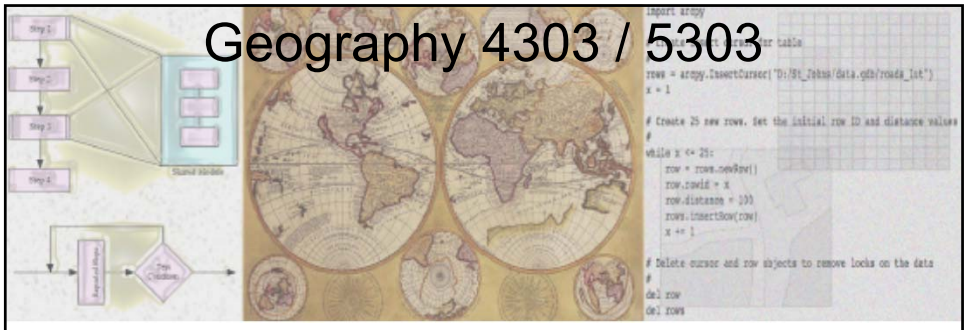


Geography 4303 / 5303

GIS PROGRAMMING FOR SPATIAL ANALYSIS

Class 07: Working with Raster Data in Python

Some Updates

- Project group selection next Monday!
- Lab learning curve
- Schedule
- Exercise class 06 transect sampling
- Imagine: Create a systematic point sample along a polyline?

Last Lecture / Last Week

- On the example of spatial sampling: How is everything coming together...?
  - **use of complex data structures,**
  - **branching,**
  - **control flow,**
  - **logic** elements and operations
  - **Iterations, Looping**
- **Error search and Debugging**
- Modularity: Working with modules and functions
- **GP**: How to work with **vector data and Geometry**

Today ‘s Outline

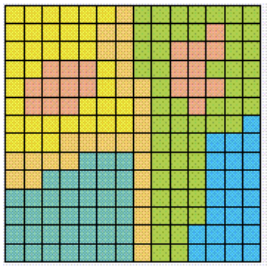
- We start with **raster data** and **raster operations**
- We will look into **raster data structure** and characteristics
- First steps in accessing and analyzing **raster datasets** for geoprocessing
- Understand how to develop **user-specified raster operators**
- numpy, scipy and gdal

Learning Objectives

- Understanding how to access **raster datasets**
- **Implementing raster operations** using data structure and properties
- Understanding programming logic in raster analysis
- Moving windows, focal operations etc.
- Design your own solution in raster analysis

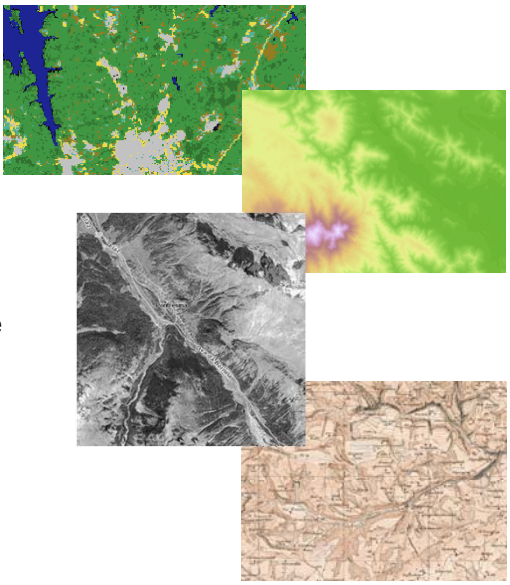
Raster data

- **Matrix of cells** (pixels) organized into **rows and columns** (grid)
- Each cell contains a **value** representing information (e.g., spectral information)
- Digital aerial photographs, satellite images, digital pictures, or even scanned maps



What can raster data tell us?

- **Thematic information** (discrete data):
  Land use, soil data
- **Continuous data**:
  Data regarding phenomena of changing degree over space on a continuous scale
- **Images / pictures**
  Pixel values represent reflectance of real world phenomena captured by the **sensor**



