

## **SPIS TREŚCI**

<b>1. Wstęp</b>	<b>4</b>
<b>2. Rozwiązania zastosowane w pracy</b>	<b>5</b>
<b>3. Wybór oprogramowania GIS</b>	<b>6</b>
<b>3.1. Analiza widoczności dla radaru w Zabierzowie w programie QGIS</b>	<b>6</b>
<b>3.2. Analiza widoczności dla radaru w Zabierzowie w programie ArcGIS</b>	<b>11</b>
<b>3.3. Wnioski</b>	<b>15</b>
<b>4. Automatyzacja działań oprogramowania QGIS</b>	<b>16</b>
<b>4.1. Struktura plików</b>	<b>16</b>
<b>4.2. Uruchomienie narzędzia</b>	<b>23</b>
<b>4.3. Wnioski</b>	<b>25</b>
<b>5. Wnioski końcowe i podsumowanie</b>	<b>26</b>
<b>6. Słownik skrótów</b>	<b>27</b>
<b>7. Bibliografia</b>	<b>27</b>
<b>8. Załączniki</b>	<b>28</b>

## 1. Wstęp

Wśród możliwych do przeprowadzenia przy pomocy oprogramowania GIS (ang. *Geographic Information Systems*) analiz przestrzennych znajduje się analiza widoczności (ang. *viewshed*). Analiza bazuje na NMT (Numeryczny Model Terenu) i daje, proporcjonalnie od dokładności zastosowanego modelu terenu oraz algorytmu analizy, możliwe dokładne oszacowanie zasięgu widoczności z danej lokalizacji w przestrzeni. Jeżeli analiza zezwala na ustalenie dodatkowych różnic w wysokości posadowienia względem terenu źródła oraz celu obserwacji, funkcjonalność taka staje się bardzo użyteczna dla potrzeb służb ruchu lotniczego. Daje bowiem możliwość przykładowych analiz: widoczności pola wzlotów z wież KRL (Kontroli Ruchu Lotniczego), możliwości połączenia teletechnicznego przy pomocy radiolinii w paśmie X czy możliwości dozoru radiolokacyjnego w pasmach L i S.

Celem pracy jest przeprowadzenie analizy w wybranym oprogramowaniu GIS oraz próba zautomatyzowania tego procesu. Zostaje to osiągnięte przez serię skryptów które są alternatywnym dla GUI (ang. *Graphic User Interface*) sposobem interakcji z oprogramowaniem GIS. Użytkownik może zatem stworzyć listę interesujących go lokalizacji - w formie odpowiednio sformatowanego pliku i następnie uruchomić skrypt, który bez konieczności dalszej interakcji wygeneruje pliki zawierające wynik analiz dla wszystkich podanych lokalizacji. Ostatnim etapem użycia jest wymagające minimum znajomości oprogramowania GIS wyświetlenie plików wynikowych na tle dowolnego zobrazowania terenu.

Rozwiązanie ma zatem stanowić narzędzie umożliwiające korzystanie z konkretnej funkcjonalności oprogramowania GIS osobom pracującym na co dzień z innymi niż GIS systemami i oprogramowaniami.

## 2. Rozwiązania zastosowane w pracy

Do opracowania prezentowanego rozwiązania posłużono się następująco skonfigurowanym środowiskiem:

- komputer PC z procesorem Intel(R) Core(TM) i7-4510U i 16GB RAM,
- system operacyjny Windows 7 Professional z uprawnieniami administracyjnymi,
- oprogramowanie ArcGIS 10.4.1,
- oprogramowanie QGIS 2.18.19(LTR - ang. *Long Term Release*) wraz ze składowymi;
  - GRASS GIS 7.4.0,
  - SAGA GIS 2.3.2,
  - Python 2.7.

W pracy korzystano z dostępnych bezpłatnie danych:

L.P.	rodzaj danych	typ pliku	źródło	lokalizacja
1	NMT 100m	tekstowy	CODGIK	<a href="http://www.codgik.gov.pl/index.php/darmowe-dane/nmt-100.html">http://www.codgik.gov.pl/index.php/darmowe-dane/nmt-100.html</a>
2	Państwowy Rejestr Granic	<i>ESRI shapefile</i>	CODGIK	<a href="https://www.gis-support.pl/downloads/wojewodztwa.zip">https://www.gis-support.pl/downloads/wojewodztwa.zip</a>

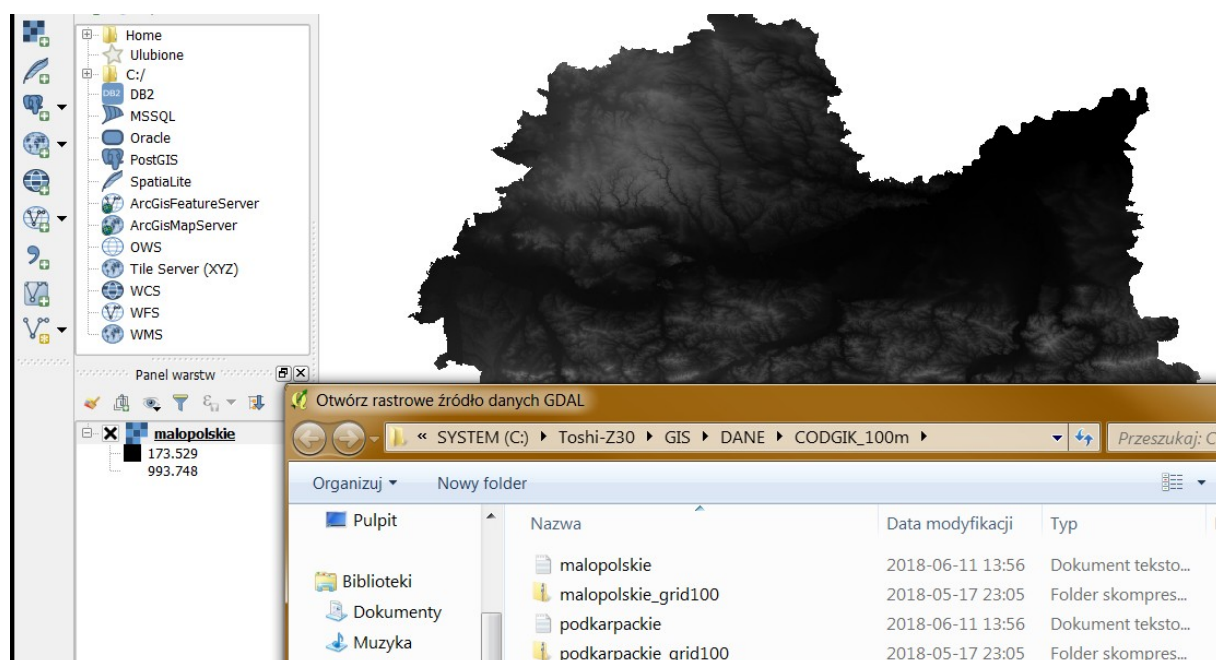
Tab.1. Zestawienie rodzajów i źródeł danych przestrzennych

### 3. Wybór oprogramowania GIS

W rozdziale opisano analizę zasięgu (widoczności) dla radaru KRL w Zabierzowie k.Krakowa. Obliczenia wykonano dla założenia, że antena posadowiona jest na wieży radarowej 40m nad terenem a cel leci 300m nad terenem. Analizy wykonano w oprogramowaniu ArcGIS i QGIS (z GRASS i SAGA) oraz dla porównania w programie *Spx Radar Coverage* firmy *Cambridge Pixel*. Opracowania kartograficzne przeprowadzonych analiz są załącznikami do pracy.

