Subject: How to extract pixel values from a GeoTIFF using an Esri Shapefile
Posted by Paul Magdon on Thu, 23 Dec 2010 20:53:55 GMT
View Forum Message <> Reply to Message

Dear all,
currently I am developing an pure IDL (No ENVI functions) based remote
sensing application which involves supervised classification.
Therefore I need to import a Geotiff with the raster data plus an ESRI
shapefile (Polygons) with the trainings-data. I then converted the
shape file to an IDLanROI object and build a mask using
IDLanROI::ComputeMask method. At this point I got stuck as I don't
know how spatially link both image and roi even when they are in UTM
projection.

Here is my code so far:
;Import Multispectral Geo TIFF
 img=READ_TIFF('some_multispectral.tiff',GEOTIFF=geokeys,inte rleave=2,)
s= SIZE (img, /DIMENSIONS)

;Setup Map projection based on GeoTIFF tags (UTM 17N, Wgs84)
mapCoord = GeoCoord(img(*,*,1),geokeys)
mapStruct = mapCoord -> GetMapStructure()

;import shapfile (aggain UTM 17N, Wgs84)
myshape=OBJ_NEW('IDLffShape', 'some_esri_multipolygone.shp')

;Get the first polygon
ent=myshape->IDLffShape::GetEntity(1)

;Convert SHP to ROI object
myroi = OBJ_NEW('IDLanROI',(*ent.vertices)[0,*] ,(*ent.vertices)
[1,*])

;Apply ComputMask method
maskResult = myroi -> ComputeMask(DIMENSIONS = [s[0],
s[1]],MASK_RULE=2)

; Here starts the problem; I have a 5000x5000 raster (img) and the
image coordinate system start with [0,0] but the myroi coordinates are
much higher (eg 227984.00 ; 991472.00) Thus everything is masked.
;There is a keyword LOCATION for ComputeMask but I am not sure how to
use it and where to get the right values for it

index= WHERE(maskResult gt 0)
subset=img(index)

OBJ_DESTROY, myshape
OBJ_DESTROY, myroi

Subject: Re: How to extract pixel values from a GeoTIFF using an Esri Shapefile
Posted by David Fanning on Mon, 03 Jan 2011 13:55:56 GMT
View Forum Message <> Reply to Message

Paul Magdon writes:

> did you received my mail with the test dataset? I don't want to
> stress you it's just that I am not sure if I used the right E-mail
> adress :)

Yes, sorry, I received it. Just pretty busy right now.
I'll get to it. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: How to extract pixel values from a GeoTIFF using an Esri Shapefile
Posted by guillermo.castilla.ca on Mon, 03 Jan 2011 22:46:24 GMT
View Forum Message <> Reply to Message

Hi Paul,

If what you want is a mask where the pixels within each polygon have
as DN its ID, there is a simpler way which doesn't require the
IDLanROI class:

;Import GeoTIFF and create mask array
 img=READ_TIFF('some_multispectral.tif',GEOTIFF=geokeys,inter leave=2)
psz= mapinfo.ModelPixelSCALETAG[0] ; pixel size
x0= geokeys.ModelTiePointTag[3] - geokeys.ModelTiePointTag[0]*psz
y0= geokeys.ModelTiePointTag[4] + geokeys.ModelTiePointTag[1]*psz
; NB, x0 and y0 are respectively the easting and northing
; of the NW corner of the image
s= SIZE(img, /DIMENSIONS)
mask= LONARR(s)

;import shapefile
myshape= OBJ_NEW('IDLffShape', 'some_esri_multipolygone.shp')

; populate the mask (assumes is in the same projection
;and covers the same extent as the geotiff)
myshape->IDLffShape::GetProperty, N_ENTITIES=n
FOR i=0L, n-1 DO BEGIN
  feati= myshape->IDLffShape::GetEntity(i)
  featix= Round((Reform((*feati.vertices)[0,*])-x0)/psz)
  featiy= Round((y0-Reform((*feati.vertices)[1,*]))/psz)
  featis= POLYFILLV(featix, featiy, ns, nl)
  IF featis[0] NE -1 THEN mask[featis]= feati.ishape +1
ENDFOR

If instead of the polygon ID you want to store in the mask the
thematic class to which the polygon belongs, you just add attr=iShp-
> IDLffShape::GetAttributes(i) in the beginning of the loop, replace in
the last line of the loop with mask[featis]= attr.(j) (where j is the
position in the dbf table of the field that contains the polygon
class).

Cheers

Guillermo

Subject: Re: How to extract pixel values from a GeoTIFF using an Esri Shapefile
Posted by David Fanning on Tue, 04 Jan 2011 00:25:55 GMT
View Forum Message <> Reply to Message

Guillermo writes:

> ; populate the mask (assumes is in the same projection
> ;and covers the same extent as the geotiff)
> myshape->IDLffShape::GetProperty, N_ENTITIES=n
> FOR i=0L, n-1 DO BEGIN
>   feati= myshape->IDLffShape::GetEntity(i)
>   featix= Round((Reform((*feati.vertices)[0,*])-x0)/psz)
>   featiy= Round((y0-Reform((*feati.vertices)[1,*]))/psz)
>   featis= POLYFILLV(featix, featiy, ns, nl)
>   IF featis[0] NE -1 THEN mask[featis]= feati.ishape +1
> ENDFOR

This is nearly identical to the solution I sent Paul
earlier today. The problem with IDLanROI is that it
can't be used in the native projected meter coordinates
that the image and shape files are in. The projected
meter coordinates have to be "converted" to "pixel"
coordinates by subtracting the offset and dividing
by the image range, before they can be loaded in the
object.