Raster .mxd

Raster Analysis in arcpy

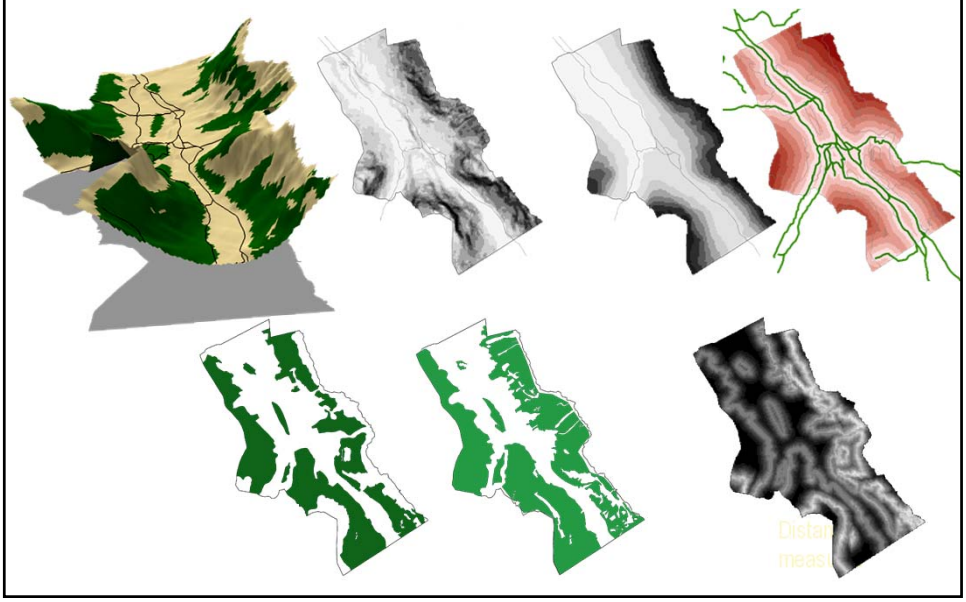
- **Native** raster analysis **methods** available (**SA**, **Management**, **3D toolboxes)**
- Map algebra in a **workflow** (**ModelBuilder)** or via Scripting
- **Raster objects**
- For many cases we don't "re-invent the wheel":
  - Math and Conditions
  - Distance / Surface Analysis / Hydrology tools
  - Extraction
  - Filtering/Generalization
  - Reclassify
  - Statistical Analysis (Global, zonal, focal) and Geostatistics

Example curvature fct

Constraints of arcpy raster analysis

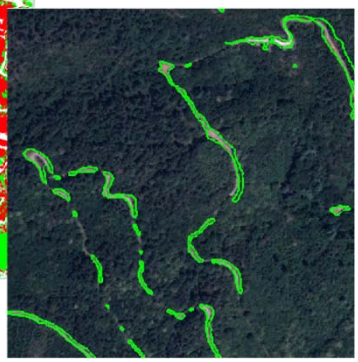
- Limited information **how** operators / tools work
- Difficult **modification** or **extension** of existing tools (only parameterization)
- **Limited access** to the raster data structure for developing customized operations (e.g., filtering)
- Specific needs (remember our project topics):
  - Spatial statistics
  - classification procedures
  - feature extraction
  - recognition/ detection in RS (AI)
  - **Map algebra, uncertainty** analysis
  - Some quick examples

Raster overlay and uncertainty analysis



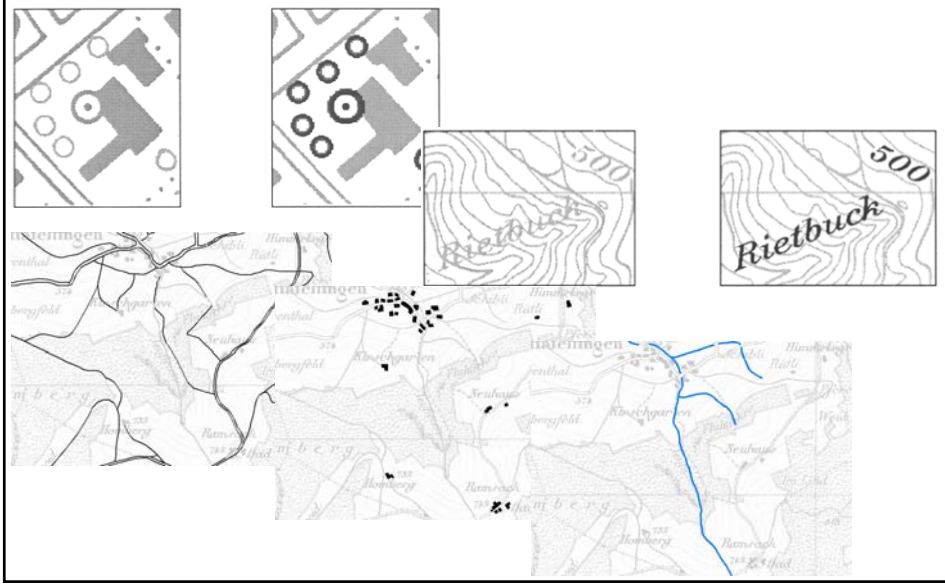
Information extraction and detection

Building extraction (Shorter & Kasparis, 2009)



Contour models (El Ghoul, A. 2011)

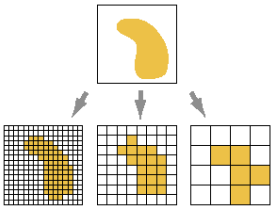
Feature Extraction and Pattern Recognition



How to start working with raster data?