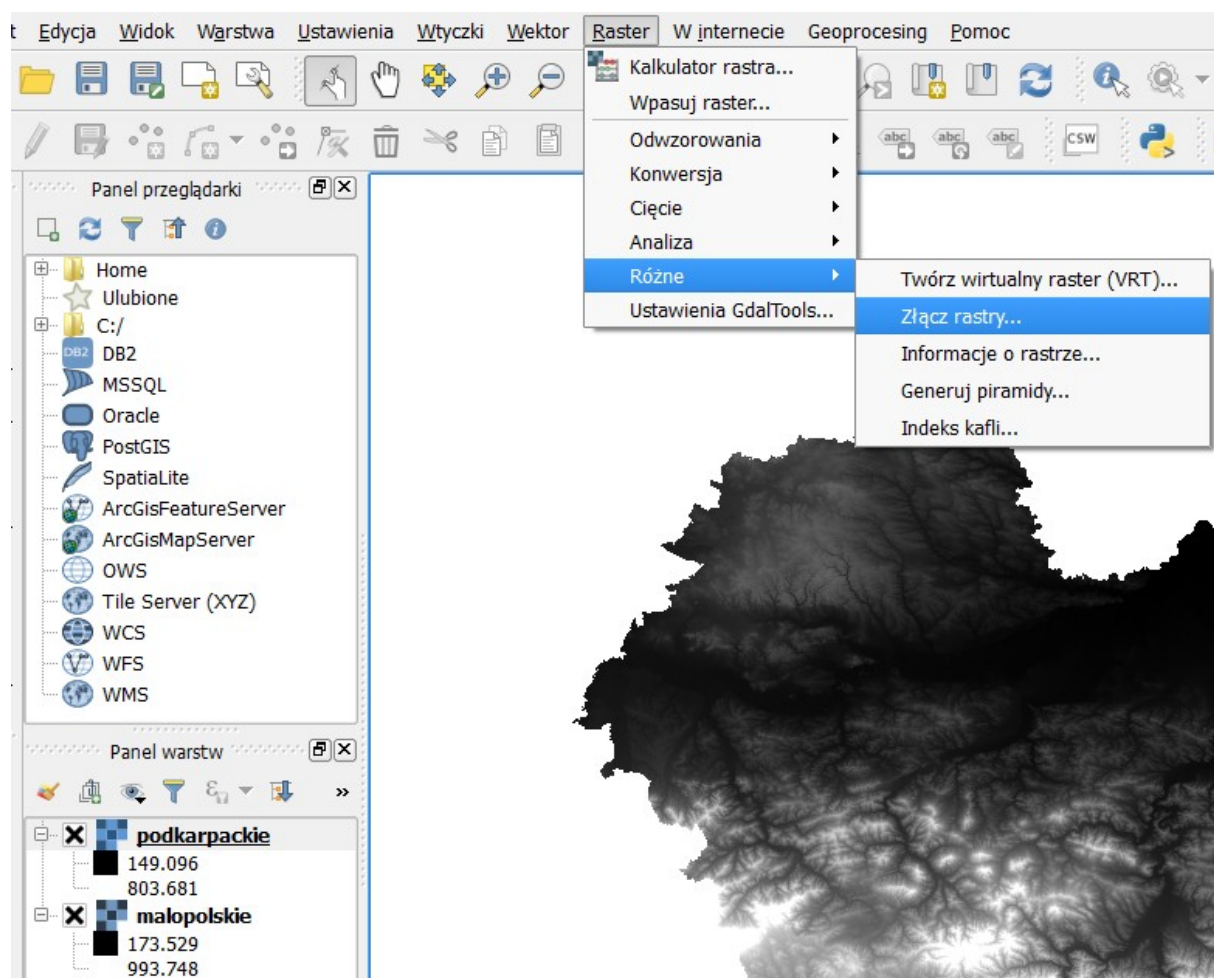
#### 3.1. Analiza widoczności dla radaru w Zabierzowie w programie QGIS

Pierwszym krokiem analizy jest stworzenie NMT przez zaimportowanie plików tekstowych dla interesujących nas województw z CODGIK. Pamiętać należy, że dane z CODGIK zapisane są w PL-1992 (EPSG:2180) i taki układ należy zadeklarować importując plik. Dane dodajemy przez ikonę „Dodaj warstwę rastrową” na lewej krawędzi obszaru pracy.



Rys.1. Import danych tekstowych NMT 100m CODGIK do warstwy rastrowej

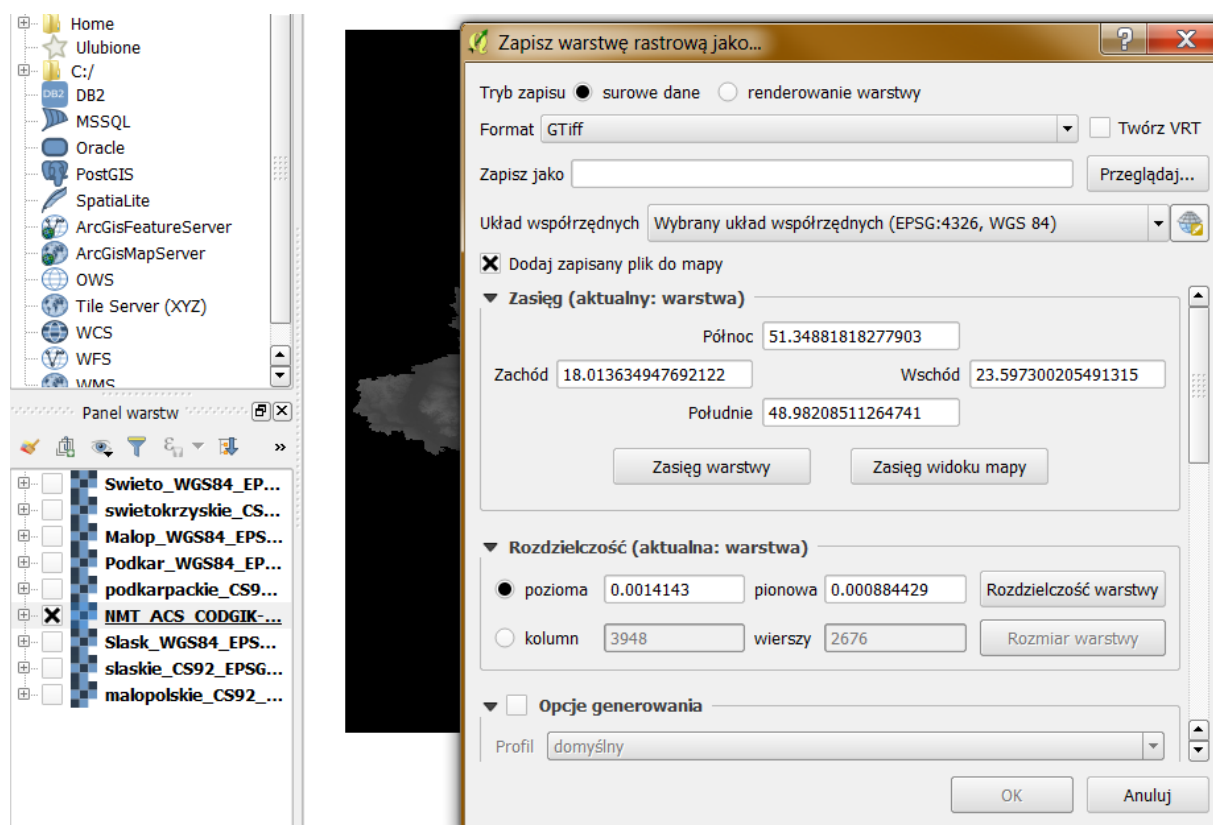
Po zaimportowaniu danych interesujących nas województw skalamy wynikowy raster.



Rys.2. Scalenie rastrów zawierających NMT poszczególnych województw

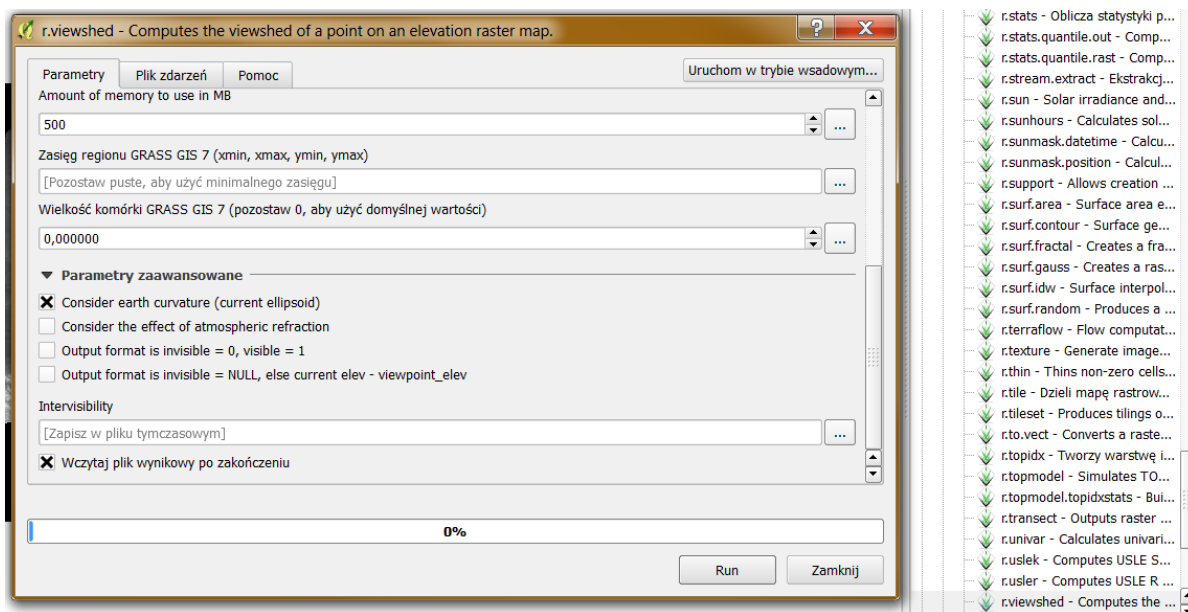
Otrzymaną warstwę zapisujemy do pliku gTiff jako NMT.tif.

Będzie to plik bazowy dla wszystkich dalszych analiz.



