

Digitale Höhendaten

1 Einleitung

Ein digitales Höhenmodell (engl. Digital Elevation Map, DEM) ist eine Darstellung der Geländeoberfläche eines Planeten, die keine Objekte wie Pflanzen oder Gebäude enthält. Im Gegensatz zu DSM (engl. Digital Surface Model) und DTM (engl. Digital Terrain Model), welche Informationen über Gebäude und Vegetation enthalten, beschreiben DEMs die topographischen Attribute und allgemeinen Informationen des Geländes. Es existieren mehrere Methoden zur Erstellung solcher Geländedaten. So können diese, durch Aufnahme von Satelliten oder Flugzeugen, sowie durch das Messen von Lichtimpulsen an der Oberfläche erstellt werden.

Digitale Höhendaten sind für unzählige Anwendungen von Bedeutung. Im wissenschaftlichen Bereich kann das Höhenmodell für topographiebezogene Analysen und Simulationen verwendet werden. Hinsichtlich der Umwelt können z.B. die Abbildung des Wasserflusses der Erde oder die Entstehung eventueller Probleme, wie die Entwicklung von Tornados in Erfahrung gebracht werden. Durch diese Informationen lassen sich Bewegungen im Erdreich und deren Einflüsse besser beurteilen [1]. In einer dreidimensionalen Sicht des Erdreichs lassen sich z.B. Aussagen über das Grundwasser treffen, in welchen potenziellen Bereichen es sich anreichert oder wann die Gefahr einer Kontaminierung besteht.

Auf dem heutigen Markt sind DEMs in immer größerer Vielfalt und Menge erhältlich. Im Februar 2000 hat die NASA an einem Projekt namens Shuttle Radar Topography Mission (SRTM) teilgenommen. Das Ziel der Mission war es, ein globales Datenset mit Landdaten der Erde zu erhalten. Mit Hilfe des Space Shuttles wurden ungefähr 80% der Erdoberfläche gescannt und als Datensatz zusammengestellt, der eine erstmalige Bodenauflösung von 30 Metern aufweist. Er ist somit einer der genauesten Datenaufzeichnungen der Erde [2].

1.1 Analyse der Aufgabenstellung

In dieser Arbeit sollen Höhendaten der Erde im zweidimensionalen Raum in eine bestehende Software importiert werden. Die vorhandenen Daten sind Bestandteil der Lösung für thermodynamische, partielle Differenzialgleichungen, für die die Daten als Input verwendet werden sollen.

Durch die umfangreiche Datensammlung eines Großteils der Erdoberfläche mit einer sehr guten Auflösung von 30 - 40 Metern, sind die Daten gut für das Projekt geeignet. Diese müssen für passende Bereiche als GeoTIFF-Datei heruntergeladen und in einem Array abgespeichert werden. Zum Auslesen der Höhendaten aus der Datei und zum Schreiben in das Array, dient die Open-Source-Bibliothek GDAL, die für jedes Betriebssystem zur Verfügung steht. Beim Schreiben der Daten sind mehrere Aspekte zu beachten, z.B. die Überlappung und die unterschiedliche Auflösung der Bilder.

1.2 Auswahl und Struktur der Daten

Das Datenset der NASA-Mission steht kostenlos zum Download zur Verfügung [3]. Zum Herunterladen der benötigten Daten wird ein Account benötigt, der frei zugänglich ist. Es besteht die Möglichkeit den gewünschten Bereich auf einer Weltkarte entweder frei durch das Setzen von Koordinaten oder durch vordefinierte Bereiche zu markieren.

Die Karte ist in Längen- und Breitengrade unterteilt und bildet somit eine Rastergrafik. Jede Kachel im Raster entspricht einem Bild. Von 49° Süd nach 49° Nord haben die Pixel eine Auflösung von ca. 30 Metern und jede Kachel des Rasters hat eine Größe von einem Grad, was $3601 * 3601$ Pixeln entspricht. Von jeweils 50° bis 60° sinkt die Auflösung auf 40 Meter pro Pixel und eine Kachel hat bei gleicher Breite eine Größe von $1801 * 3601$ Pixel.

