

# Umgang mit großen bzw. vielen Raster-Daten

Häufig ist es erforderlich, große und/oder viele Rasterdaten in QGIS zu laden oder zu verarbeiten. Zum Beispiel, wenn wir Reliefanalysen an einem Höhenmodell durchführen möchten, von Orthofotos abdigitalisieren wollen oder schlicht eine lokale Basishintergrundkarte benötigen. Die räumliche Auflösung der verfügbaren Daten wird immer besser und die Datenmengen entsprechend größer (denken wir z.B. an die LiDAR Höhenmodelle, welche uns seit wenigen Jahren zur Verfügung stehen oder die vielerorts erhältlichen Orthofotos mit 1x1m Auflösung!), Wir müssen dann mit den begrenzten Ressourcen unseren PCs vernünftig haushalten. Hier einige Methoden und Tipps.

## Wiederholung: Rasterdaten

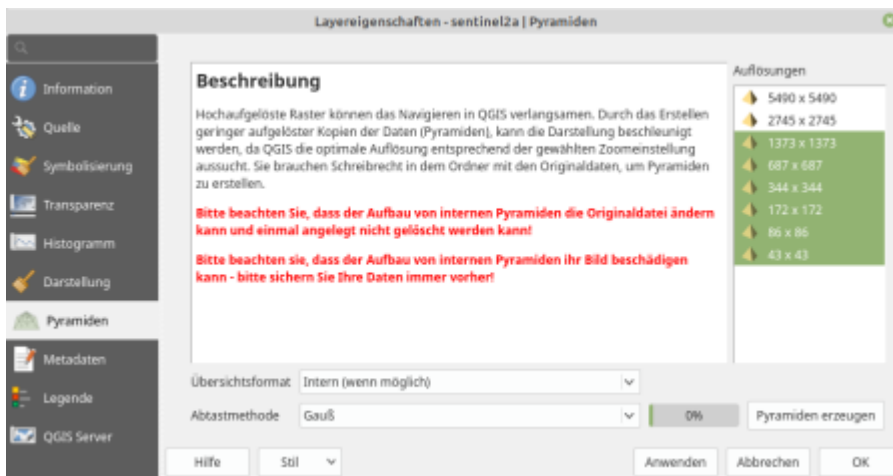
Rasterdaten sind aufgrund ihres Speichereigenschaften häufig groß und rechenintensiv. Die Verarbeitung von großen oder vielen Rasterdaten kann daher sehr langwierig sein oder an mangelnden Hardware-Ressourcen scheitern. Es gibt aber **Funktionen und Methoden**, die einem Helfen, auch mit **wenig Ressourcen und Zeit, große Datenmengen zu verarbeiten** oder diese zumindest **häppchenweise zu servieren**.

## Das richtige Format

Es gibt viele verschiedene Raster-Speicherformate<sup>1)</sup>. Die einen resultieren in kleinen Dateien, die anderen in großen. Doch worin unterscheiden sie sich?

Ein Rasterbild ist eine Matrix in welchem **jede Zelle einen Wert** (Farbwert, Höhe, Lärmpegel, CO<sup>2</sup>-Gehalt) besitzt und **eindeutig zur Nachbarzelle abgegrenzt** (diskret) ist. Speichert man eine Matrix genau so, also *Roh bzw. RAW* (📁 [GeoTIFF](#), ASC, XYZ etc.), so wird es in einer relativ großen Datei resultieren - eben unkomprimiert. Wählt man hingegen ein komprimierendes Speicherformat wie 📁 [JP2000](#), 📁 [PNG](#) etc., welches nicht jede einzelne Zelle diskret abspeichert, sondern nur „einige“ während die dazwischen liegenden Zelle *interpoliert* werden, so resultiert das in einer kleinen und vergleichsmäßig schnellen Datei. **ABER** die kleine Datei ist zwar *optisch* einwandfrei, doch tatsächlich sind die einzelnen Zellen nicht mehr diskret von einander getrennt, **sondern miteinander vermischt**. Das ist Okay bei Orthofotos, digitalen topographischen Karten oder ggf. bei Satellitenbildern, da diese uns oft nur zur Orientierung dienen, für **wissenschaftliche Aufgaben und Analysen sind sie jedoch unbedingt zu meiden!** Bei Relief-Analysen verwenden wir Digitale Gelände Modelle (DGM, DEM) welche unbedingt in einem rohen Format gespeichert sein müssen, sonst werden unsere Analysen und Ableitungen (z.B. Höhenlinien, Exposition, Lokale Maxima und Minima etc.) falsche Ergebnisse liefern!

## Ein einzelnes großes Raster-Bild schnell darstellen (rendern)



**Abb. X:** Pyramiden in den Eigenschaften des Rasterlayers erzeugen

**Kleinere Bildformate sind schneller.** So kann es bereits eine Lösung sein, ein großes und schwerfälliges Raster **in ein anderes Format zu exportieren** (z.B. das JP2<sup>2)</sup> ist ein schnelles Format). Das sollten wir jedoch nicht, wenn es sich um Roh-Daten handelt, welche uns zu Analysen und Ableitungen dienen sollen (siehe oben). Ein DGM zum Beispiel muss im rohen Format bleiben (häufig sind das TIFF, ASC oder XYZ).