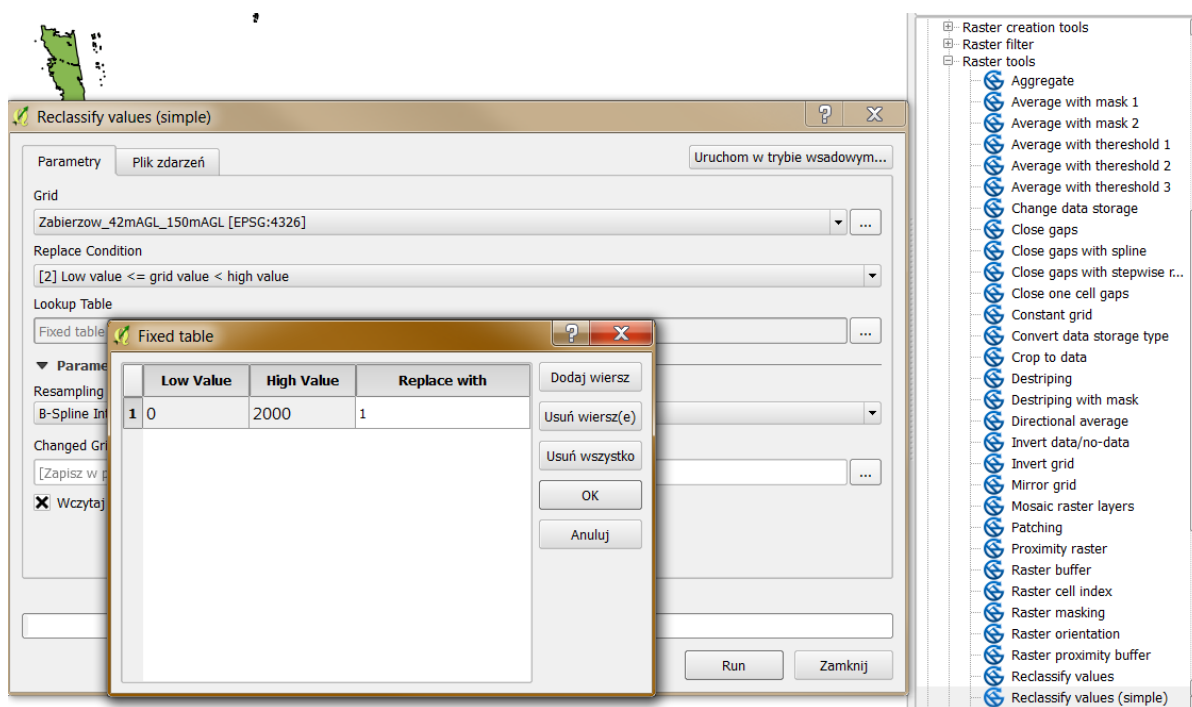
Rys.3. Zapis warstwy połączonych rastrow do formatu gTiff.

Możemy już wykonać analizę widoczności (*viewshed*). Z menu górnego „Geoprocessing – Narzędzia” wybieramy w sekcji która pojawi się przy prawej krawędzi obszaru pracy „Polecenia GRASS GIS 7 – Raster – r.viewshed...”. Podajemy parametry do analizy dla interesującej nas lokalizacji. Wyczerpujące objaśnienie dla wszystkich parametrów znajduje się w zakładce „Pomoc” okienka analizy.



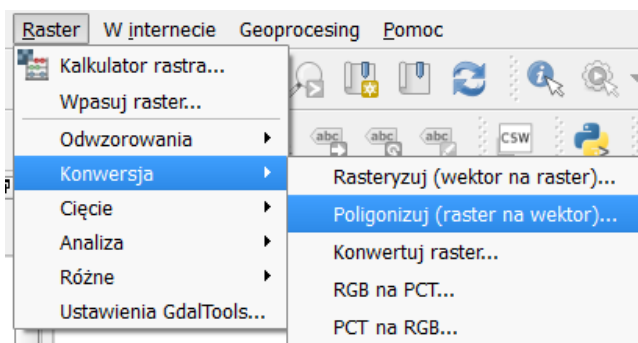
Rys.4. Sekcja narzędzi geoprocesingu i okienko analizy *r.viewshed* GRASS

Na warstwie wynikowej analizy wykonujemy reklasyfikację rastra, korzystając ponownie z menu „Geoprocesing – Narzędzia”. Tym razem w sekcji narzędzi po prawej stronie wybieramy „SAGA (2.3.2)... – Raster tools – Reclassify value (simple)”. Jako wejściowy „Grid” wybieramy warstwę naszej analizy, jako „Replace condition” wybieramy przypadek [2] i modyfikujemy „Lookup Table” do wartości wymiarów 1x1. Wartości graniczne dobieramy tak aby cały raster przyjął jedną wartość.

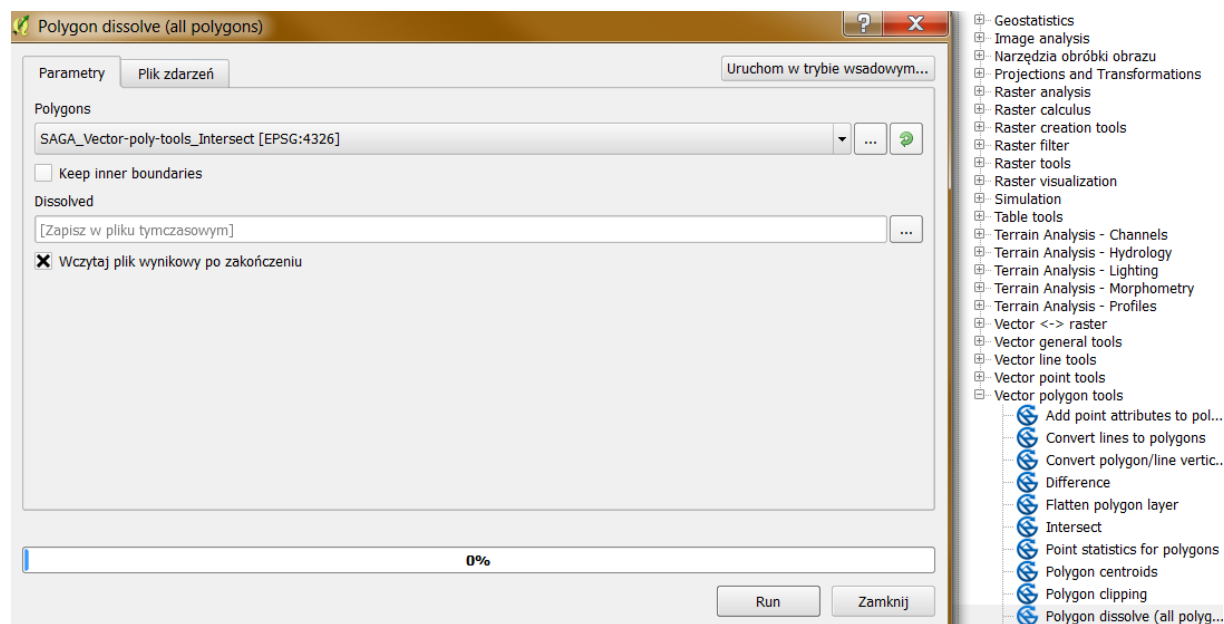


Rys.5. Reklasyfikacji rastra przy pomocy funkcji programu SAGA

Wynikową warstwę poligonizujemy przy pomocy natywnego dla QGIS narzędzia w menu górnym „Raster – Konwersja – Poligonizuj (raster na wektor)...”. Otrzymamy warstwę wektorową której, dzięki wcześniejszej reklasyfikacji bazowego rastra, tabela atrybutów zawierać będzie jedną wartość we wszystkich wierszach. Aby zadbać o poprawność wykonania następnych operacji na tak otrzymanej warstwie wektorowej używamy jeszcze funkcji SAGA „*Polygon dissolve (all polygons)*”.



Rys.6. Konwersja warstwy rastrowej na wektorową.



