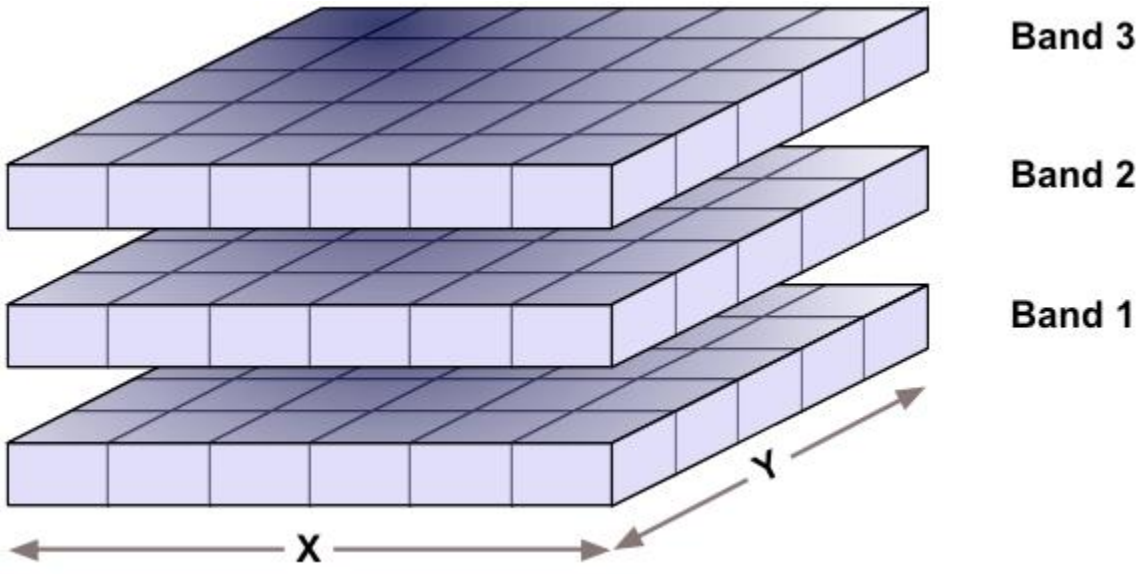
Podstawy programowania

Raster geoprocesing
w GDAL i NumPy

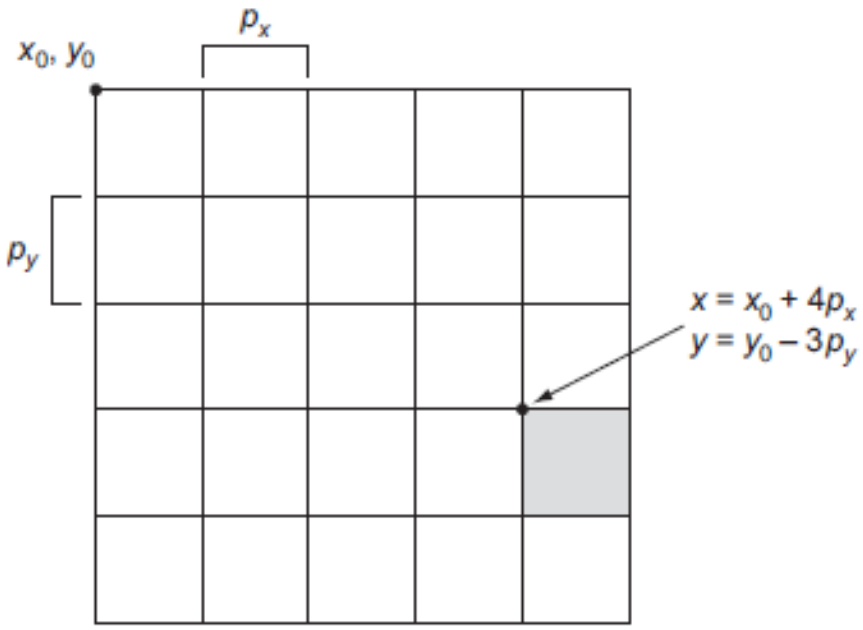
Cechy danych rastrowych

- Ilość kanałów w rastrze
- Typ komórki rastra (pixela) w kanale:
  - 1 bit Boolean - GDT_Byte
  - Unsigned integer 16, 32 bits – GDT_UInt16, GDT_UInt32
  - Signed integers 16, 32 bits – GDT_Int16, GDT_Int32
  - float: 32, 64 bits – GDT_Float32, GDT_Float64
- Układ współrzędnych rastra
- Wysokość i szerokość komórki rastra (pixela) np. 30x30m