Durch die gleiche Breite der niedrigen Auflösung sind halb so viele Pixel vorhanden. Die fehlenden, mittleren Pixel können mit Bildung eines Mittelwerts der Nachbarpixel berechnet und hinzugefügt werden.

Nach der Auswahl des Datensets SRTM stehen die Kacheln, mit denen sich der markierte Bereich überschneidet, zum Download bereit. Beim Herunterladen der Daten stehen drei verschiedene Datentypen zur Verfügung. Der Standarddatentyp

zum Speichern der Rasterdateien ist als GeoTIFF-Datei verfügbar, die zum Unterschied zur normalen TIFF-Datei zusätzliche Informationen wie z.B. Koordinaten enthalten kann [2].

Um das ganze Gebiet abzudecken, müssen alle Kacheln in einem Array gespeichert werden. Dabei ist zu beachten, dass die Grenzen des ausgewählten Bereichs nicht gleich den Grenzen der Gitterpunkte sind. Die Höhendaten einer Reihe liegen jeweils auf den Eckenpunkten und die letzten Pixel eines Bildes entsprechen den ersten Pixeln des nächsten Bildes. Um die gewünschte Region zu bestimmen, müssen diese Bereiche jeweils nur einmal kopiert werden. Außerdem sind die Kacheln zeilenweise und von oben nach unten nach West – Ost – Richtung ausgerichtet, worauf beim Aufrufen und Speichern der Daten geachtet werden muss.

2 Erstellung des Arrays mit Höhendaten

Beim Import der Daten sollen die im letzten Kapitel angesprochenen Probleme aufgefasst und gelöst werden. Dabei ist darauf zu achten, dass der Zugriff auf die Höhendaten möglichst schnell und ohne Overhead geschehen soll. Da die Daten mehrmals zur Berechnung aufgerufen werden, ist die Performance von besonderer Bedeutung, um den Verlust an Geschwindigkeit zu minimieren.

2.1.1 Aufbau des Rasters

Abhängig vom ausgewählten Gebiet stellt USGS ein oder mehrere Bilder zur Verfügung. Jedes Bild stellt eine Kachel dar und entspricht sowohl bei Längen- bzw. Breitengraden der Größe von einem Grad. Ein Mittelpunkt bei der Bearbeitung der Bilder ist der Name, der von USGS generiert wird. Die Dateinamen haben die Form „latXX_lonYYY_1arc_v3.tif“. Bei einem führendem „n“ befindet sich die Koordinate am Nordpol und bei einem „s“ am Südpol. Bei „lon“=“w“ entspricht sie dem Westpol und bei „lon“=“e“ dem Ostpol.

Jede Koordinate ist in einem Bild mit dem Namen dieser Form enthalten, wobei XX zu den abgerundeten Längengraden und YYY zu der abgerundeten Breitengraden dieser Koordinate gehört. Zum Beispiel die Koordinate mit Breitengrad 49°25'46" N und mit Längengrad 125°36'65" E ist in Datei mit dem Namen „n49_e125_1arc_v3.tif“ enthalten. Dieses gilt aber für Nord-Ost -Pol. Für die Koordinate mit der gleichen Breiten- und Längengraden, wird im Süd-West-Pol der Dateiname „s50_e126_1arc_v3.tif“ verwendet. Wenn der Bereich sich am Rand im Netz befindet, werden zusätzliche alle umliegenden Bilder generiert. Z.B. für den Bereich mit der oberen linken Koordinate (10°00'00" S, 51°00'00"W) und mit der unteren rechten Koordinate (11°00'00" S, 50°00'00"W) werden insgesamt 9 Dateien zur Verfügung gestellt, da z.B. die Koordinate (10°00'00" S, 51°00'00"W) in folgenden 4 Dateien enthalten ist:

s10_w050_1arc_v3.tif

s10_w051_1arc_v3.tif

s11_w050_1arc_v3.tif

s11_w051_1arc_v3.tif

2.1.2 Einlesen der Bilder

Die heruntergeladenen GeoTiff-Dateien müssen automatisch eingelesen und in ein Array gespeichert werden. Mit Hilfe der Open Source Bibliothek „GDAL“ ist es möglich die Struktur des Formates zu lesen und die enthaltenen Daten direkt in ein Array zu speichern, wie im folgenden Codeabschnitt 1 zu sehen ist.

```
1 void ReadFile::getData(Options& opt)
2 {
3     GDALAllRegister();
4
5     dataset = (GDALDataset*)GDALOpen(opt.path.c_str(), GA_ReadOnly);
6     if(dataset == NULL)
7     {
8         throw "Can not load dataset";
9         return;
10    }
11
12    GDALRasterBand* band;
13    int nBlockXSize, nBlockYSize;
14
15    band = dataset->GetRasterBand(1);
16    band->GetBlockSize(&nBlockXSize, &nBlockYSize);
17
18    int nXSize = band->GetXSize();
19    int nYSize = band->GetYSize();
20
21    if(opt.lat < 50 && opt.lat > -51)
22    {
23        band->RasterIO(GF_Read, opt.offsetXTopCorner, opt.offsetYTopCorner,
24                      opt.imgSizeX, opt.imgSizeY,
25                      opt.rasterData + opt.numberOfElements,
26                      opt.imgSizeX, opt.imgSizeY, GDT_Float32,
27                      0, 0);
28    } else
29    {
30        linearInterpolation(band, opt);
31    }
32
33
34    GDALClose(dataset);
35 }
```

Abbildung 1 Höhendaten einlesen

Die Funktion *GDALOpen()* in Zeile 5 bekommt als Übergabeparameter den Dateipfad und ein Flag zum Lesen der Datei. Das daraus resultierende Datenset umfasst alle Informationen über das GeoTiff Format. Sowohl Metainformationen, als auch ein Rasterband, Zeile 15, welches in diesem Fall die Höhendaten enthält, können aus diesem Set ausgelesen werden.

Bei allen Bildern, die einen Längengrad von größer als 50 oder kleiner -50 haben, sinkt die Auflösung und würde zur Verfälschung des Gesamtarrays führen.

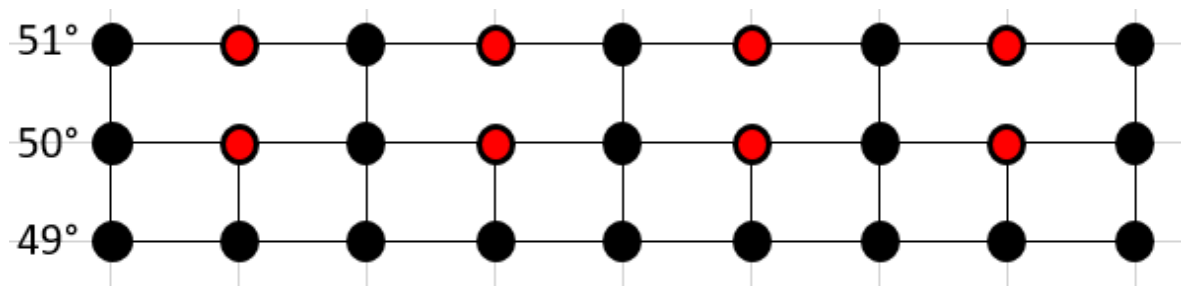


Abbildung 2 Fallunterscheidung

Die einzelnen Pixel liegen jeweils auf den Eckpunkten der jeweiligen Zeile, wie in Abb. 2 zu sehen ist. Bei einer niedrigeren Auflösung, ab 50 oder -50 Grad, halbiert sich die Anzahl der Pixel. Ohne die Breite mit den fehlenden Pixeln zu ergänzen, würde dies beim Zugriff auf die jeweilige Koordinate zu einem falschen Wert führen. Um dies zu verhindern, wird für diese Bilder in der Programmierung eine Fallunterscheidung in Zeile 21 durchgeführt. Bei Bildern mit hoher Auflösung werden alle Pixel nacheinander im Array gespeichert. Bei Daten mit niedriger Auflösung wird mit der linearen Interpolation der Mittelwert über die Nachbarpixel berechnet und nach jedem Pixel hinzugefügt, siehe Abb. 2. Somit wird durch die gleiche Anzahl von Pixeln jedes Bildes ein Sprung im Array vermieden.