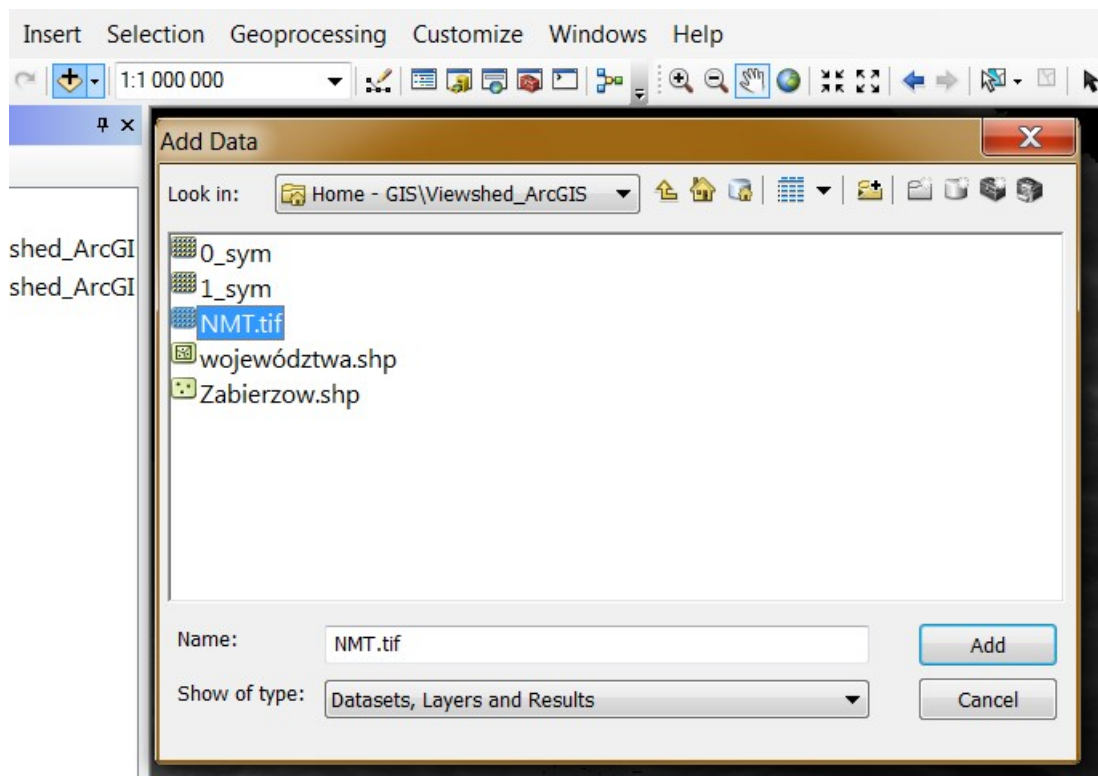
Rys.7. Agregacja obiektów warstwy wektorowej przy pomocy funkcji programu SAGA

Na bazie zajmującej kilkadziesiąt MB warstwy rastrowej NMT.tif otrzymaliśmy kompaktową warstwę wektorową analizy widoczności z punktu o podanej lokalizacji i dla zadanych wysokości źródło-cel. Warstwę tą wystarczy wyświetlić na dowolnym tle (warstwie) obrazującej interesujący nas fragment przestrzeni.



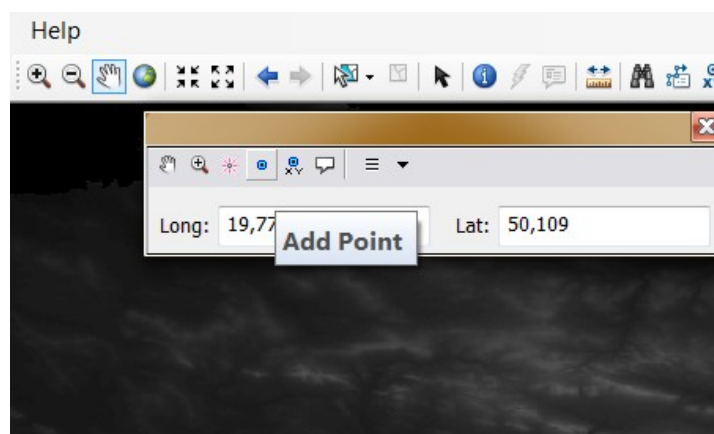
### 3.2. Analiza widoczności dla radaru w Zabierzowie w programie ArcGIS

W analizie wykorzystano NMT w formie pliku gTiff opracowanego w sposób opisany w poprzednim punkcie. W aplikacji ArcMap dodajemy warstwę NMT przez „Add Data”.



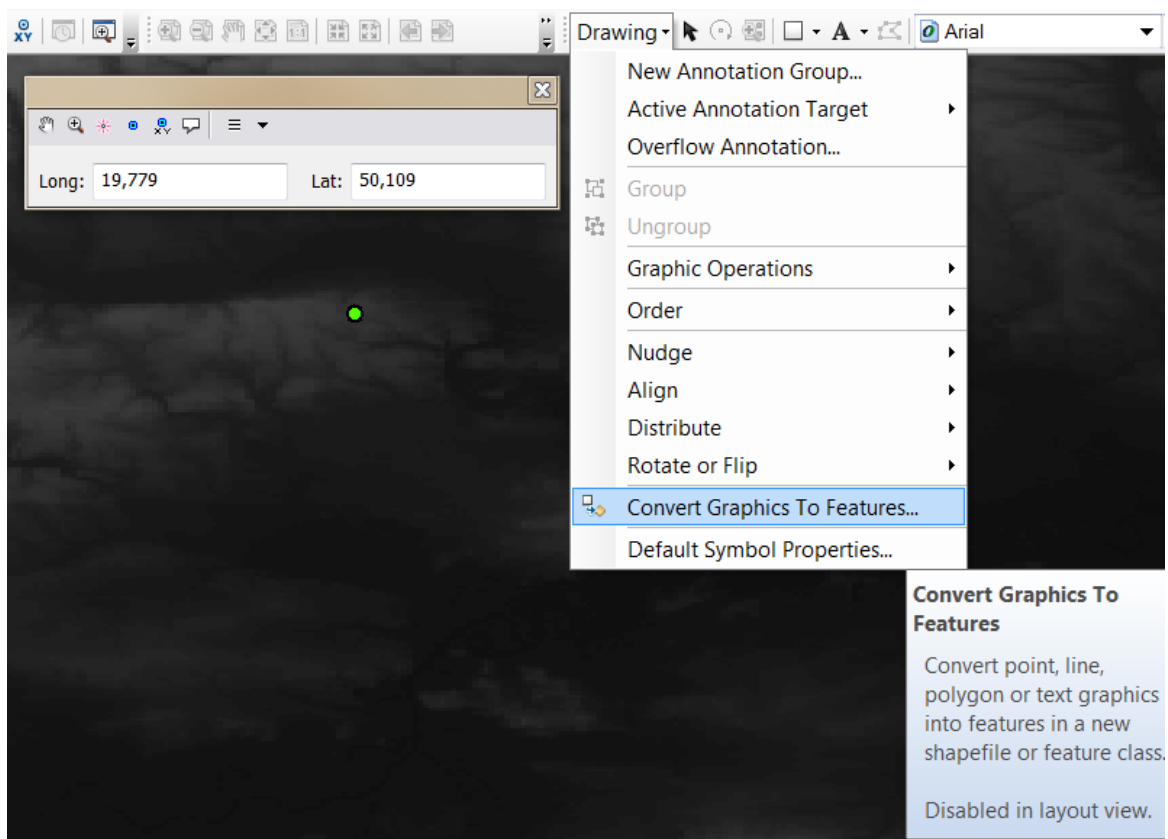
Rys.8. Dodanie NMT (gTiff) do obszaru pracy aplikacji ArcMap przez „Add Data”

Używając funkcjonalności „Go TO XY” zadajemy współrzędne lokalizacji radaru a następnie klikamy „Add Point” i „Zoom To”.