**Um Rohdaten schneller darstellen zu können**, gibt es die Möglichkeit, sogenannte **Pyramiden** zu erzeugen. Dabei handelt es sich um Duplikate des ursprünglichen Rasters in **geringeren Auflösungen**, die je nach Maßstab so bereit gestellt werden, dass sie uns optisch gerade so genügen. Diese Duplikate können je nach Speicherformat mit in die Originaldatei geschrieben oder als separate, externe Datei erzeugt werden.



Aus einem bereits **komprimierten Format** wie PNG oder JPEG/JPEG2000 **können keine Pyramiden** erzeugt werden!

## Große Gebiete mit hoch aufgelösten Bildern wiedergeben

Hochaufgelöste Orthofots für ganze Bundesländer sind auch in einem kleinen Speicherformat wie JP2 immer noch riesig und rechenintensiv. Nehmen wir folgendes Beispiel: Der Regierungsbezirks Aachen wird mit 136 hochaufgelösten Orthofots abgedeckt, was in insgesamt 10GB Speicherbedarf resultiert [Download Aachen DOP \(10GB!!!\)](#). Fügen wir alle zusammen (mit merge) so erhalten wir ein einzelnes 10GB-großes Bild, dessen Erzeugung mehr als 14GB RAM benötigt (weshalb meine Hardware nicht in der Lage war, es zu generieren). Alternativ können wir aber auch ein sogenanntes **virtuelles Raster** erzeugen, welches die einzelnen Szenen nicht physisch zusammenfügt, sondern einen Katalog mit Verlinkungen zu jedem einzelnen Bild erzeugt. Das Erzeugen eines solchen Virtuellen Raster dauert zwar auch einige Minuten, die dafür benötigte Rechenleistung ist aber überschaubar. Es resultiert eine winzige-Indexdatei von 0,5 MB.

Laden wir nun das virtuelle Raster, so werden alle verlinkten Kacheln geladen. Auch dieser Vorgang dauert lange, denn es werden immerhin 40.800.000.000 Bildpunkte geladen. Um auch dies zu beschleunigen, könnten wir wieder Pyramiden für das Virtuelle Raster erzeugen, da dieses jedoch sich

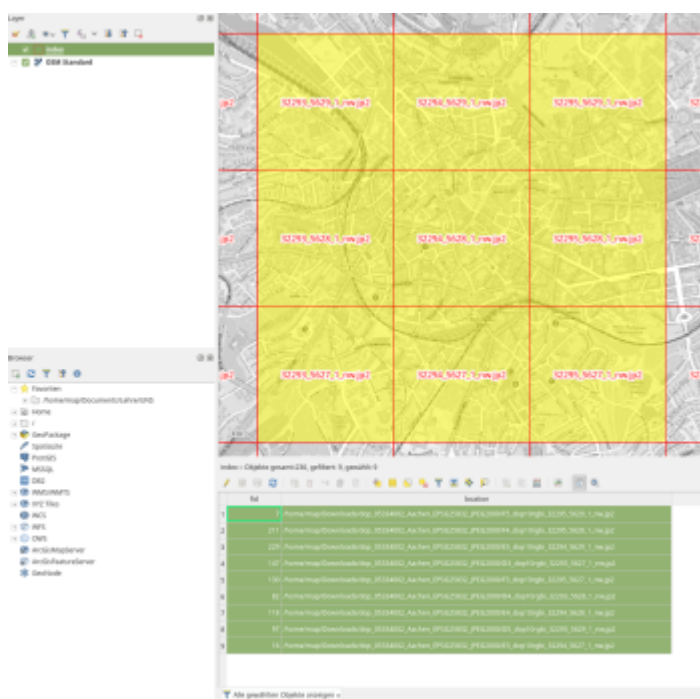
auf JP2000-Dateien bezieht, ist das leider nicht möglich. Wir müssen alle Dateien in z.B. TIF umwandeln und damit weiterarbeiten....

## Raster-Indexdatei erzeugen



**Abb. X:** Die Funktion „Kachelindex erzeugen“ aus der GDAL bibliothek

In der Regel ist es gar nicht nötig, alle Kacheln zu laden oder zu einem Bild zusammen zu fügen. Meistens genügt es uns, **die Bilder zu laden, welche für die Ausdehnung unseren Projekts von Relevanz sind**. Hierzu müsste man aber wissen, welches Bild (Dateiname) an welcher Stelle sich befindet (und das geht aus dem Dateinamen oft nicht hervor!). Hier hilft ein **Kachel-Index**! Diesen erzeugen wir mit dem Befehl `Kachel - Index erzeugen` oder über das Menü `Raster → Verschiedenes → Kachel Index....` Es wird eine Shapedatei erstellt, in welcher für jede Bildkachel ein Polygon (Rechteck) erzeugt wird. In den Attributen der einzelnen Kachel-Polygone finden wir dann den Pfad und Dateinamen zur ursprünglichen Raster-Kachel (siehe Abb. X).



**Abb. X:** Das Kachelindex-Shape nach ein paar gestalterischen Handgriffen

1)

## GDAL Raster Format List

2)

### JPEG2000

From:

<https://lernplattform.map-site.de/> - **Lernplattform für OpenSource GIS**

Permanent link:

[https://lernplattform.map-site.de/doku.php/qgis/advanced/c\\_fortgeschrittene\\_rasterverarbeitung/lektion-5](https://lernplattform.map-site.de/doku.php/qgis/advanced/c_fortgeschrittene_rasterverarbeitung/lektion-5)

Last update: **2022/09/10 00:07**

