document.
- Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- Include an unaltered copy of this License.
- Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your

option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties – for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## **B.5. Combining Documents**

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

## **B.6. Collections of Documents**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document,

and follow this License in all other respects regarding verbatim copying of that document.

## **B.7. Aggregation With Independent Works**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an “aggregate”, and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document’s Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

## **B.8. Translation**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

## **B.9. Termination**

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## **B.10. Future Revisions of This License**

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License or any later version applies to it, you have the option of following the terms and conditions either of that specified version or

of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## **ADDENDUM: How to use this License for your documents**

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have no Invariant Sections, write “with no Invariant Sections” instead of saying which ones are invariant. If you have no Front-Cover Texts, write “no Front-Cover Texts” instead of “Front-Cover Texts being LIST”; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.





# Literaturverzeichnis

- [1] Bill, Ralf(1999): Grundlagen der Geo-Informationssysteme Band 1
- [2] Bill, Ralf(1999): Grundlagen der Geo-Informationssysteme Band 2
- [3] Neteler, Markus(2000): Geosynthesis GRASS-Handbuch
- [4] Greve, Georg C. F.: Was ist das GNU-Projekt?<sup>1</sup>

---

<sup>1</sup>[www.fsfeurope.org/documents/gnuproject.de.html](http://www.fsfeurope.org/documents/gnuproject.de.html)

# Index

- ALK, 153
- ATKIS, 153
- Bildschirmkoordinaten, 78
- CDF, 57
- CELL HAEDER, 73
- d.mon, 35
- d.sites, 37
- Datum, 28, 59
- DTED, 57
- EDBS, 153
- ESRI, 57
- g.region, 59
- g3DGMV, 56
- Gauss-Krüger, 29
- Georeferenzierung, 71
- geotiff, 61
- Hohenlinien, 63
- Hybride Graphik, 15
- Meridianstreifen, 59
- Monitor, 35
- nviz2.2, 123
- Passpunkte, 78
- PGM, 57
- ppmtompeg, 124
- Punktdaten, 37
- r.contur, 63
- r.in.png, 72
- r.in.ppm, 72
- r.in.tiff, 72
- r.support, 73
- Rasterdaten, 15
- Rasterdaten, anzeigen, 46
- RGB-Farbmodell, 72
- rpm, 128
- RTCM, 117
- s.in.ascii, 35
- s.surf.idw, 44
- s.surf.rst, 44
- sampeg, 125
- SDTS, 55
- shape, 57
- Shapefiles, 136
- Shell-Skript, 151
- Spline, 124
- Surfer GRD, 20
- Tischkoordinaten, 78
- Vektordaten, 15
- VTBUILDER, 56

# Glossar

**GPS** Das Akronym GPS steht für Global Positioning System. GPS ist ein System zur 3-dimensionalen Positionsbestimmung mittels räumlicher Bogenschläge von mindestens 4 Satelliten, bzw 2-dimensionalen Positionsbestimmung mit 3 Satelliten . Die bekanntesten Satelliten-Systeme sind unter anderen navstar-gps (USA) und glonass(GUS) sowie in Zukunft auch gallileo(EU).  
Heiko Kehlenbrink Nov.2001

ALK leer

ASCII leer

ATKIS leer

Attribut leer

Bildschirmkoordinaten leer

Bogenminute leer

Bogensekunde leer

BSD leer

BT Datei leer

CDF

Datum leer

Digital Elevation Model siehe Digitales Höhenmodell

Digital Line Graphs leer

DTED

EDBS leer

Ellipsoid leer

Geokodierung siehe Georeferenzierung

Georeferenzierung leer

[geotiff](#) [leer](#)

[GIFF](#) [leer](#)

[Gon](#) [leer](#)

[GPL](#) [leer](#)

[GTOPO30](#) [leer](#)

[GUI](#) [leer](#)

[Header](#) [leer](#)

[Hohenlinien](#) [leer](#)

[HTML](#) [leer](#)

[HTTP](#) [leer](#)

[Hybride Graphik](#) [leer](#)

[IP-Adresse](#) [leer](#)

[jpeg](#) [leer](#)

[Koordinatensystem](#) [leer](#)

[Koordinatensystem, Gauss-Krüger](#) [leer](#)

[Koordinatensystem, kartesisch](#) [leer](#)

[Lizenz GPL:](#) [leer](#)

[Lizenz MIT](#) [leer](#)

[Lizenz Public Domain](#) [leer](#)

[MacOSX](#) [leer](#)

[makefile](#) [leer](#)

[Meridianstreifen](#) [leer](#)

[NMEA](#) [leer](#)

[Passpunkte](#) [leer](#)

[Perl](#) [leer](#)

[PGM](#)

[PNG](#) [leer](#)

Polygon leer  
PPM leer  
PPM-Datei leer  
Proj.4 leer  
Projektion leer  
Projektion, Mercator leer  
Punktdaten leer  
Python leer  
Rasterdaten leer  
Rendering leer  
RGB Farbmodell leer  
rpm leer  
RTCM leer  
RTCM leer  
Sachdaten leer  
SDTS leer  
Shapefiles leer  
Shell-Skript leer  
Skriptsprache leer  
Spline leer  
Surfer GRD leer  
TIFF leer  
Tischkoordinaten leer  
Triangulation, Delaunay leer  
URL leer  
USGS ASCII-DEM leer  
USGS SDTS-DEM leer

UTM leer

Vektordaten leer

XPMI leer

**GNU/Linux:** Der Name des GNU-Projektes leitet sich von dem rekursiven Akronym "GNU's Not Unix", also "GNU ist nicht Unix" ab. Da Unix ursprünglich nicht nur eine Art von Systemen sondern auch ein Produkt bezeichnete, war dies dazu gedacht, klar zu machen, daß das GNU-Projekt ein System schaffen soll, welches zwar kompatibel zu aber nicht identisch mit Unix ist. Das GNU-System ist, wie es für Unix-Systeme üblich ist, modular angelegt, und heutzutage erfreut sich vor allem das GNU System mit Linux-Kernel - das sogenannte GNU/Linux-System - großer Beliebtheit und dient als Basis aller sogenannten Linux-Distributionen".  
Georg C. F. Greve[4]