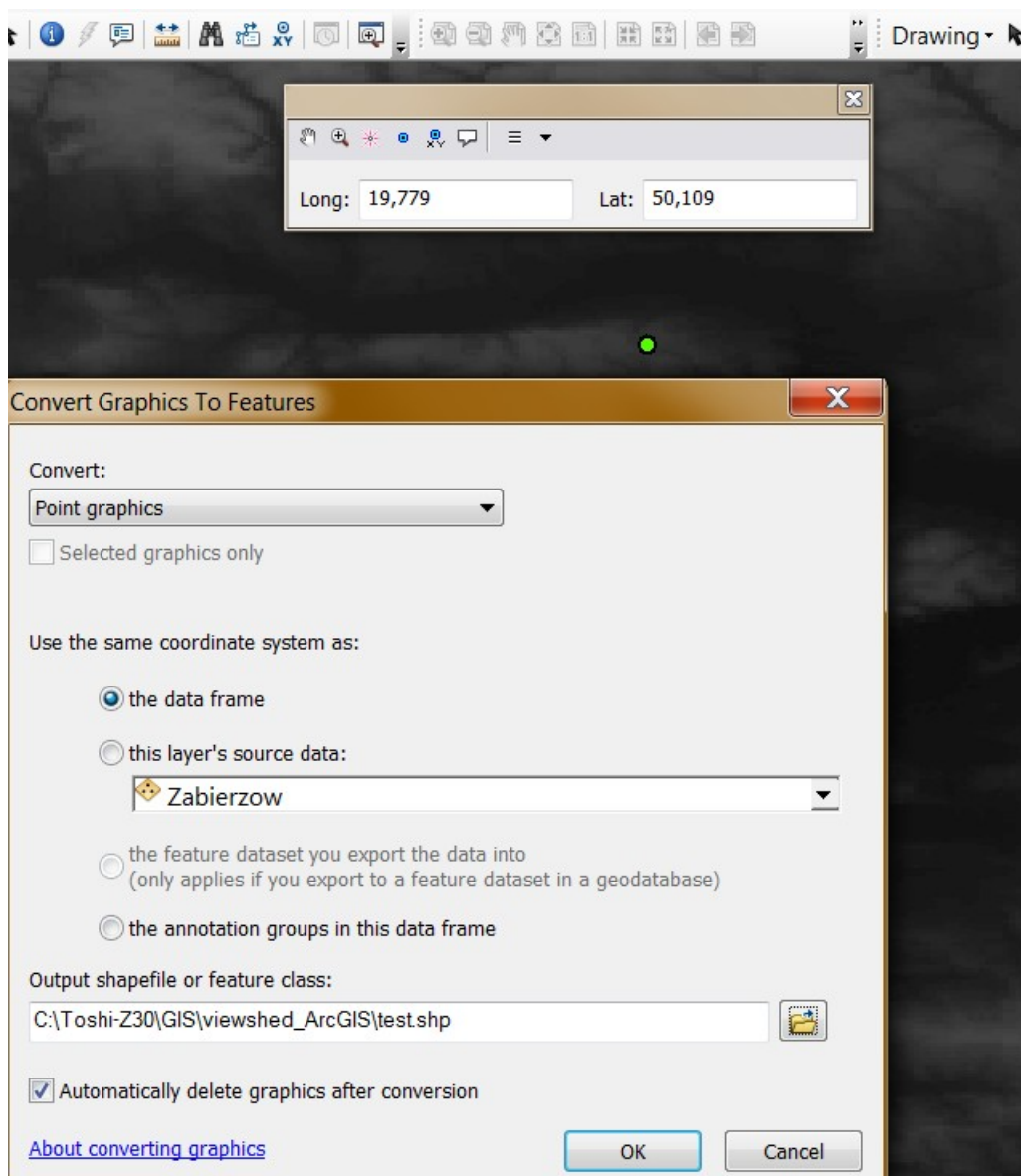


Rys.9. Wskazanie lokalizacji radaru przy pomocy „Go To XY”

Wskazany punkt zamieniamy na warstwę opisującą lokalizację używając funkcjonalności „*Drawing – Convert Graphics to Features – Automatically delete graphics after conversion*” i edytując nazwę i lokalizację pliku wynikowego „*Output shapefile or feature class*”.

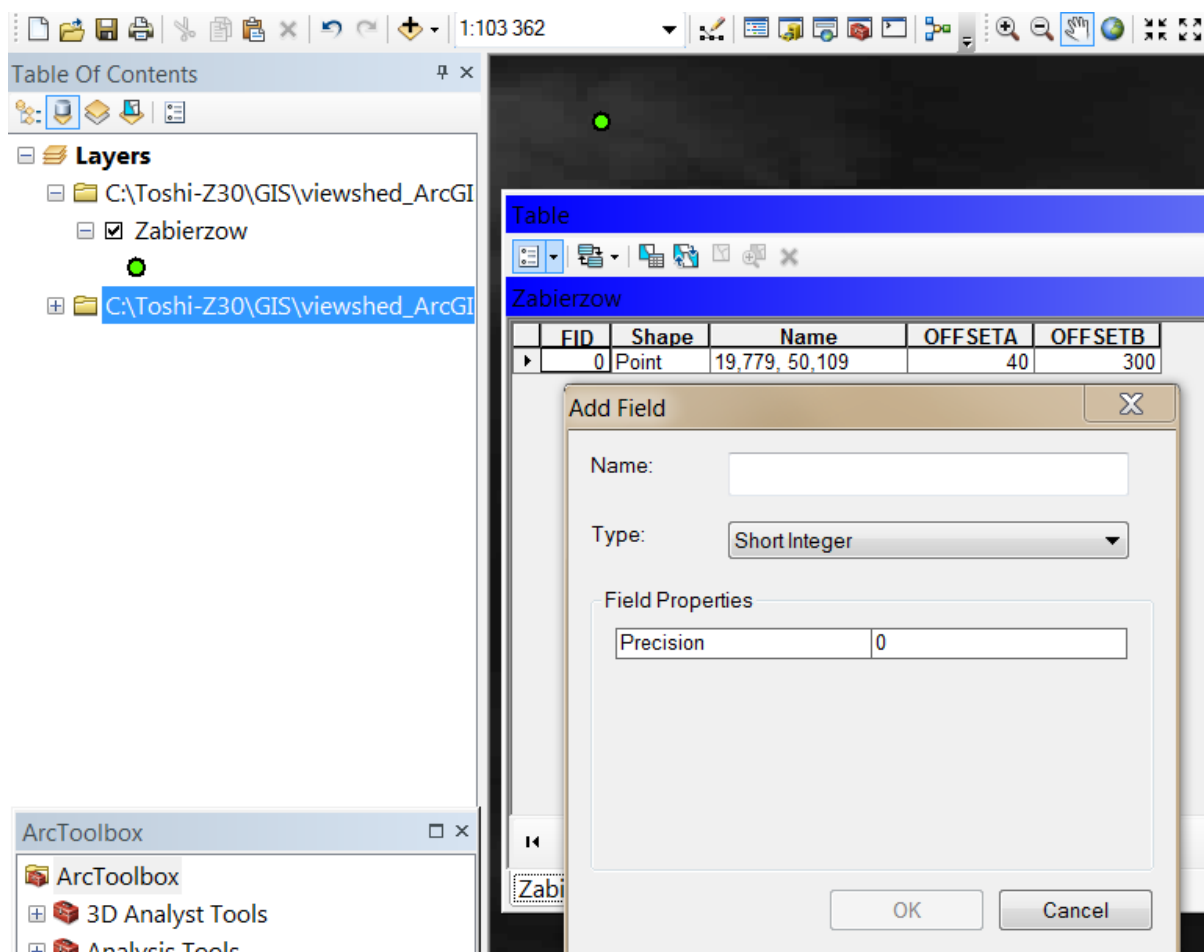


Rys.10a. Uruchomienie procesu konwersji punktu na warstwę.



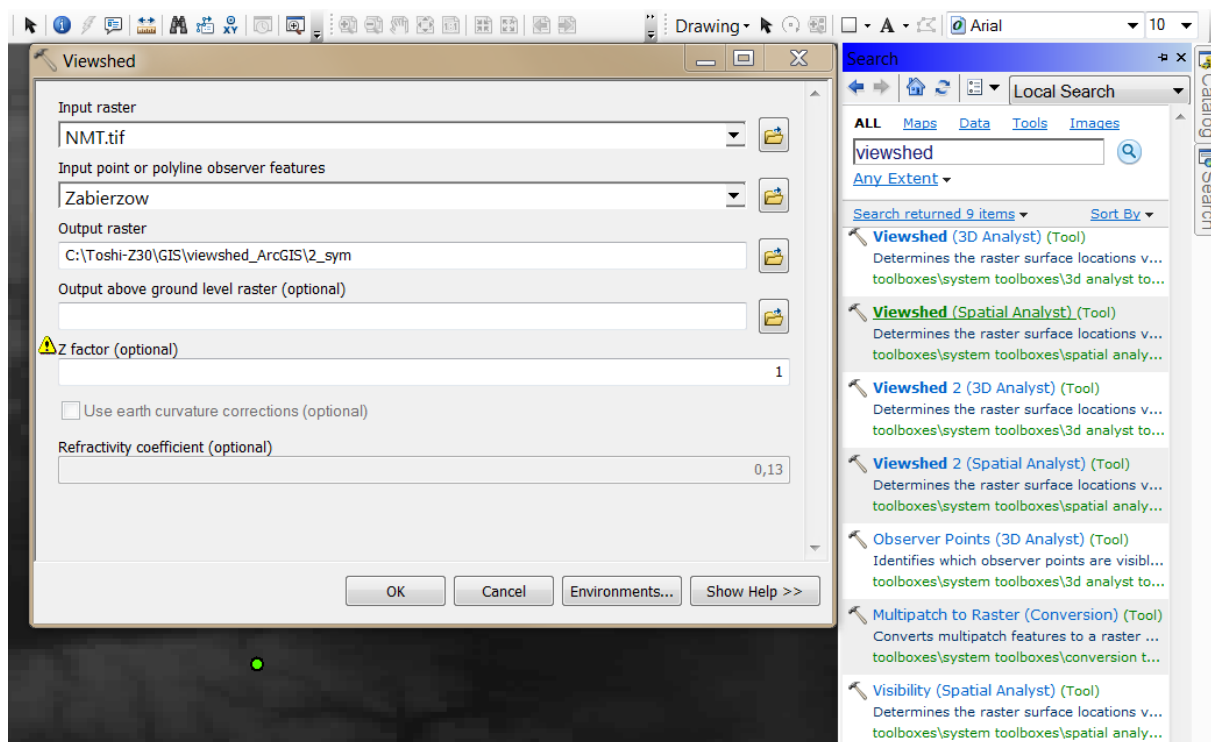
Rys.10b. Ustalenie parametrów konwersji punktu na warstwę.

Klikając prawym klawiszem myszy w warstwę wynikową operacji wybieramy „Add Field” i dodajemy kolejno pola „OFFSETA” i „OFFSETB”. Pierwsze pole przechowuje wysokość obserwatora nad teren (posadowienia anteny). Druga to wysokość celu nad terenem. Uwaga! Nazwa pól musi być dokładnie jak podano, z wielkością znaków włącznie. Pozostałe parametry podano na Rys.11.