Explore data characteristics

- Limitations in detection due to **cell size**
- Size of datasets depending on **resolution/cell size**
- Data type
- **Shape "approximations"** (RS data acquisition)
- **Whole** pixel = homogeneous entity (discrete vs. continuous)



Explore data characteristics

- **Format**: File type (with specific properties)
- **Number of bands**: # spatially coincident layers (min 1)
- **Data type**: Pixel type - int or float
- **Data depth**: Bit depth - range of possible values in a band
  8: 2**8=256 (0 to 255); **16**: 2**16=65536 (0 to 65535)
- **Statistics**
- **Extents**: Left, right, top, bottom
- **Projection**
- **Size**: rows and columns (uncompressed)

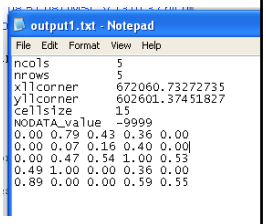
Example raster properties ArcCat.

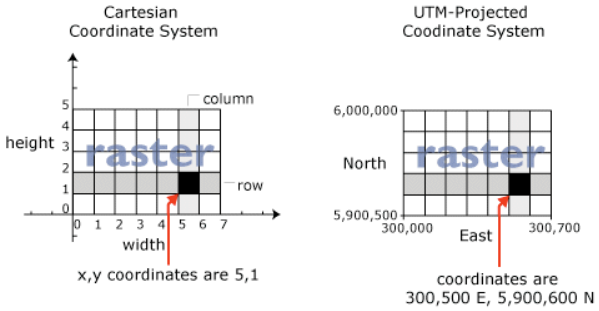
Bit depth

Bit depth	Range of values that each cell can contain
1 bit	0 to 1
2 bit	0 to 3
4 bit	0 to 15
Unsigned 8 bit	0 to 255
Signed 8 bit	-128 to 127
Unsigned 16 bit	0 to 65535
Signed 16 bit	-32768 to 32767
Unsigned 32 bit	0 to 4294967295
Signed 32 bit	-2147483648 to 2147483647
Floating-point 32 bit	-3.402823466e+38 to 3.402823466e+38

Georeferencing in raster data

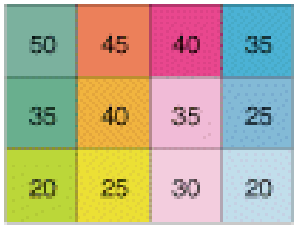
- **Cartesian vs projected** coordinate system representation: row || to x, col || to y
- Header of image file (grids, img, GeoTIFF)
- World (ASCII) files (.tfw)
- Origin of an image is the upper left corner (ul)
- Row values increase downward





Array origin

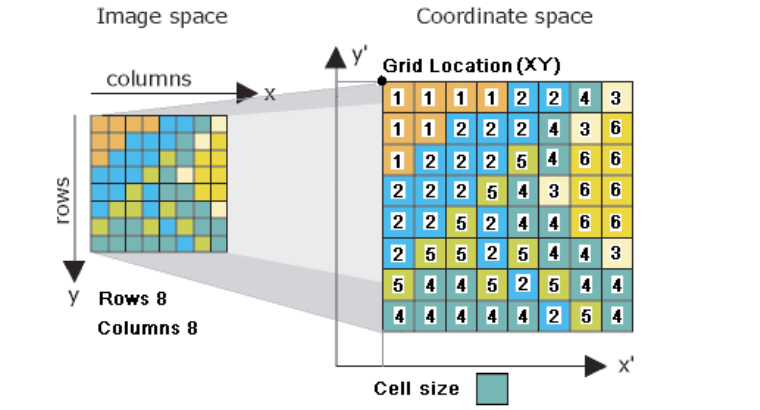
Four Resolutions



- **Spatial**: pixel resolution
- **Temporal**: time till recapture
- **Spectral**: width of bands and how many bands or channels; which portions of the spectrum
- **Radiometric**: sensitivity/precision (pixel depth)
- How to make a decision…