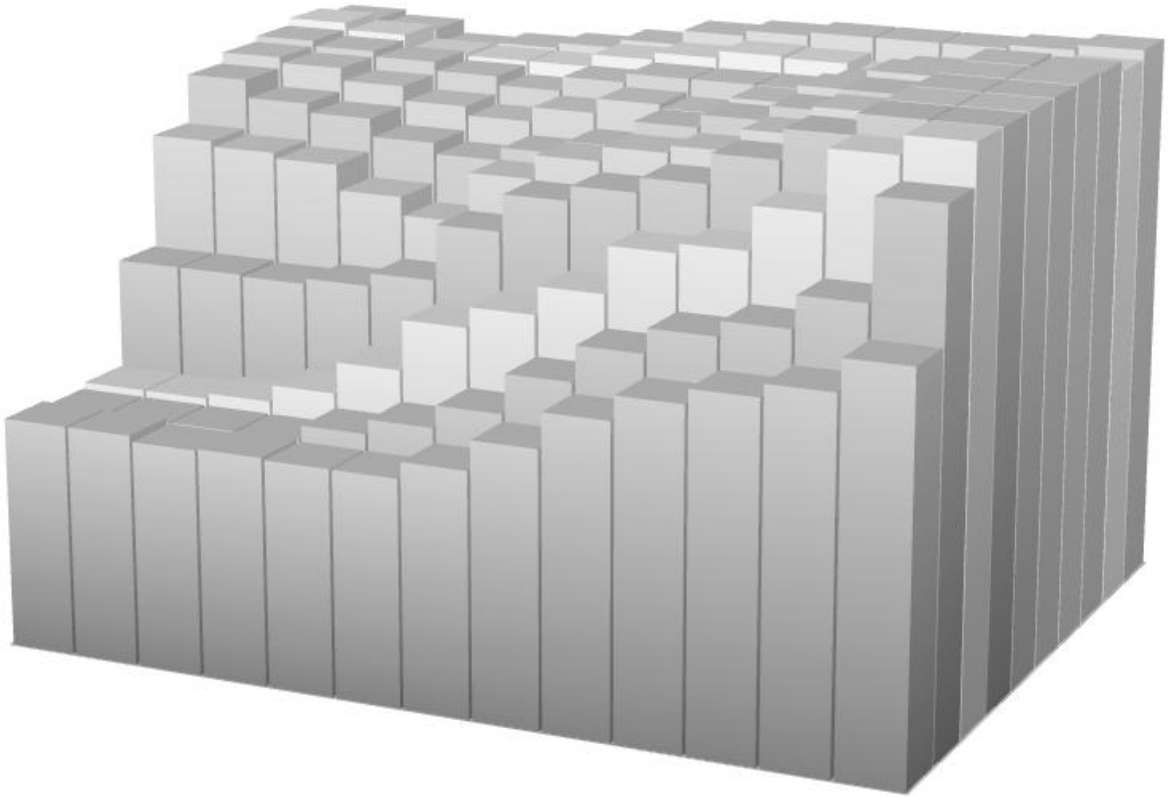
Raster wielokanałowy



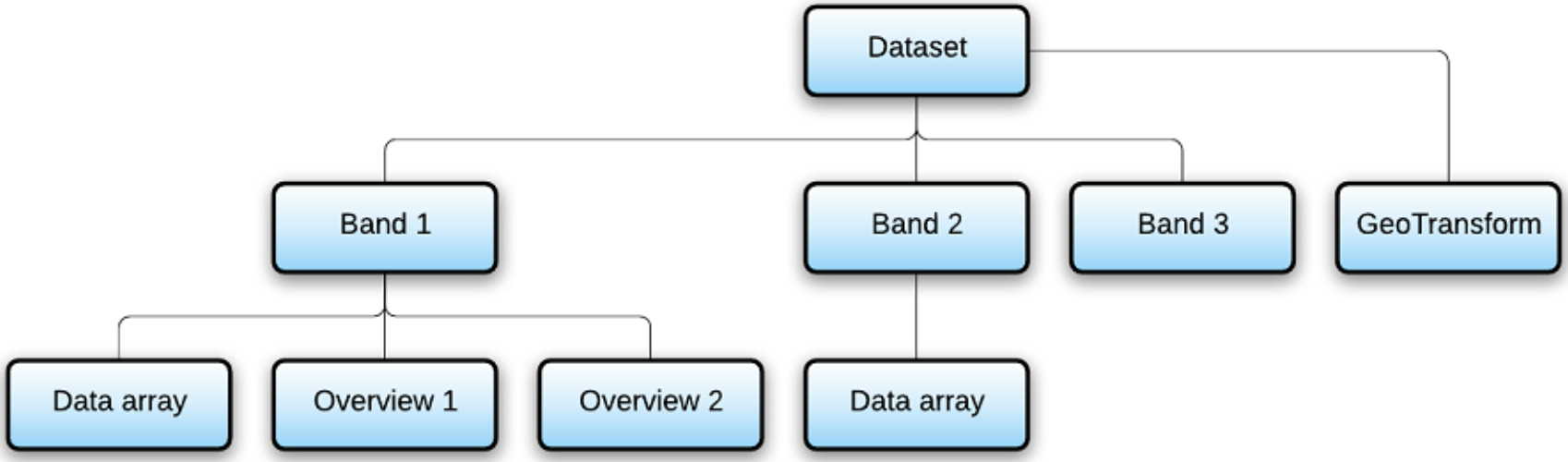
Współrzędne rastra



Przykład rastra
cyfrowy model wysokościowy (CMW)



GDAL raster


Dataset
Band 1
Band 2
Band 3
GeoTransform
Data array
Overview 1
Overview 2
Data array

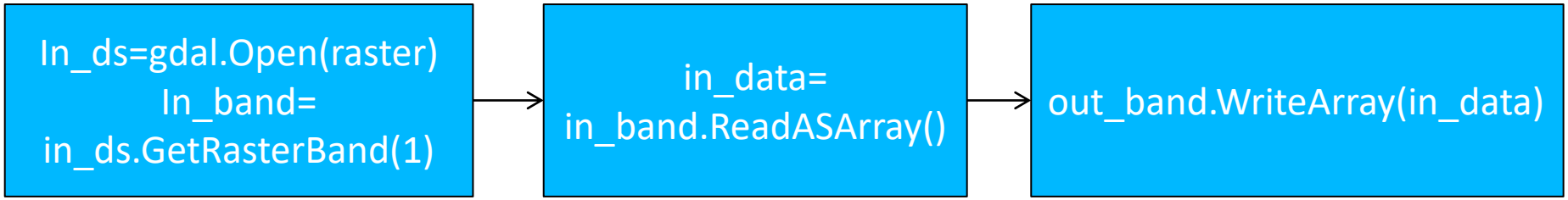
GeoTransform

- GeoTransform[0] /* top left x */
- GeoTransform[1] /* w-e pixel resolution */
- GeoTransform[2] /* 0 */
- GeoTransform[3] /* top left y */