Rys.11. Uruchomienie i parametry edycji tabeli atrybutów warstwy lokalizacji

Przy pomocy zakładki „Search” znajdującej się na prawej krawędzi przestrzeni roboczej ArcMap wyszukujemy hasło *viewshed* i wybieramy „*Viewshed (Spatial Analyst)*”. Jako „*Input raster*” wskazujemy plik .tif załadowanego NMT wcześniej NMT. Jako „*Input point...*” wskazujemy zaś warstwę opisującą lokalizację. Pozostaje podać nazwę i lokalizację wyjściowego pliku rastrowego wykonywanej analizy.



Rys.12. Uruchomienie i parametry analizy widoczności

### 3.3. Wnioski

Analiza przeprowadzona w oprogramowaniu QGIS jest bardzo zbliżona do tej przeprowadzonej w programie *Spx Radar Coverage* – przytoczonej tu dla porównania. Niewielkie różnice w tych dwóch wynikach mogą wynikać z działania programu na innym NMT (SRTM 90m). Oprogramowanie *Spx* jest punktem wyjściowym niniejszej pracy – próby uruchomienia podobnego rozwiązania z funkcjonalnością rozszerzoną o pracę na warstwach z wieloma lokalizacjami oraz tryb pracy wsadowej, bez GUI. Więcej na temat ww. programu przeczytać można na stronie producenta <https://www.cambridgepixel.com/products/Radar-Coverage-Tool/>.

Znaczne różnice są natomiast widoczne w analizie przeprowadzonej w ArcGIS. O ile na kierunku północ-południe wyniki są zgodne z dwoma pozostałymi analizami – da się rozróżnić bardzo podobne kształty w poligonach widoczności, o tyle w kierunku wschód-zachód wyniki odbiegają znacznie. Możliwym powodem jest nieciągłość danych NMT na styku województw – składowych NMT powstałego z połączenia plików z CODGIK. Algorytm *viewshed* w ArcGIS może być na taki problem bardziej wrażliwy niż ten wbudowany w GRASS.

## **4. Automatyzacja działań oprogramowania QGIS**

Jak wspomniano we wstępie pracy prezentowane tu rozwiązanie ma stanowić narzędzie umożliwiające skryptową obsługę oprogramowania GIS. Skrypt stanowiący zapis pojedynczej analizy widoczności (opisanej wcześniej) wraz z operacjami pośrednimi ma być wykonywany w pętli. W każdej iteracji pętli jako zmienne dla skryptu podawane są wartości odczytywane z pliku stworzonego przez użytkownika. Plik ten zawiera linie znaków, z których każda opisuje w pewien zdefiniowany sposób pojedynczą lokalizację dla której przeprowadzona ma być analiza. Po przeanalizowaniu wszystkich linii pliku wejściowego narzędzie kończy pracę i podaje wyniki w formie plików. Pliki można wyświetlić przy pomocy oprogramowania GIS i dodać wedle uznania pliki (warstwy) pomocnicze ułatwiające wizualną ocenę wyników.

Dla przeprowadzenia analiz widoczności zastosowano oprogramowanie QGIS.

Dla skryptów pomocniczych zastosowano technologię plików wsadowych linii komend systemu operacyjnego Windows (.bat).

Dla skryptu obsługującego funkcjonalność GIS zastosowano natywną dla oprogramowania QGIS technologię Python (.py).

### **4.1. Struktura plików**

W wykonanej pracy przyjęto założenia dotyczące lokalizacji dyskowej plików:

- oprogramowanie QGIS zainstalowane jest w domyślnym dla wersji 2.18.19 folderze

C:\Program Files\QGIS 2.18\,

- wszystkie pliki wejściowe i wyjściowe oraz operujące na nich skrypty znajdują się w założonym przez użytkownika folderze C:\QGIS.

Zawartość folderu C:\QGIS stanowi następująca struktura plików:

### \dane

#### **- NMT.tif**

Numeryczny model terenu dla czterech południowych województw Polski: małopolskiego, podkarpackiego, śląskiego i świętokrzyskiego powstał na bazie danych udostępnionych darmowo przez CODGIK.

Sumaryczna wielkość czterech plików dla wymienionych województw wynosi około 32MB. Sposób pobrania danych oraz ich zaimportowania do oprogramowania GIS zostały opisane w rozdziale dotyczącym analizy widoczności w QGIS. Użytkownik może zastosować dowolny plik zawierający NMT Polski w formacie GTiff - na przykład opracowany na podstawie danych SRTM (ang. *Shuttle Radar Topography Mission*) w wersji 30m. Należy nadać takiemu plikowi nazwę NMT.tif i umieścić go w omawianej lokalizacji dyskowej.

#### **- pliki zobrazowania**

Dowolne pliki służące za warstwy podkładowe pomagające zobrazować końcowe wyniki graficznej analizy zasięgu (widoczności). Przykładowo zastosować można plik typu ESRI *shapefile* z konturową mapą województw, lub opracowanie danego obszaru typu OpenStreetMap. Po zakończeniu zautomatyzowanej analizy użytkownik importuje w GUI QGIS warstwy wynikowe analizy oraz właśnie zapisane w omawianych plikach warstwy zobrazowania obszaru.

Listingi opisanych poniżej skryptów stanowią załącznik do pracy.

### \skrypty

#### **- uruchom.bat**

Środowisko Python wbudowane jest w pakiet oprogramowania QGIS, dzięki czemu nie trzeba instalować go oddzielnie. Potrzebne jest jedynie odpowiednie zainicjalizowanie środowiska i uruchomienie skryptu napisanego w języku Python, za co między innymi odpowiedzialny jest w niniejszej pracy omawiany plik wsadowy. Następuje to kolejno w sekcjach INIT\_PYTHON oraz START skryptu. W sekcji INIT\_BATCH następuje zadeklarowanie zmiennych koniecznych dla iteracyjnego działania skryptu LOS.py - to znaczy uruchomienia go w pętli której ilość powtórzeń równa jest ilości linii (opisów lokalizacji) w pliku lokalizacje.txt. Ilość ta obliczana jest w sekcji LINE\_COUNT. W ostatniej sekcji pliku - FORMATOR, wykonywane są wszystkie operacje związane z uporządkowaniem danych wyjściowych. Służy to przejrzystości rozwiązania oraz zapobiega gwałtownemu przyrostowi ilości zapisywanych lokalnie danych.

#### **- LOS.py**

Skrypt w języku Python w którym zapisana jest właściwa sekwencja działań w programach QGIS, GRASS i SAGA. Kod jest tekstowym odpowiednikiem sekwencji wykonywanej manualnie w GUI i opisanej we wcześniejszym rozdziale dotyczącym analizy widoczności w QGIS. Punktem początkowym tworzenia skryptu Python automatyzującego pracę oprogramowania QGIS może być manualne wykonanie w GUI operacji którą chcemy zautomatyzować. Następnie w menu górnym programu QGIS Geoprocessing-Historia możemy sprawdzić kod Python równoważny uruchomionej w GUI funkcjonalności. Zapisując kolejne sekcje kodu do pliku możemy więc łatwo przetestować sekwencyjne uruchomienie kolejnych funkcjonalności QGIS, GRASS i SAGA. Następnym etapem jest uporządkowanie kodu, zwracając uwagę na czytelne usystematyzowanie zmiennych i sposobu ich przekazywania co ma szczególne znaczenie przy sekwencyjnym wykonywaniu skryptu w pętli z warstwy omawianego wcześniej pliku wsadowego (uruchom.bat).



### **- polygonize.bat**

W opracowanym rozwiązaniu wszystkie funkcjonalności programów GRASS oraz SAGA uruchamiane są z warstwy skryptu Python (LOS.py). Natomiast funkcjonalność zamiany rastra na wektor (poligon) wykonywana jest bezpośrednio w QGIS. W tym przypadku wykonanie operacji w GUI QGIS i sprawdzenie menu Geoprocessing-Historia pokaże, że żaden kod Python nie został wygenerowany. Za to wybierając w menu QGIS Raster-Konwersja-Poligonizuj (Raster na wektor)... uruchomi się okienko w dolnej części którego zasugerowany zostanie odpowiadający operacji w GUI kod dla pliku wsadowego. Tak więc, dla uproszczenia struktury plików można użyć równoważnej funkcjonalności programu SAGA uruchamianej bezpośrednio z poziomu LOS.py. Jest to rozwiązanie szybsze i stabilniejsze. Jednak w celach poglądowych zdecydowano się na użycie QGIS z poziomu omawianego pliku wsadowego który uruchamiany jest przez kod Python (LOS.py). Służy to zapoznaniu się z możliwością przekazywania zmiennych w odwrotnym niż dotychczas opisywanym kierunku tj. ze skryptu Python do wsadowego.

