
Subject: speed-up computation of kernel-based "statistics"

Posted by [lbusett](#) on Wed, 18 Apr 2012 09:28:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi all,

I have two large (20000*20000) images. The first one is a classification, with discrete values from 1 to 10, while the second one contains values of a variable of interest. For each pixel, I have to compute the 5th and the 95th percentile of the values of the variable in a 801*801 window (centered on the selected pixel), for the pixels of the same class of the center pixel.

What I'm doing now is shown below, but due to the size of the inputs it takes forever to complete. Does anyone have suggestions for improving the speed of the processing ?

In practice, what I'm doing now is:

- 1)for each row of the input data, load the data of the 801 rows centered on the selected one (20000*801 matrix)
- 2)start cycling on the columns. For each column, extract the correct 801*801 "kernel" window, find which pixels are "classified" as the central one, and compute the statistics.
- 3) Save results line-by-line using writeu

; Example Code

; Open output files. Will be filled "line by line"

```
openw, unit_5, out_file_min, /get_lun, width = 25000
openw, unit_95, out_file_max, /get_lun, width = 25000
```

```
; Start cycling on rows of the input
for row = 399,n_row-402 DO begin
```

```
    ; Get data of the 801 rows centered on selected row (I'm using
    envi_get_data, here but I could use something else)
    dims = [-1L,0, ns-1,row-399,row+401]
    var_row_data = ENVI_GET_DATA(dims = dims, fid = DVI_fid, pos =
[0])
    class_row_data = ENVI_GET_DATA(dims = dims, fid = CLC00_fid, pos =
[0])
```

```
; initialize "temporary" output variables.
```

```
    temp_out_5th = intarr(ns)
    temp_out_95th = intarr(ns)
```

```
    for col = 399L, ns-402 do begin        ; cycle on columns
```

```

        pix_class = class_row_data[col,
400]           ; get "pixel of interest" class
        var_data = var_row_data[(col-399):(col+401),*] ;
extract kernel on "variable" image
        class_data = class_row_data[(col-399):(col
+401),*] ; extract kernel on classification image
        var_data_class = var_data[where(class_data EQ
pix_class)] ; get data from pixels classified like the center

        ; Compute percentiles and set the values in the
output arrays
        sort_var= sort(var_data_class )
        ndata = n_elements(sort_var)
        temp_out_5th[col] = var_data_class[sort_var[(5*ndata/
100)]]
        temp_out_95th[col] =
var_data_class[sort_var[(95*ndata/100)]]
        endfor

; Write the "rows" of the outputs
writeu, unit_5, temp_out_5th
writeu, unit_95, temp_out_95th

endfor

```

Thanks in advance for any help,

Lorenzo