- GeoTransform[4] /* 0 */
- GeoTransform[5] /* n-s pixel resolution (negative value) */

Analizy danych rastrowych w GDAL

Przetwarzanie rastra:

Raster-> NumPy Array->Analiza -> Raster



Odczyt pliku rastrowego

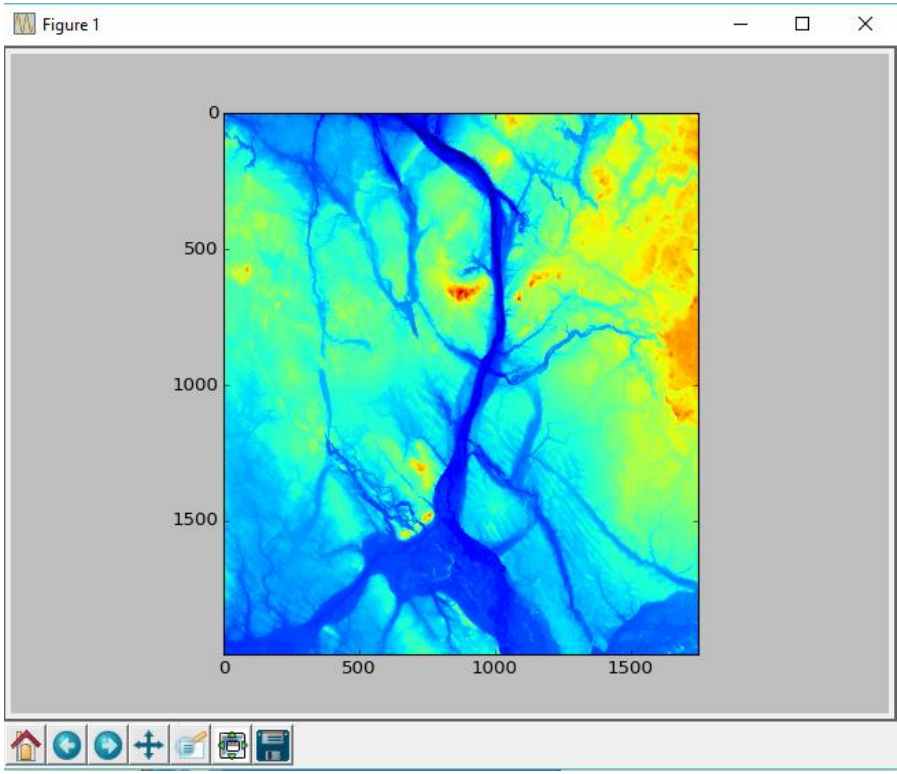
```
import gdal
#otwarcie pliku

In_fn='C:/sciezka_do_pliku/plik.tiff'
in_ds = gdal.Open(in_fn)
#wybranie właściwego kanału
in_band = in_ds.GetRasterBand(1)
#wczytanie do tablicy NumPy
in_data = in_band.ReadAsArray()
```