GDAL stellt zum Speichern der Daten die Funktion *RasterIO()* zur Verfügung. Der erste Parameter legt dabei fest, ob Daten im Bild gelesen oder geschrieben werden sollen. Die vier darauffolgenden Integerwerte setzen sowohl den oberen linken Pixel zum Start fest, als auch die Anzahl der zu lesenden Pixel. Mit der Übergabe eines Voidpointers können die gelesenen Werte, in diesem Fall in ein Floatarray, gespeichert werden. Um auf unnötiges Kopieren der Daten zu verzichten wird dabei immer das gleiche, vorher erstellte Array verwendet und die Elemente mit einer Verschiebung des Pointers mit der Gesamtanzahl aller Elemente am Ende hinzugefügt. Die nächsten zwei Parameter legen die Anzahl der bei jedem Vorgang zu speichernden Pixel fest. Im Fall der linearen Interpolation erfolgt der Einsatz eines temporären Buffers als Array, der den Durchschnitt der Werte berechnet und im Anschluss diese zweimal am Ende des Gesamtarrays hinzufügt.

2.1.3 Erstellung des Gesamtarrays

Zum Einlesen und Erstellen eines gesamten Bereichs über mehrere Bilder findet die Klasse *ElevationData2D* Anwendung.

```
1 ElevationData::ElevationData(const Coordinate& c1, const Coordinate& c2) : c1(c1), c2(c2)
2 {
3     Vector<String> files = filesSet(c1, c2);
4     for (uint i = 0; i < files.size(); ++i)
5         fileNames.push_back(files[i]);
6
7     rasterData = new float[rasterDataSize()];
8     createElevationArray();
9 }
```

Abbildung 3 *ElevationData2D*

Der Konstruktor der Klasse erwartet einen String und zwei Koordinaten als Parameter. Diese legen den Bereich, der eingelesen werden soll, mit der oberen linken und der unteren rechten Ecke fest. Der String enthält den Pfad zu den Bildern und in den Zeilen 3 – 5 werden automatisch die benötigten GeoTiff-Dateien für den Aufbau des gewählten Bereichs ausgewählt. Die Erstellung und Berechnung der Größe des Arrays erfolgt in Zeile 7 und mit dem Aufruf der Methode *createElevationArray()* wird dieses mit den passenden Daten befüllt. Somit ist nur eine einfache Erstellung der Klasse nötig, um das Array automatisch zu generieren. Mit dem darin überschriebenen Klammeroperator kann mit zwei Integerwerten einfach die gewünschte Position zurückgegeben werden.

Die Anzahl der Bilder stellt einen zweidimensionalen Bereich dar und damit der Zugriff ohne Overhead stattfinden kann, wird dieser zeilenweise eingelesen und das Array ist somit ein exaktes Abbild. Daher sind zwei Werte ohne weitere Berechnung für den Zugriff ausreichend.

Zum Erstellen des gesamten Arrays mit der *createElevationArray()*-Methode wird die Methode *ReadFile()* eingesetzt, um die Anzahl der Bilder in einer Reihe, zeilenweise über den gesamten Bereich aufzurufen. Für jeden Aufruf werden die nötigen Parameter, die linke obere Koordinate, die Anzahl der zu lesenden und zu schreibenden Pixel und das Floatarray, welches im Konstruktor erstellt wurde, weitergegeben. Da die letzten Pixel des ersten Bildes und die ersten Pixel des nächsten Bildes in Reihe und Spalte übereinstimmen, sind die Parameter so zu setzen, dass die Pixel jeweils nur einmal in das Array gespeichert werden. Die im

vorherigen Kapitel angesprochene Fallunterscheidung wird mit der Übergabe von Längen- und Breitengraden beachtet.

Mehreres Öffnen der einzelnen Bilder führt dazu, dass die Erstellung des Arrays zwar länger dauert, aber die Gewinnung an Performance beim Zugriff auf die Höhendaten überwiegt, da das Array insgesamt nur einmal erstellt werden muss. Allerdings muss der Zugriff öfter und ohne Overhead geschehen.