I didn't round my values, and I don't think you need
to do so here, even with PolyFillV. In fact, I think
you might get slightly more accurate values by not
rounding, although this is a quibble with Paul's
image.

I've made myself a note to write an article when
I get some time. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: How to extract pixel values from a GeoTIFF using an Esri Shapefile
Posted by Adam Erickson on Wed, 19 Aug 2015 21:57:36 GMT

On Monday, January 3, 2011 at 4:25:55 PM UTC-8, David Fanning wrote:
> Guillermo writes:
>
>> ; populate the mask (assumes is in the same projection
>> ;and covers the same extent as the geotiff)
>> myshape->IDLffShape::GetProperty, N_ENTITIES=n
>> FOR i=0L, n-1 DO BEGIN
>>   feati= myshape->IDLffShape::GetEntity(i)
>>   featix= Round((Reform((*feati.vertices)[0,*])-x0)/psz)
>>   featiy= Round((y0-Reform((*feati.vertices)[1,*]))/psz)
>>   featis= POLYFILLV(featix, featiy, ns, nl)
>>   IF featis[0] NE -1 THEN mask[featis]= feati.ishape +1
>> ENDFOR
>
> This is nearly identical to the solution I sent Paul
> earlier today. The problem with IDLanROI is that it
> can't be used in the native projected meter coordinates
> that the image and shape files are in. The projected
> meter coordinates have to be "converted" to "pixel"
> coordinates by subtracting the offset and dividing
> by the image range, before they can be loaded in the
> object.

>
> I didn't round my values, and I don't think you need
> to do so here, even with PolyFillV. In fact, I think
> you might get slightly more accurate values by not
> rounding, although this is a quibble with Paul's
> image.
>
> I've made myself a note to write an article when
> I get some time. :-)
>
> Cheers,
>
> David
>
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Honestly, use QGIS for this. It's 64-bit, unlike ArcGIS, and its GDAL functions outperform everything else I've tried, aside from my own GDAL python code. Simply add the Zonal Statistics tool in Plugins -> Manage and Install Plugins... then select Zonal statistics under the Raster menu. The operation computes in one minute on very large mosaics with many polygons and writes the values to new fields in the shapefile attribute table - something I could not get ENVI to do.

ArcGIS ran for three days before crashing (one bad polygon is all it took), with the raster placed in a File Geodatabase to mitigate its 32-bit limitations. ENVI took a day to compute this, plus one hour simply to draw the polygons to screen, while outputting the data in what appears to be a useless format. Originally, I was doing this operation with IDL similar to the above methods, but my mosaic is much larger than my 96 GB of RAM. My web browser uses more RAM than QGIS, which draws to screen very quickly as well. I cannot recommend it enough for zonal statistics.

Cheers,

Adam Erickson
Integrated Remote Sensing Studio
UBC

Subject: Re: How to extract pixel values from a GeoTIFF using an Esri Shapefile
Posted by ittai9 on Fri, 29 Apr 2016 13:36:31 GMT
View Forum Message <> Reply to Message

Hello,

Was it successful to convert shpfiles to ROIs of ENVI?

Is there a way to do it?

Best,

On Thursday, December 23, 2010 at 2:53:55 PM UTC-6, PMagdon wrote:
> Dear all,
> currently I am developing an pure IDL (No ENVI functions) based remote
> sensing application which involves supervised classification.
> Therefore I need to import a Geotiff with the raster data plus an ESRI
> shapefile (Polygons) with the trainings-data. I then converted the
> shape file to an IDLanROI object and build a mask using
> IDLanROI::ComputeMask method. At this point I got stuck as I don't
> know how spatially link both image and roi even when they are in UTM
> projection.
>
> Here is my code so far:
> ;Import Multispectral Geo TIFF
> img=READ_TIFF('some_multispectral.tiff',GEOTIFF=geokeys,inte rleave=2,)
> s= SIZE (img, /DIMENSIONS)
>
> ;Setup Map projection based on GeoTIFF tags (UTM 17N, Wgs84)
> mapCoord = GeoCoord(img(*,*,1),geokeys)
> mapStruct = mapCoord -> GetMapStructure()
>
> ;import shapfile (aggain UTM 17N, Wgs84)
> myshape=OBJ_NEW('IDLffShape', 'some_esri_multipolygone.shp')
>
> ;Get the first polygon
> ent=myshape->IDLffShape::GetEntity(1)
>
> ;Convert SHP to ROI object
> myroi = OBJ_NEW('IDLanROI',(*ent.vertices)[0,*] ,(*ent.vertices)
> [1,*])
>
> ;Apply ComputMask method
> maskResult = myroi -> ComputeMask(DIMENSIONS = [s[0],
> s[1]],MASK_RULE=2)
>
> ; Here starts the problem; I have a 5000x5000 raster (img) and the
> image coordinate system start with [0,0] but the myroi coordinates are
> much higher (eg 227984.00 ; 991472.00) Thus everything is masked.
> ;There is a keyword LOCATION for ComputeMask but I am not sure how to
> use it and where to get the right values for it
>
> index= WHERE(maskResult gt 0)
> subset=img(index)
>

> OBJ_DESTROY, myshape
> OBJ_DESTROY, myroi