Wyświetlanie tablicy rastrowej

```
import matplotlib.pyplot as plt
import gdal

ds_raster = gdal.Open(r'C:/GIS/dem.tif')
band = ds_raster.GetRasterBand(1)
array=band.ReadAsArray()
plt.imshow(array)
plt.show()
```



Wczytywanie raster gdal jako tablicay NumPy

In_band.ReadAsArray([xoff], [yoff], [win_xsize], [win_ysize], [buf_xsize],[buf_ysize], [buf_obj])

xoff – kolumna początkowa wczytywania (domyślnie 0)

yoff – wiersz początkowy wczytywania (domyślnie 0)

win_xsize – ilość kolumn do wczytania (domyślnie wszystkie)

win_ysize - ilość wierszy do wczytania (domyślnie wszystkie)

buf_xsize – ilość kolumn tworzonej tablicy (domyślnie win_xsize)

buf_ysize – ilość wierszy tworzonej tablicy (domyslnie win_ysize)

buf _obj – tablica NumPy do wstawiania danych zamiast tworzenia nowej tablicy

Biblioteka NumPy

Podstawowy pakiet do analiz naukowych w Pythonie.

Podstawowym obiektem jest N-wymiarowa tablica (ndarray) oraz szereg funkcji przetwarzających i matematycznych wspierających operacje na całych tablicach bez potrzeby stosowania pętli

Aby uruchomić zainstalowaną uprzednio bibliotekę należy wykonać polecenie

```
import numpy as np
```

https://docs.scipy.org/doc/numpy-dev/user/index.html

Funkcje NumPy

- **ndarray.ndim** – ilość wymiarów tablicy
- **ndarray.shape** – rozmiar poszczególnych wymiarów
- **ndarray.size** - całkowita ilość elementów w tablicy
- **ndarray.dtype** – typ elementów tablicy
- **ndarray.itemsize** – rozmiar elementu w bajtach
- **ndarray.data** – bufor tabeli w pamięci

Tablica NumPy

```
>>> import numpy as np
>>> a = np.array([[1, 3, 4], [2, 7, 6]])
>>> a
array([[1, 3, 4],
       [2, 7, 6]])
>>> a.ndim
2
>>> a.shape
(2, 3)
>>> a.size
6
>>> a.dtype
dtype('int32')
>>> a.itemsize
4
>>> a.data
<memory at 0x03F77C60>
```

NumPy – tworzenie tablic

```
a = np.array([[1, 3, 4], [2, 7, 6]])
b = np.array([[5, 2, 9], [3, 6, 4]])
a+b
array([[ 6,  5, 13],
       [ 5, 13, 10]])
a>b
array([[False,  True, False],
       [False,  True,  True]], dtype=bool)

#wylosuj do tablicy 12-elementowej liczby od 0 do 20
a = np.random.randint(0, 20, 12)
>>> a
array([ 6, 12, 16, 17,  3, 11,  5, 14, 19,  2,  9,  7])
```