Show RS data - res

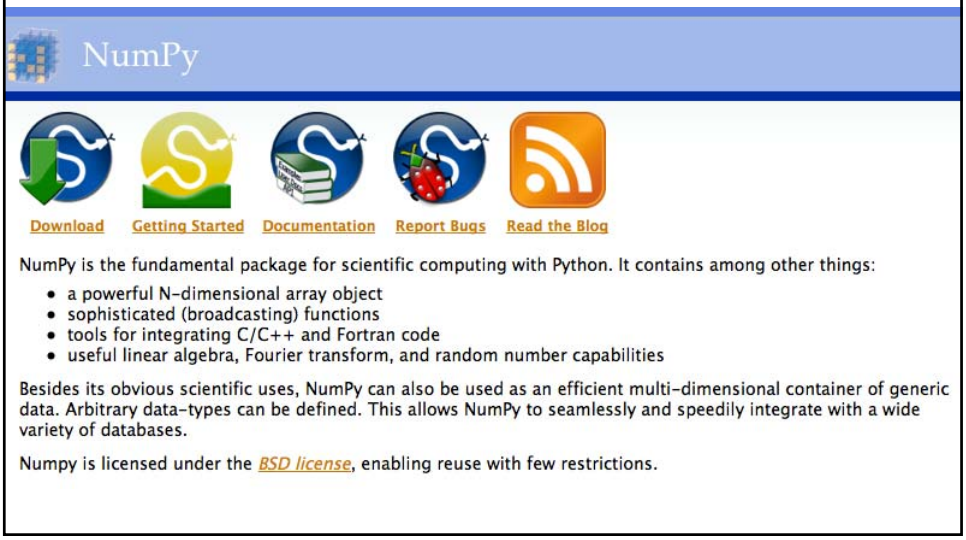
Addressing the locations



A good way to access raster information? Time for numpy…

myArray[rows,cols]

Scientific Computing with numpy (& scipy)



What Numpy Brings In

- Multidimensional array language for Python
- The numpy module defines the **array object** (**`multiarray`** object)
- … and a **set of universal functions** for manipulating and converting these objects
- Translating (any kind of) **raster data** into Numpy arrays and back (also arc grids)
- See some examples

Class07_numpy create array … class07_rstobj for DEM

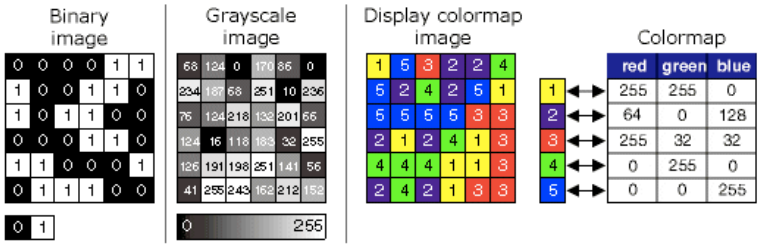
About Numpy Arrays

- Python-style **slicing/indexing:** access individual locations and **subregions**
- **Collections** of large **amounts of numbers** – all of the **same kind**
- Array objects must be **full** (**no empty cells**)
- Array **size is immutable**; Values **can change**
- Mathematical operations on arrays return **new arrays**
- ***Element-wise*** *performance: myArray[rows,cols]*

Class07_numpy – op. examples; univ. fct vs. pix-wise analysis

Single-band raster

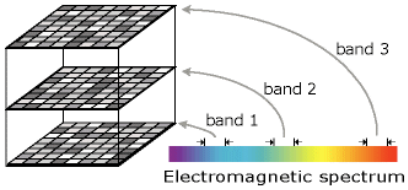
- DEM (each cell with only one value)
- Areal photographs (e.g., 8 bit)
- **Binary** images (parcel maps, query results)
- **Color map** images

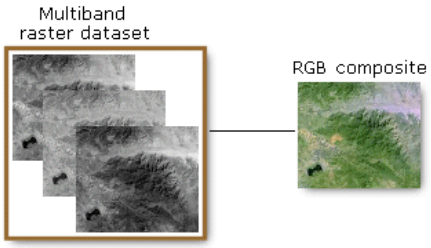


Show DEM ex.

Multiple bands

- N single 2D matrices of cell values
- **Spatially coincident matrices** of cell values (same area)
- Several values at each **cell location**
- **Band** = Segment of the **electromagnetic** spectrum
- **Satellite** imagery





Show NDVI ex.

Critical in working with raster data

- Map Algebra (e.g., focal functions)
- Segmentation: Object identification
- Classification
- Detection: Edges, contours, transitions between objects

gdal: https://pypi.python.org/pypi/GDAL/
scipy: https://scipy.org/

Show mean filter

Summary

- Raster datasets have a very simple structure and yet are powerful in representing, analyzing and modeling
- You have to understand some basic properties to work effectively with them
- Experience in programming on raster datasets gives you a basic skill in many professional/scientific fields