

Sprawozdanie nr 1
Wykrywanie miast, centrów przez analizę Danych
katastralnych.

Wrocław, dnia 19.01.2015

Ewelina Mielczarek
Paulina Sosnowska
Krystian Wielichowski
Grupa 4

Opis zdania

Naszym zadaniem jest wykrycie centrów miejscowości na podstawie danych katastralnych udostępnionych przez geoportal w formie serwisu WFS. Działki położone na terenach zabudowanych są mniejsze niż poza, więc ich zagęszczenie jest większe. Dlatego do wyznaczenia centrów miejscowości zamierzamy wykorzystać analizę skupień, ale w tym sprawozdaniu skupimy się na obróbce danych tak, aby można było je wykorzystać.

Pobranie danych:

Wysłano zapytanie WFS do serwera geoportalu w następującej formie:

http://mapy.geoportal.gov.pl/wss/service/pub/guest/G2_GO_WFS/MapServer/WFSServer?service=wfs&request=getfeature&bbox=53.13,15.36,53.137,15.369&version=1.1.0&typename=Dzialki&srsname=EPSG:4326

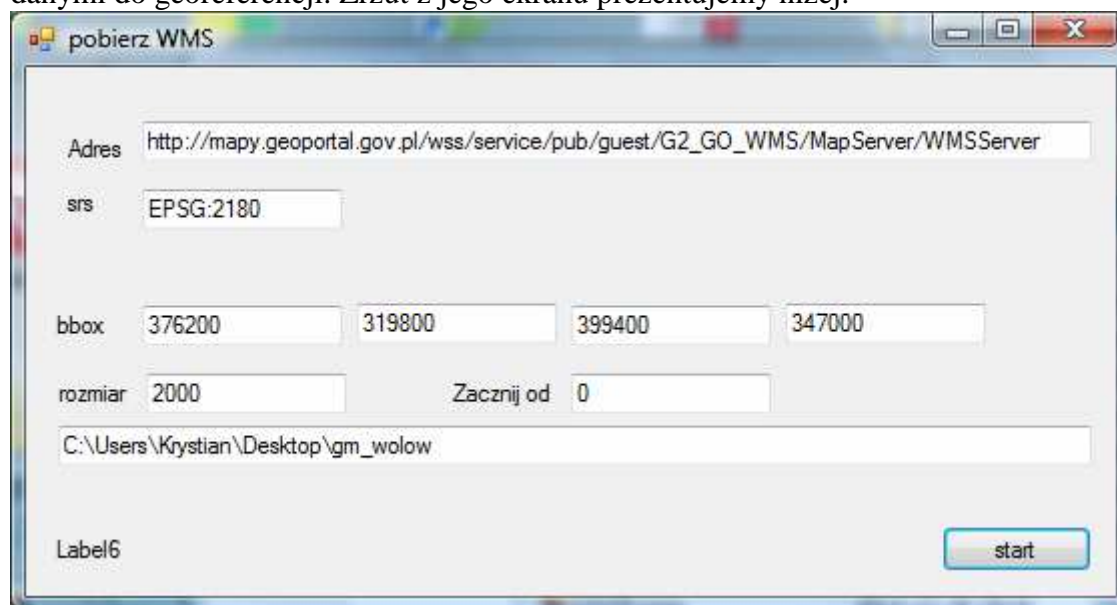
i otrzymano odpowiedź:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceExceptionReport version="1.1.0">
<ServiceException code="AuthorizationFailed">
<![CDATA[No permission to access any of the requested types.]]></ServiceException>
</ServiceExceptionReport>
```

Ta odpowiedź oznacza, że nie jesteśmy odpowiednim użytkownikiem, by pobrać te dane. Aby ominąć urzędniczą procedurę nadawania uprawnień, pobrano dane rastrowe z serwisu WMS i przetworzono do formatu wektorowego. Cały obszar gminy Wołów podzielono na kwadraty o powierzchni 4 ha za pomocą żądania WMS:

http://mapy.geoportal.gov.pl/wss/service/pub/guest/G2_GO_WMS/MapServer/WMServer?service=wms&version=1.1.1&request=GetMap&layers=Dzialki&styles=default&srs=EPSG:2180&bbox=376200,319800,376400,320000&width=2000&height=2000&format=image/png

Pobrano 15775 takich obrazów. Aby zautomatyzować ten proces stworzyliśmy w Visual Basicu program, który oprócz pobrania danych stworzył dla każdego obrazu plik tekstowy z danymi do georeferencji. Zrzut z jego ekranu prezentujemy niżej.



Przetworzenie danych:

Przetworzenie danych rastrowych na wektorowe nastąpiło w programie ArcGis a wykorzystaniem kodu Python'owego:

```
import arcpy
from arcpy import env
import urllib2
from xml.dom import minidom
arcpy.ImportToolbox("C:\Users\Krystian\AppData\Roaming\ESRI\Desktop10.2\ArcToolbox\My Toolboxes\ToolboxMoj.tbx")

folder = 'C:/Users/Krystian/Desktop/gm_wolow2/'
baza = 'D:/dane/gm Wołów/Wołów2.gdb/'

for i in xrange(0, 15775):
    obraz = folder + str(i) + '.png/Band_2'
    wsp = folder + str(i) + '.txt'
    plik = open(wsp, "r")
    w = plik.readline()
    plik.close()
    w = w.split()
    x = float(w[2]) + 0.5
    y = float(w[3]) + 2000 - 0.5
    poligon = baza + 'poligon' + str(i)
    arcpy.RasterToVector(obraz, x, y, poligon)
    warstwa = arcpy.mapping.Layer(poligon)
    bbox = str(warstwa.getExtent().XMin)+ ',' + str(warstwa.getExtent().YMin)+ ',' + str(warstwa.getExtent().XMax)+ ',' + str(warstwa.getExtent().YMax)
    x0 = warstwa.getExtent().XMin
    y0 = warstwa.getExtent().YMax
    r = "http://mapy.geoportal.gov.pl/wss/service/pub/guest/G2_GO_WMS/MapServer/WMServer?service=wms&version=1.1.1&request=GetFeatureInfo&layers=Dzialki&styles=default&srs=EPSG:2180&bbox="
    with arcpy.da.UpdateCursor(poligon, ["SHAPE@XY", "idDzialki", "numer", "powierzchnia"]) as rows:
        for row in rows:
            try:
                cx = row[0][0]
                cy = row[0][1]
                x = (cx - x0)*10
                y = (y0 - cy)*10
                rzadanie = r + bbox + '&width=2000&height=2000&query_layers=Dzialki&x=' + str(x) + '&y=' + str(y)
                odpowiedz = urllib2.urlopen(rzadanie).read()
                xml = 'C:/Users/Krystian/Desktop/xml/x' + str(i) + '.xml'
                plik = open(xml, "w")
                plik.write(odpowiedz)
                plik.close()
                xmldoc = minidom.parse(xml)
                listaObjectow = xmldoc.getElementsByTagName('FIELDS')
                idDzialki = listaObjectow[0].attributes['IDENTYFIKATOR'].value
                nr = listaObjectow[0].attributes['NUMER'].value
                powierzchnia = listaObjectow[0].attributes['SHAPE.AREA'].value
                row[1] = idDzialki
                row[2] = nr
                row[3] = powierzchnia
                rows.updateRow(row)
            except:
```

```

        next
    mxd = arcpy.mapping.MapDocument("CURRENT")
    for df in arcpy.mapping.ListDataFrames(mxd):
        for lyr in arcpy.mapping.ListLayers(mxd, "", df):
            if lyr.name.lower() == "poligon"+str(i):
                arcpy.mapping.RemoveLayer(df, lyr)
print i

```

Ten kod po kolei odczytuje każdy obraz i plik z jego danymi do georeferencji. Następnie z wykorzystaniem wykonanego przez nas narzędzia raster2vector przetworzyliśmy je do formatu wektorowego i zapisaliśmy do bazy danych.

Następnie za pomocą żądania getFeatureInfo wypełniliśmy kolumnę idDziałki. Po przetworzeniu wszystkich plików otrzymaliśmy 15775 warstw wektorowych, a potrzebowaliśmy jednej to wykonaliśmy operację scalenia za pomocą polecenia:

```
arcpy.Merge_management(poligony,polaczone)
```

Potem połączyliśmy wszystkie części działek o tym samym numerze za pomocą polecenia:

```
arcpy.Dissolve_management(polaczone, wynik, ['idDzialki'])
```

Następnie otrzymaną warstwę przycięliśmy do obrysu gminy Wołów i wyeksportowaliśmy do formatu shapefile.

Zapis do bazy danych PostGis

W programie do zarządzanie bazą danych pgAdmin III utworzyliśmy bazę danych postgres. Aby mogła przechowywać dane przestrzenne należy dodać rozszerzenie Postgis za pomocą zapytania SQL:

```
create extension postgis
```

Następnie wczytaliśmy dane do bazy za pomocą narzędzia shp2pgsql. Można je włączyć z linii komend