NumPy – tworzenie tablic

```
>>>b = np.arange(12) # szybkie tworzenie tablicy jednowymiarowej
>>> b
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
>>>b = np.arange(12).reshape(4,3)  #szybkie tworzenie tablicy dwuwymiarowej
>>> b
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11]])
>>> a=np.ones((3,4),np.int32) #generowanie tabeli z identycznymi danymi
>>> a
array([[1, 1, 1, 1],
       [1, 1, 1, 1],
       [1, 1, 1, 1]])
>>> np.zeros((3,4))
array([[ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.]])
```

NumPy – tworzenie tablic losowych

```
>>> a=np.random.randint(0, 9, 16 ).reshape(4,4)
>>> a
array([[1, 3, 6, 3],
       [8, 6, 1, 8],
       [2, 3, 1, 7],
       [6, 3, 6, 6]])
```

Zmiany rozmiarów tablic

```
>>> a.resize((2,8))
>>> a
array([[0, 1, 0, 0, 1, 0, 0, 0],
       [1, 0, 0, 0, 1, 1, 1, 1]])
>>> a.reshape(8,-1)
array([[0, 1],
       [0, 0],
       [1, 0],
       [0, 0],
       [1, 0],
       [0, 0],
       [1, 1],
       [1, 1]])
>>> a.shape
(2, 8)
```

Sortowanie tablic

```
>>> np.sort(a)
array([[0, 1, 1, 8],
       [2, 2, 6, 8],
       [2, 4, 7, 8],
       [2, 5, 5, 7]])
```

Funkcja NumPy.choose

```
#inicjalizacja listy
>>> a=[1,2,3,4,5,6,7]
>>> np.choose([0,2,4],a)
array([1, 3, 5])
#inicjalizacja listy 2-wym.
>>> a=[[1,2,3],[2,3,4]]
>>> np.choose([0,1,0],a)
#pierwszy element z wiersza  pierwszego 0+1,
#drugi element z wiersza drugiego 1+1
#trzeci element z wiersza pierwszego 0+1
array([1, 3, 3])
>>> np.choose([[0,1,0],[1,1,0]],a)
array([[1, 3, 3],
       [2, 3, 3]])
```

NumPy.choose

```
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11]])
>>> a = np.random.randint(0, 3, 12) .reshape(4,3)
>>> a
array([[1, 1, 1],
       [2, 2, 0],
       [1, 0, 2],
       [1, 2, 1]])
>>> np.choose(a, b)
array([[3, 4, 5],
       [6, 7, 2],
       [3, 1, 8],
       [3, 7, 5]])
```

NumPy.choose (kadrowanie tablicy)

```
>>> mask = np.random.randint(0, 2, 16) .reshape(4,4)
>>> mask
array([[1, 1, 0, 1],
       [0, 1, 1, 1],
       [0, 1, 1, 1],
       [0, 0, 1, 0]])
>>> clip = np.arange(16).reshape(4,4)
>>> clip
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15]])
>>> np.choose(mask, (clip,0)) #wstaw 0 gdzie jest 1
array([[ 0,  0,  2,  0],
       [ 4,  0,  0,  0],
       [ 8,  0,  0,  0],
       [12, 13,  0, 15]])
```

NumPy – obliczenia statystyk

```
#utworzenie tablicy
>>> b = np.arange(9).reshape(3,3)
>>> b
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
#wyliczenie średniej dla całej tablicy
>>> np.mean(b)
4.0
#wyliczenie odchylenia standardowego
>>> np.std(b)
2.581988897471611 2
```

Histogram tablicy rastrowej

```
import gdal
import matplotlib.pyplot as plt

#otwarcie pliku
in_fn=r"C:/GIS/dem.tiff"
in_ds = gdal.Open(in_fn)
#wybranie właściwego kanału
in_band = in_ds.GetRasterBand(1)

#wczytanie do tablicy NumPy
in_data = in_band.ReadAsArray()

#odczyt ilości wierszy i kolumn
num_rows, num_cols = in_data.shape

#utworzenie słownika do zliczenia liczebności poszczególnych wartości
histogram={}
```

Histogram tablicy rastrowej

```
x=[]
y=[]
for row in range(num_rows):
        for col in range(num_cols):
                #odczyt zawartości komórki
                elevation = int(in_data[row, col])
                try:
                        histogram[elevation] += 1
                except KeyError:
                        histogram[elevation] = 1
#obsługa wyników
for elevation in sorted(histogram.keys()):
        print (elevation, histogram[elevation])
        if elevation >0:
                x.append(elevation)
                y.append(histogram[elevation])
plt.bar(x, y)
plt.show()
```