### **- py\_start.bat**

Plik wsadowy którego zawartość edytowana jest dynamicznie przez uruchamiający go uruchom.bat dla każdej iteracji pętli w której wykonywany jest skrypt LOS.py. Zapis do omawianego pliku odbywa się w sekcji START pliku uruchom.bat - wykorzystując wbudowane narzędzia linii komend systemu Windows wydzielane są kolejne linie (opisy analizowanych lokalizacji) z pliku lokalizacje.txt. Poszczególne zmienne są następnie wydzielane z tej linii i wpisywane w odpowiedniej sekwencji do omawianego pliku a uruchomienie go powoduje przekazanie zmiennych i uruchomienie skryptu LOS.py.

### **- znacznik\_czasu.bat**

Pomocniczy plik wsadowy którego zadaniem jest wygenerowanie jednoznacznego ciągu dzięki któremu można zidentyfikować uruchomioną analizę, uporządkować jej pliki wynikowe w strukturze dyskowej a także ocenić czas działania rozwiązania. Operując na wbudowanych narzędziach linii komend systemu Windows skrypt generuje ciąg tożsamy z czasem uruchomienia prezentowanego w pracy rozwiązania w formie data-godzina, format RRRRMMDD-GGMM.

### **- lokalizacje.txt**

Plik edytowany przez użytkownika, zawierający opis wszystkich lokalizacji dla których przeprowadzone ma być dane uruchomienie opisywanego w pracy rozwiązania. Każda linia to pojedyncza lokalizacja. Przykładowa linia pliku to

```
1 Zabierzow 50.10916667 19.77944444 40 300
```

gdzie:

- 1 - [dowolna liczba] numer nadany danej lokalizacji,
- Zabierzow - [dowolny ciąg znaków] nazwa lokalizacji - np. nazwa miejscowości lub inny identyfikator,
- 50.10916667 - [szerokość geograficzna lokalizacji w formacie dziesiętnym],
- 19.77944444 - [długość geograficzna lokalizacji w formacie dziesiętnym],
- 40 - wysokość źródła obserwacji (posadowienia anteny lub obserwatora) [mAGL],
- 300 - wysokość celu obserwacji [mAGL] (ang. *Meters Above Ground Level*).

Ze względu na specyfikę oprogramowania GIS wskazane jest unikanie polskich znaków w nazwach lokalizacji. W przypadku współrzędnych dokładność podania lokalizacji jest dowolna ale separatorem dziesiętnym musi być kropka.

Wbudowana w sekcję START pliku uruchom.bat sekwencja formatowania danych wejściowych dla skryptów umożliwia pewną dowolność w formatowaniu symboli rozdzielających poszczególne wartości w pojedynczych liniach omawianego pliku. Może to być, zgodnie ze specyfiką CSV (ang. *Comma-Separated Values*), przecinek albo dowolna liczba spacji. Wybór dogodnego rozwiązania należy do użytkownika. Z technicznego punktu widzenia zawartość pliku sformatowana jak poniżej jest prawidłowa.

```
1,Zabierzow,50.10916667,19.77944444,40,300
```

```
2 Zab    49.3425    19.9675    10,600
```

```
999 Dabrowica  50.50250000, 21.61888889 40 1000
```

Oczywiście wskazane jest zachowanie pewnego ładu co umożliwia wzrokową analizę spójności pliku już na etapie edycji, np.

```
1 Zabierzow 50.10916667 19.77944444 40 1000
```

```
2 Dabrowica 50.50250000 21.61888889 40 1000
```

```
3 Pyrzowice 50.48083333 19.08277778 40 1000
```

```
4 Zab      49.3425      19.9675      10 1000
```

```
5 Olesno   50.21166667  20.93555556 10 1000
```

```
6 Jedrzejow 50.64722222 20.25111111 10 1000
```

Poza wartościami zapisanymi przez użytkownika w pliku w sposób jawny, skrypty rozwiązania operują jeszcze na wartościach wyliczanych z jego zawartości: numerze aktualnie analizowanej linii oraz liczbie linii (lokalizacji) zapisanych w omawianym pliku.

Analizowane przy pomocy prezentowanego rozwiązania lokalizacje muszą znajdować się na terenie Polski. W przeciwnym wypadku konieczne jest użycie pliku NMT.tif dla danego obszaru oraz edytowanie sekcji dotyczącej zasięgu analizy *viewshed* w odpowiedniej linii pliku LOS.py - podane tam współrzędne odpowiadają bowiem skrajnym punktom granic Polski (Jastrzębia Góra, Opołonek, zakole Bugu w Zosinie, zakole Odry w Osinowie Dolnym) tworzące ramę zasięgu analiz.

### \temp:

W toku działania rozwiązania wszystkie pośrednie pliki wynikowe trafiają do omawianego folderu tymczasowego. Kolejne pliki wygenerowane na trzech pierwszych etapach analizy opisanej w rozdziale dot. analizy widoczności w GUI QGIS zakodowane są w następujący sposób:

- **viewshed\_X.\*** to wyniki analizy typu *viewshed* w GRASS,
- **viewshed\_reclass\_X.\*** to wyniki rekasyfikacji SAGA Reclassify values (simple),
- **viewshed\_reclass\_poly\_X.\*** to wyniki wektoryzacji QGIS Raster-Konwersja-Poligonizuj (Raster na wektor)...

gdzie zamiast X występuje numer który jest pierwszym symbolem występującym w analizowanej linii pliku lokalizacje.txt co zapobiega nadpisywaniu wyników kolejnych iteracji pętli. Wszystkie te pliki pośrednie usuwane są z omawianego folderu z chwilą zakończenia działania rozwiązania - przeanalizowania wszystkich linii pliku lokalizacje.txt. Gdyby użytkownik chciał zachować te pliki należy zakomentować lub usunąć odpowiednią linię w sekcji FORMATOR pliku wsadowego uruchom.bat i zachować pliki w lokalizacji innej niż omawiany folder - w przeciwnym wypadku mogą zostać nadpisane kolejnymi uruchomieniami rozwiązania jeżeli użytkownik nada liniom w pliku lokalizacje.txt takie same numery.

### \folder wynikowy (RRRRMMDD-GGMM):

W wyniku uruchomienia prezentowanego rozwiązania tworzony jest folder o nazwie będącej numerem - jednoznacznym identyfikatorem, tego uruchomienia wygenerowanym przez plik wsadowy znacznik\_czasu.bat. Po zakończeniu pracy zapisywane są w nim pliki wynikowe zakodowane w sposób zobrazowany na przykładzie:

## **1\_Zabierzow\_40-1000mAGL.\***

co odpowiada źródłowej linii pliku lokalizacje.txt:

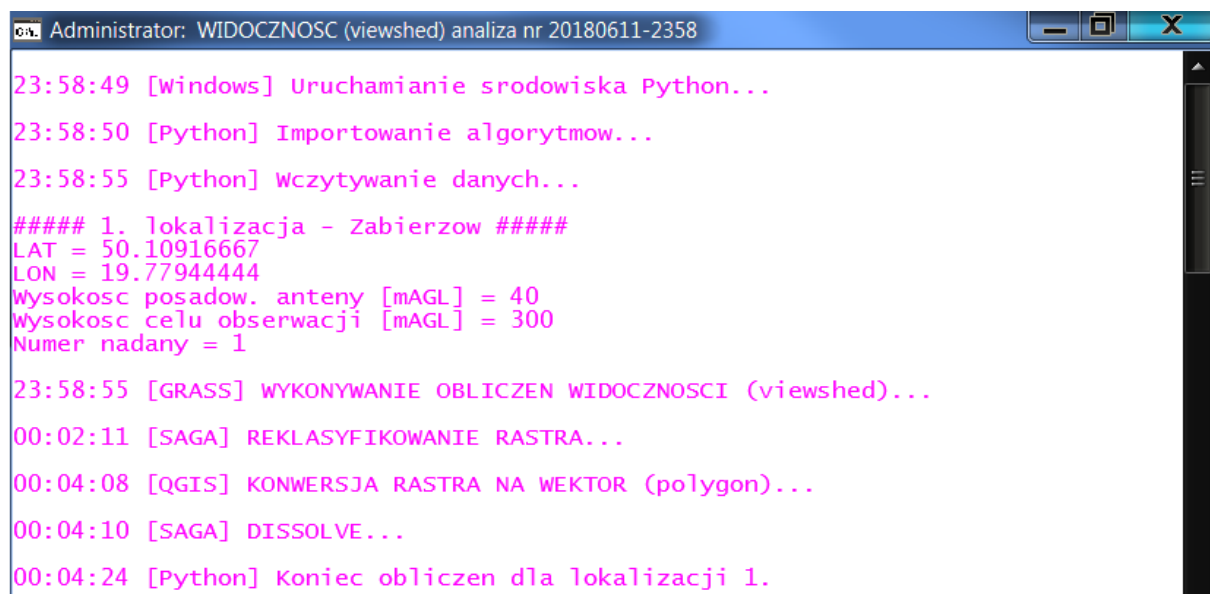
```
1 Zabierzow 50.10916667 19.77944444 40 1000
```

Dodatkowo do omawianej lokalizacji kopiowany jest źródłowy plik lokalizacje.txt ze zmienioną nazwą do której dodawany jest znacznik uruchomienia (lokalizacje\_RRRRMMDD-GGMM.txt). Dodanie znacznika umożliwia ewentualne odróżnienie dwóch otwartych w systemie Windows okien: już przeanalizowanego zestawienia lokalizacji z nową, aktualnie edytowaną wersją w folderze C:\QGIS\skrypty.

