

Przetwarzanie obrazów

Zajęcia 13

Zastosowanie SML do Panshapingu.

Zasady wykonania ćwiczenia

- Obrazy wynikowe do zadań zapisujemy w pliku nazwiskonr.rvc (bieżące nr 1) a komentarze do wyników zapisujemy w pliku NazwiskoNR.doc, według wskazówek schematów zawartych w ćwiczeniu
- **NIE UMIESZCZAĆ SPACJI I POLSKICH ZNAKÓW W NAZWACH PLIKÓW (geoinfo posiada w2k)**
- Po zajęciach ładujemy pliki z wynikami i komentarzem na serwer przez stronę <http://www.geoinfo.amu.edu.pl/geoinf>
- 1) Logujemy się, 2) wybieramy plik do upload'u, 3) Ładujemy,
- 4) powtarzamy tę czynność odpowiednią ilość razy 5) sprawdzamy zawartość katalogu ćwiczenia nr....*
- **Jeżeli ktoś jest nieobecny na ćwiczeniach to wykonuje ćwiczenie samodzielnie w domu. Opisy do ćwiczeń dostępne są w Internecie pod adresem <http://ztg.amu.edu.pl/zajecia.htm>, dalej należy wybrać odpowiedni przedmiot**
- * Jak ten sposób nie będzie działał, to wówczas to proszę wysłać pocztą (lotniczą ☺) na adres: **skrol@amu.edu.pl**

Zadanie

- Zadanie polega na napisaniu skryptu SML, który będzie stanowił implementację jednej z przedstawionych na poprzednich zajęciach metod Pansharpeningu. W skrypcie trzeba się wykazać własnymi pomysłami, np. na modyfikację metod panshaping'u, na oryginalne rozwiązanie struktury skryptu;
- Skrypt powinien być napisany przejrzysto, z wyczerpującymi objaśnieniami, co w danym kroku jest wykonywane.
- Przy modyfikowaniu metod Pansharpening'u pomocna będzie analiza histogramów obrazów składowych zawartych w pliku ikonos.rvc.
- Podsumowaniem powinien być „pamiętnik” rozwoju skryptu z ilustracjami osiągniętych wyników...
- Do ćwiczeń jest dołączony przykładowy skrypt SML pansharp.sml, opisane są jego podstawowe składowe, takie jak otwieranie obiektów rastrowych, obliczenia zapisywanie i oglądanie wyników operacji Panshaping'u
- Pewnych wskazówek, „natchnienia” - jak można rozwiązać różne problemy związane ze składnią znajdziecie państwo również w skrypcie water.sml, który ma inny cel niż pansharpening, ale znajdują się takie elementy jak rozciąganie histogramu metodą wykładniczą itp.

TNTmips 2006.72 Serial# 7312

Main Raster Geometric Convert Script Tools Help

Display Manager

Display Add Options Help

Display Group 1

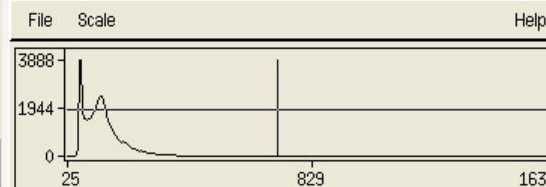
- ☒ ikonos / po_3604966_red_0000000
- ☒ ikonos / po_3604966_pan_0000000
- ☒ ikonos / po_3604966_nir_0000000
- ☒ ikonos / po_3604966_grn_0000000
- ☒ ikonos / po_3604966_blu_0000000

Display Group 1 - View 1

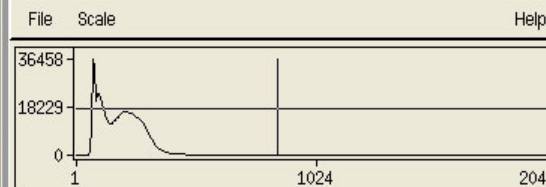
View Tool LegendView GPS Options HotKeys