```
shp2pgsql -s 2180 D:\dane\wołów\dzialki gminaWolow.dzialki >
dzialki.sql
```

W wyniku otrzymaliśmy plik SQL, który możemy wykonać w bazie danych. Tak przygotowane dane możemy zobaczyć np. w programie QGIS.

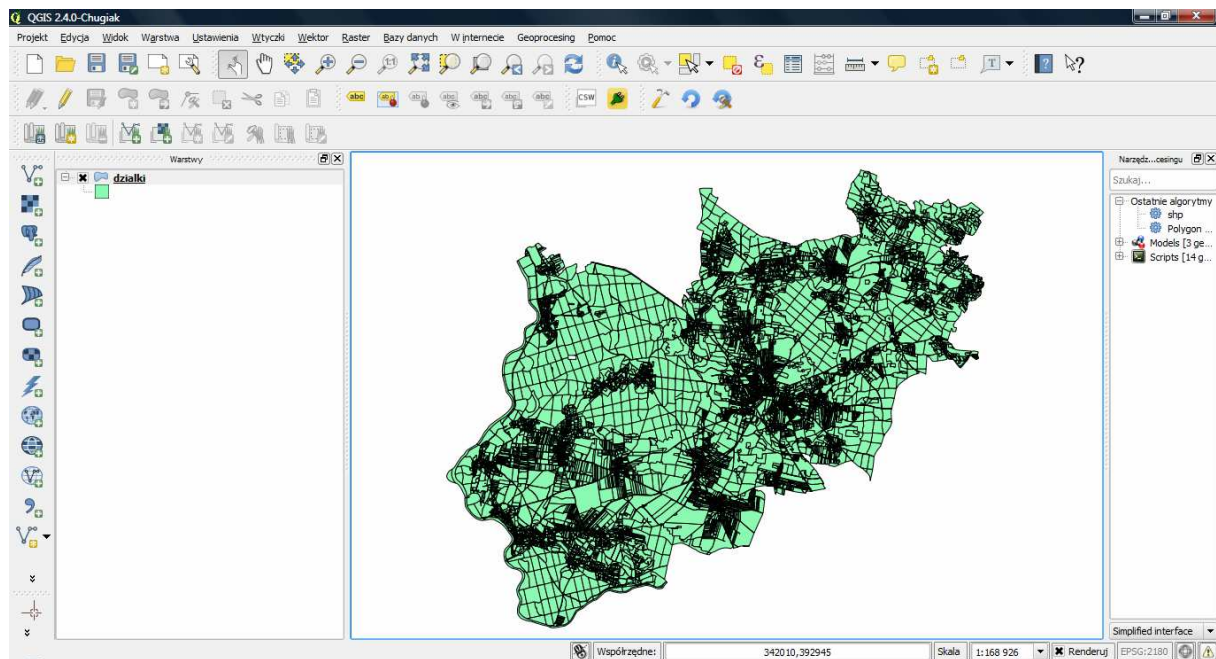


Tabela atrybutów:

Tabela atrybutów - działki :: Łączenie obiektów: 20800, filtrowanych: 20800, wybranych: 0

	gid	objectid	iddziałki	shape_leng	shape_area
0	1	17674	022202_2.0023...	505.147322935	792.764999998
1	2	20699	022203_4.0001...	4341.47079401	506070.894125
2	3	20700	022203_4.0001...	2090.49385972	107782.77
3	4	20701	022203_4.0001...	439.784744606	6140.47
4	5	20702	022203_4.0001...	236.013252582	1256.2790625
5	6	20703	022203_4.0001...	1182.78296473	4127.55
6	7	20704	022203_4.0001...	1244.59125769	70992.492146
7	8	20705	022203_4.0001...	460.930905387	10092.31
8	9	20706	022203_4.0001...	293.782373289	2624.04
9	10	20707	022203_4.0001...	622.450126658	5448.215
10	56	20753	022203_4.0001...	227.379272582	2518.725
11	300	20997	022203_4.0001...	77.5366484035	371.409999999
12	11	20708	022203_4.0001...	1149.40850432	28513.68
13	12	20709	022203_4.0001...	214.878320148	2501.46
14	13	20710	022203_4.0001...	206.968347654	2446.82
15	14	20711	022203_4.0001...	455.293594277	12821.07
16	15	20712	022203_4.0001...	2992.4745134	22950.9574841
17	16	20713	022203_4.0001...	384.647739664	839.884999998
18	17	20714	022203_4.0001...	877.253306747	26467.525
19	18	20715	022203_4.0001...	5206.05585574	606427.811929

Pokaż wszystkie obiekty