Bezpośrednio w folderze C:\QGIS znajdują się skróty do dwóch kluczowych z punktu widzenia użytkownika plików. Najprostszą formą stosowania omawianego rozwiązania jest bowiem wyedytowanie według potrzeb użytkownika, zgodnie z omówionymi wcześniej zasadami, pliku lokalizacje.txt a następnie uruchomienie pliku wsadowego uruchom.bat. Cała opisana w rozdziale struktura plików znajduje się na załączonym do pracy nośniku cyfrowym. Użytkownik może z niej skorzystać po skonfigurowaniu kompatybilnego z opisanym środowiska pracy (QGIS 2.18) i skopiowaniu opracowanego rozwiązania do właściwej lokalizacji dyskowej (C:\QGIS\).

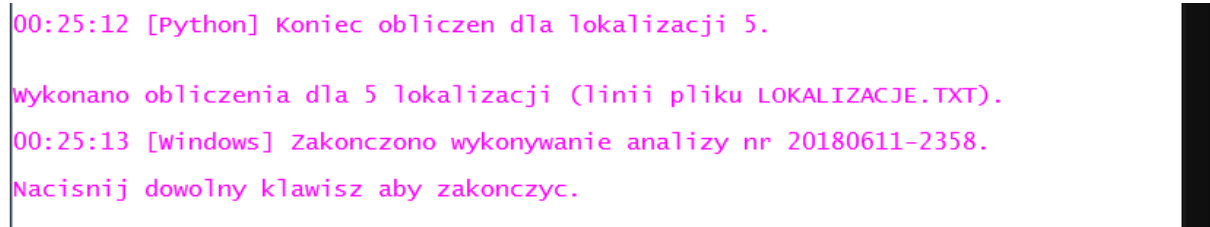
### **4.2. Uruchomienie narzędzia**

Przykładowe uruchomienie sekwencji skryptów dla pliku wejściowego zawierające opis pięciu lokalizacji przedstawiono poniżej. Opracowanie kartograficzne tego uruchomienia stanowi załącznik do pracy. Na zrzutach ekranu konsoli skryptu przedstawiono blok komunikatów typowych dla przetwarzania pojedynczej linii pliku wejściowego lokalizacje.txt. Dobrze widoczne są znaczniki czasu pozwalające oszacować czas działania narzędzia.



```
Administrator: WIDOCZNOSC (viewshed) analiza nr 20180611-2358
23:58:49 [Windows] Uruchamianie srodowiska Python...
23:58:50 [Python] Importowanie algorytmow...
23:58:55 [Python] Wczytywanie danych...
##### 1. lokalizacja - Zabierzow #####
LAT = 50.10916667
LON = 19.77944444
Wysokosc posadow. anteny [mAGL] = 40
Wysokosc celu obserwacji [mAGL] = 300
Numer nadany = 1
23:58:55 [GRASS] WYKONYWANIE OBLICZEN WIDOCZNOSCI (viewshed)...
00:02:11 [SAGA] REKLASYFIKOWANIE RASTRA...
00:04:08 [QGIS] KONWERSJA RASTRA NA WEKTOR (polygon)...
00:04:10 [SAGA] DISSOLVE...
00:04:24 [Python] Koniec obliczen dla lokalizacji 1.
```

Rys. 13. Okno konsoli po uruchomieniu narzędzia.



```
00:25:12 [Python] Koniec obliczen dla lokalizacji 5.
Wykonano obliczenia dla 5 lokalizacji (linii pliku LOKALIZACJE.TXT).
00:25:13 [Windows] Zakonczono wykonywanie analizy nr 20180611-2358.
Nacisnij dowolny klawisz aby zakonczyc.
```

Rys. 14. Okno konsoli po zakończeniu pracy narzędzia.

### 4.3. Wnioski

Podczas programowania natrafiono na dwa ciekawe problemy.

1. Iteracyjne wykonywanie skryptu Python LOS.py w pętli pliku wsadowego uruchom.bat wydaje się być definitywnym zastosowaniem pętli FOR jak w poniższym fragmencie kodu.

```
:START  
  
:: uruchom Python-skrypt dla kazdej linii pliku wej.  
  
for /f "tokens=*" %%a in (lokalizacje.txt) do (  
  
    SET /a "licznik_linii+=1"  
  
    python LOS.py %%a  
  
    ECHO.  
  
)
```

Okazuje się jednak, że z powodu specyfiki implementacji pętli FOR w plikach wsadowych linii komend systemu MS Windows wartości zmiennych modyfikowanych w pętli (np. w omawianym przykładzie licznik\_linii) nie można przesłać dynamicznie poza nią. Utrudnia to np. wykorzystanie wartości zmiennej do oznaczania plików z pośrednimi wynikami działania skryptu, chyba że całe sekcje kodu umieszczone zostaną wewnątrz pętli co niekorzystnie wpływa na stabilność działania rozwiązania. Konieczne jest zatem przesyłanie wartości przez zapis do pliku co jest rozwiązaniem powolnym. W końcowej wersji rozwiązania opisywana konieczność - iteracyjna modyfikacja py\_start.bat, została wykorzystana. Zawartość pliku jest bowiem na tyle czytelna, że w przypadku awaryjnego zatrzymania się normalnej pracy rozwiązania, zapisaną w pliku linię łatwo ocenić wzrokowo pod kątem poprawności danych zapisanych w pliku wejściowym lokalizacje.txt.

2. Sekwencyjne wykonywanie operacji oprogramowania GIS w skrypcie Python LOS.py oznacza, że pliki wyjściowe danego etapu analizy stają się plikami wejściowymi następnego. Może to prowadzić do sytuacji w której, w zależności od osiągniętych sprzętowych konkretnego komputera na którym prowadzone są obliczenia, czas zapisu plików na dysku jest na tyle długi, iż pliki nie zostaną podane następnemu etapowi analizy dostatecznie szybko. Na komputerze testowym najbardziej wymagającym pod tym względem okazał się styk operacji QGIS-poligonizuj i SAGA-Polygon dissolve. Problem rozwiązano programując, w sposób statyczny, dwusekundową zwłokę przed rozpoczęciem wykonywania drugiej operacji. Jako kompromis pomiędzy szybkością a stabilnością działania, dwusekundową zwłokę zastosowano następnie przed wszystkimi wykonywanymi sekwencyjnie etapami analizy.

## 5. Wnioski końcowe i podsumowanie

W wyniku prac udało się uzyskać narzędzie do automatyzacji analiz widoczności o dobrej stabilności i zadowalającej ergonomii użytkownika. Do minimum ograniczona została konieczność interakcji użytkownika z wymagającym oprogramowaniem. Czas działania narzędzia jest jedynie nieznacznie większy od czasu samych obliczeń. Wykonywane przez skrypty operacje nie wymagają uruchamiania GUI QGIS co obniża ilość potrzebnych zasobów komputera na którym prowadzone są obliczenia.

Rozwiązanie ma potencjał rozwojowy do dalszych analiz prowadzonych na wyjściowych plikach typu *ESRI shapefile*, przykładowo obliczanie pola powierzchni iloczynu poligonów widoczności. Takie obliczenie (lub inne kryteria) może być następnym wykonywanym iteracyjnie etapem działania co z prezentowanego rozwiązania uczyniłoby już nie tylko narzędzie automatyzacji analiz lecz także optymalizacji doboru lokalizacji.



## 6. Słownik skrótów

CODGIK – Centralny Ośrodek Dokumentacji Geodezyjnej i Kartograficznej

CSV – (ang.) *Comma-Separated Values*

GIS – (ang.) *Geographic Information System(s)*

GUI – (ang.) *Graphic User Interface*

KRL – Kontrola Ruchu Lotniczego

LOS – (ang.) *Line Of Sight*

LTR – (ang.) *Long Term Release*

mAGL – (ang.) *Meters Above Ground Level*

MB – megabajt

NMT – Numeryczny Model Terenu

SRTM – (ang.) *Shuttle Radar Topography Mission*

## 7. Bibliografia

### Witryny internetowe:

- <http://urbnews.pl/>, maj 2018
- <http://gis-support.pl/>, czerwiec 2018
- <https://doc.arcgis.com/pl/>, czerwiec 2018
- <https://docs.qgis.org/2.18/en/docs/>, czerwiec 2018

## **8. Załączniki**

Do pracy załączone są:

- listingi skryptów,
- opracowanie kartograficzne analiz wg. opisu w pracy,
- nośnik optyczny z pracą w wersji cyfrowej.