Help

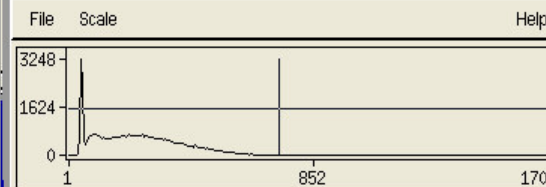
Raster Histogram



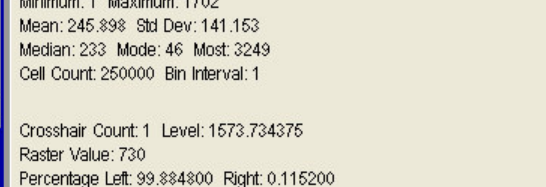
Raster Histogram



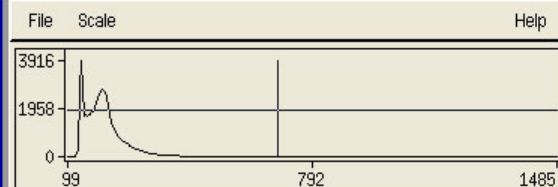
Raster Histogram



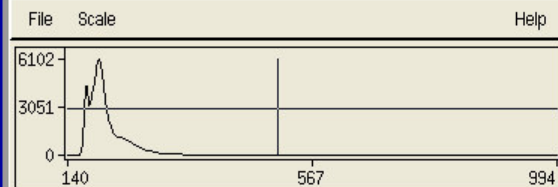
Raster Histogram



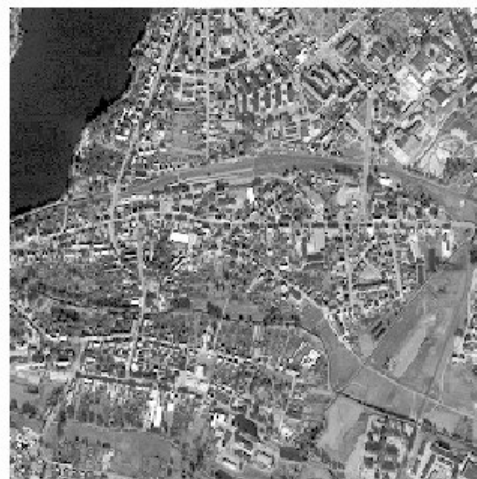
Raster Histogram



Raster Histogram



Crosshair Count: 3 Level: 2955.656250
Raster Value: 506
Percentage Left: 99.843200 Right: 0.156800



353136.56 N m E 19 00 00.000 N 90 00 00.000

Insert Function

Function Group... Raster

AreaCorrelatePoint - Adjust point position of Raster to match known position in reference Raster
 Binarization - Calculates a raster binarization on an 8 or 4 bit raster
 CloseRaster - Close an open raster
 ColScale - Get horizontal computed size of raster cell
 ComputeMeanRaster - The output raster contains the mean value (at each cell) of all of the inputs.
 ComputeRasterProperties - Compute Raster attribute tables from a Vector object
 ComputeReliefShading - Compute relief shading of the input raster
 ComputeSlopeAspectShading - Compute slope/aspect/shading from elevation Raster
 ComputeStdDevRaster - The standard deviation (at each cell) of all of the inputs is assigned to the output raster.
 ConvertCompToComp - Converts the color depth of a raster to another color depth.
 ConvertRGBToComposite - Converts and RGB raster to an 8 or 4 bit composite
 CreateHistogram - Create a histogram for a Raster
 CreatePyramid - Create a pyramid for the given raster object
 CreateRaster - Create a new raster
CreateRasterBinaryMask - Create a binary mask raster
 CreateRasterFromObject - Create a Raster to match a given Object.
 CreateRasterMask - Create a mask raster (for use as a transparency mask)
 CreateTempRaster - Create a raster in RAM
 DeleteContrast - Delete contrast subobjects of the given raster object
 DeleteHistogram - Delete histogram subobject of the given raster object
 DeletePyramid - Delete pyramid subobject of the given raster object
 DeleteTempRaster - Delete a raster that has been created with CreateTempRaster()
 GetInputRaster - Use a File/Object selection dialog to select an input Raster
 GetInputRasters - Get more than one raster.

CreateRasterBinaryMask

Create a binary mask raster

CreateRasterBinaryMask(Raster, filename\$, objname\$, MaskValue)

Where:

Raster

Parameters:

Raster Raster
 Input raster
 string filename\$
 Output raster filename
 string objname\$

Close

Insert

Details...

pansharp.sml - SML Editor

File Edit Insert Syntax Help

```
clear();
# deklaracja zmiennych
raster R, G, B, Ir, Pan;
raster Rc, Gc, Bc, Irc;
raster Rres, Gres, Bres, Ires;
raster Rpan, Gpan, Bpan, Pancor, W;
# otwarcie rastrow wejsciowych
GetInputRasters(R,G,B,Ir,Pan);
# odczytanie ilosci wierszy, kolumn i typu rastu panchromatycznego
numeric lin=NumLins(Pan);
numeric col=NumCols(Pan);
string type$=RastType(Pan);
# utworzenie rastrow tymczasowych - spektralnych dopasowanych do rastu Panchromatycznego
CreateTempRaster(Rres,lin,col,type$);
CreateTempRaster(Gres,lin,col,type$);
CreateTempRaster(Bres,lin,col,type$);
CreateTempRaster(Ires,lin,col,type$);
ResampleRasterToMatch(R,Pan,Rres,"affine","cubic");
ResampleRasterToMatch(G,Pan,Gres,"affine","cubic");
ResampleRasterToMatch(B,Pan,Bres,"affine","cubic");
ResampleRasterToMatch(Ir,Pan,Ires,"affine","cubic");
# filtr usredniajacy majacy na celu eliminacje wartosci skrajnych, poprawe ksztatu histogramu
CreateTempRaster(Rresf,lin,col,type$);
CreateTempRaster(Gresf,lin,col,type$);
CreateTempRaster(Bresf,lin,col,type$);
CreateTempRaster(Iresf,lin,col,type$);
CreateTempRaster(Panf,lin,col,type$);
Rresf=FocalMedian(Rres,3,3);
Gresf=FocalMedian(Gres,3,3);
Bresf=FocalMedian(Bres,3,3);
Iresf=FocalMedian(Ires,3,3);
Panf=FocalMedian(Pan,3,3);
## Obliczenia zwiazane z Pansharp ESRI
#numeric r, g, b, i; #wskazniki wagowe, korekcja Pan przez Ir
##=0.166; b= 0.167; g=0.167; i=0.500;
GetOutputRaster(Rpan,lin,col,type$);
GetOutputRaster(Gpan,lin,col,type$);
GetOutputRaster(Bpan,lin,col,type$);
GetOutputRaster(Pancor,lin,col,"32-bit float");
# korekcja kanalu panchromatycznego
Pancor=(Panf)/(Rresf+Gresf+Bresf);
# korekcja kanalow spektralnych
```

Run...

Cancel