3 Beispiel und Ergebnisse

Für ein Beispiel wurden die Daten für den Großteil von Deutschland heruntergeladen und stehen zum Test bereit. In diesem Beispiel wurde der Bereich vom Königsstuhl und der Altstadt in Heidelberg gewählt. Die Erstellung des Arrays sieht folgendermaßen aus.

```
String path = "";  
ElevationData2D::Coordinate c1(49.43,8.66), c2(49.40,8.73);  
ElevationData2D::ElevationData eData(path, c1, c2);  
  
float elevationData = eData(10, 20);
```

Abbildung 4 Generierung des Testarrays

Zuerst werden die Koordinaten initialisiert, um den gewünschten Bereich aufzuspannen. Mit der Erstellung der ElevationData-Klasse und der Übergabe der eben erstellten Koordinaten wird das Array mit den Höhendaten erstellt und befüllt. Die Rückgabe des Höhenwerts an Position [10, 20] erfolgt durch den Aufruf des Klammeroperators. Mit einem Matlabskript können die Höhendaten optisch dargestellt werden.

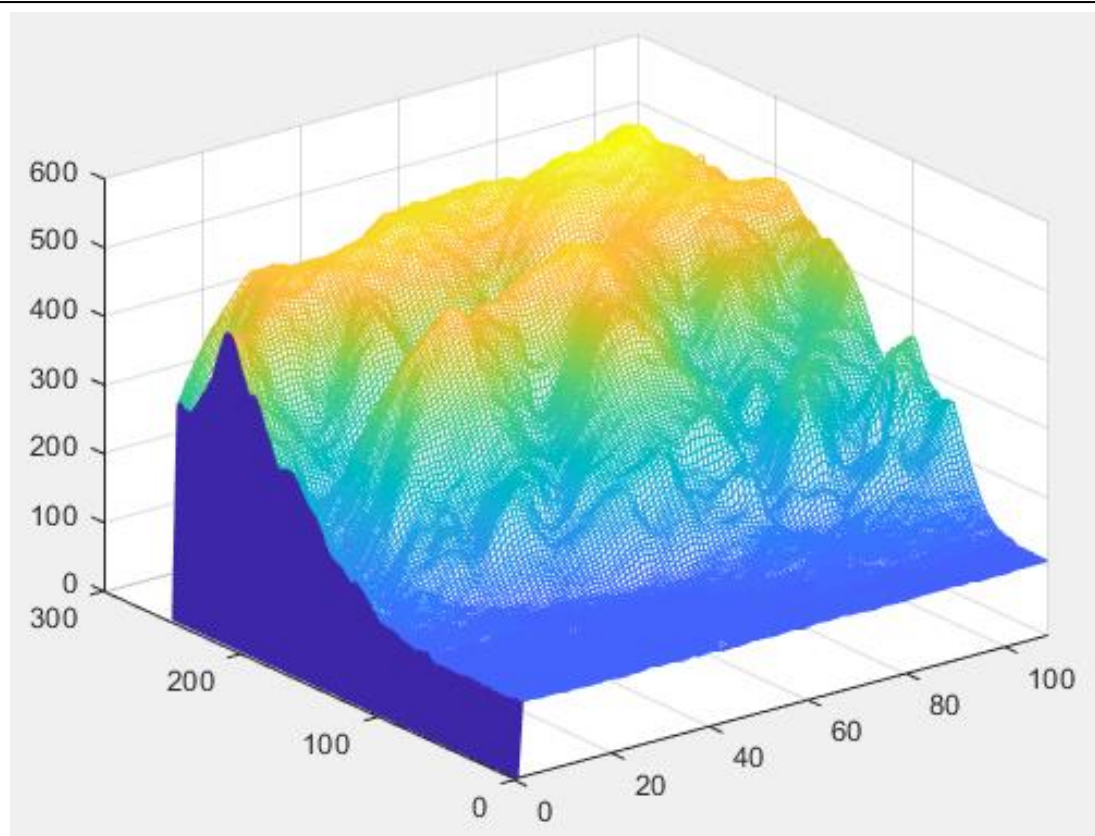


Abbildung 5 Grafik der Höhendaten

4 Literaturverzeichnis

- [1] UiZ, „Use of Digital Elevation Model,“ [Online]. Available: <http://uizentrum.de/en/use-of-digital-elevation-model-in-gis-dem-gis/>.
- [2] „Digital Elevation Models & their Applications,“ [Online]. Available: <http://www.geol-amu.org/notes/rs12-4-1.htm>.
- [3] „EarthExplorer,“ USGS, [Online]. Available: <https://earthexplorer.usgs.gov/>.