Subject: Re: aggregate classified image to create pseudo-fraction images
Posted by Allard de Wit on Fri, 06 Dec 2002 10:47:03 GMT
View Forum Message <> Reply to Message

Hi Casey,

this is the kind of problem that could be much easier carried out
in a GIS software package then in IDL, but I'll give it a try.

The basic point is that you will need to generate an image with zones
(blocks) that all have a unique ID and correspond with the pixelsize of
your Landsat TM image. THis code should do the trick although there are
probably more efficient ways to generate such an image in IDL, where
x_aggr and y_aggr are the aggregation factors in x and y directions:

```
 ------------------------------------------------------------ ---
function unique_zones, image, x_aggr, y_aggr

r=size(image, /structure)
xsize=r.dimensions[0]
ysize=r.dimensions[1]
unique_zones=lonarr(xsize,ysize)

for x=0,xsize-1 do begin
   for y=0,ysize-1 do begin
     unique_zones[x,y]= floor(y/float(y_aggr))* $
       ceil(xsize/float(x_aggr)) + $
       floor(x/float(x_aggr))
   endfor
endfor

return, unique_zones

end
 ------------------------------------------------------------ --
```

Next we need to generate the image statistics for each unique
zone. In fact we are calculating the histogram for each block with
a unique ID:

```
 ------------------------------------------------------------ --
function summarize, image, unique_zones

r=size(image, /structure)
xsize=r.dimensions[0]
ysize=r.dimensions[1]
nr_classes=max(image)
max_zone=max(unique_zones)
```

```
summary_table=lonarr(nr_classes+1, max_zone+1)

for x=0,xsize-1 do begin
  for y=0,ysize-1 do begin
    class=image[x,y]
    unique_id=unique_zones[x,y]
    summary_table[class, unique_id]=summary_table[class, unique_id]+1
  endfor
endfor

return, summary_table

end
 ---------------------------------------------------------- --
```

The last step is to use the summary table as a lookup table in order to calculate the fraction of each class for each zone and insert that value back into the image using the image with unique zones again. This function will return an array the same size as your image and the nr of classes. The index in the third dimension corresponds with your class number and contains the fraction for each class:

```
 ---------------------------------------------------------- ---
function lookup, unique_zones, summary_table

r=size(unique_zones, /structure)
xsize=r.dimensions[0]
ysize=r.dimensions[1]
r=size(summary_table, /structure)
nr_classes=r.dimensions[0]
frac_image=fltarr(xsize, ysize,nr_classes)

for x=0,xsize-1 do begin
  for y=0,ysize-1 do begin
    unique_id=unique_zones[x,y]
    pixelcount=float(total(summary_table[*,unique_id]))
    for c=0, nr_classes-1 do begin
      class_frac=summary_table[c, unique_id]/pixelcount
      frac_image[x,y,c]=class_frac
    endfor
  endfor
endfor

return, frac_image

end
 ---------------------------------------------------------- ---
```

Basically this will work, in fact it was less code then I expeced.
The only thing left is a comment on your methodology.

Your are assuming that what the Landsat TM sensor 'sees' corresponds exactly with a 30x30 meter block of your IKONOS image. This is not entirely true. In fact due to the optics of the sensor the total radiance that is collected corresponds to a circular area (or ellipsoid area in off-nadir viewing direction) with a gaussian distribution.
This means that the area in the center of the circular area is contributing more to the measured radiance then the areas on the outer rim of the circel. this is the so-called "point-spread function".
If you really want to "mimic" the behaviour of landsat TM sensor the best way is to turn your classificed IKONOS image into binary layers for each class with 1/0 values for the occurence or absense of a class at each location. Next, you convolve the binary layers with a normalised circular filter (with the size of a Landsat TM pixel) with a gaussian distribution. Using this approach you will get a much more realistic distribution of your fraction classes compared to simply calculating the fractions per block of 30x30 pixels

Hope this helps in getting your problem solved

best regards,

Allard de Wit

cody wrote:

> Hello all,
> I have a high spatial resolution classified air photo that I am trying to use tovalidate a linear spectral unmixing I ran on a coarser spatial resolution image (Landsat TM). Essentially, I am attempting to aggregate the high spatial resolution classified image into a pseudo fraction image. For example, I have five
> classes in the high spatial res classified image. I would like to aggregate
> each 30 by 30 pixel block and create a five band output image where each band
> corresponds to the percent cover of each of the input image classes for the
> aggregated pixels in the output image.
>
> Thanks,
> Casey Cody