Subject: Re: correlation of single pixels
Posted by Klemen on Mon, 05 Nov 2012 12:59:18 GMT
View Forum Message <> Reply to Message

Hi Max, check the following posts, maybe something useful for you.

Cheers, Klemen

- stereo triangulation in IDL
https://groups.google.com/forum/?hl=en&fromgroups=#!sear
chin/comp.lang.idl-pvwave/stereo$20triangulation$20in$20IDL/
comp.lang.idl-pvwave/7q8b1bClU8g/mUeQVvZlq_sJ

- image matching / allign images
https://groups.google.com/forum/?hl=en&fromgroups=#!sear
chin/comp.lang.idl-pvwave/image$20matching$20$2F$20allign$20
images/comp.lang.idl-pvwave/Ui2M71i4eoo/9vDKb20sMXcJ

- Coadd images that contain no stars
https://groups.google.com/forum/#!topic/comp.lang.idl-pvwave /pVX8wDFLwHI/discussion

Subject: Re: correlation of single pixels
Posted by haikoley on Fri, 09 Nov 2012 07:53:26 GMT
View Forum Message <> Reply to Message

Thanks Klemen,

but somehow it didn't really help. I figured I have to use a template consisting of my pixel and the neighboring pixelsto find the offset. But something's wrong.

I tried:
CONV= where(max(CONCOVAR(image1, image2, /Correl)))
where concovar is a function returning the correlationmatrix of the images, but somehow all my offsets are 0. That can't be (I checked on the images and there must be an offset around 1 or 2 pixels).

On the other hand I tried using correl_optimize (I found it on the internet), but those results are even weirder. My template is an array of 5x3 elements and the other image is an array of 11x3 elements. correl_optimize returns a y-offset between -2,5 and 2,5 pixels. But how can the y-offset possibly be in a range of 5 pixels when both images only consist of three pixels in y direction? Same with x direction.

I don't have a clue as to what I do wrong. Has someone an explanation for this?

cheers,
Max

Subject: Re: correlation of single pixels
Posted by Brian Daniel on Fri, 09 Nov 2012 12:35:10 GMT
View Forum Message <> Reply to Message

This sounds like an Optical Flow problem. Do a google scholar search. I know there are a bunch of freeware matlab code packages on the web. I haven't seen OF packages in IDL. =(

-Brian

On Friday, November 9, 2012 2:53:26 AM UTC-5, haik...@gmail.com wrote:
> Thanks Klemen,
>
>
>
> but somehow it didn't really help. I figured I have to use a template consisting of my pixel and the neighboring pixelsto find the offset. But something's wrong.
>
>
>
> I tried:
>
> CONV= where(max(CONCOVAR(image1, image2, /Correl)))
>
> where concovar is a function returning the correlationmatrix of the images, but somehow all my offsets are 0. That can't be (I checked on the images and there must be an offset around 1 or 2 pixels).
>
>
>
> On the other hand I tried using correl_optimize (I found it on the internet), but those results are even weirder. My template is an array of 5x3 elements and the other image is an array of 11x3 elements. correl_optimize returns a y-offset between -2,5 and 2,5 pixels. But how can the y-offset possibly be in a range of 5 pixels when both images only consist of three pixels in y direction? Same with x direction.
>
>
>
> I don't have a clue as to what I do wrong. Has someone an explanation for this?
>
>
>
> cheers,
>
> Max

Subject: Re: correlation of single pixels
Posted by Klemen on Fri, 09 Nov 2012 15:31:18 GMT

Hi Max, my code compares one image with two other images and it is not really well documented :(But if you want some exaple here is a part of my old code that is more simple, without pyramids. I am not sure if it functions as it is, as I just tried to simplifiy it...
Cheers, Klemen

```
PRO cloud_height_correlation, file, modis_path, seviri_path

  ;Cross-corealtaion settings
  d_search_area = 3L    ;width of the half of the sqaure  defining the search area
  d_window = 3          ;width of the half of the search window (sqaure)
  d_mask = d_search_area - d_window
  max_pixels = 10L*(10L)^5   ;maximal number of pixels to be processed at the same time - avoid to run out of memory

  ;Read path...
  IF ~file THEN RETURN, ''
  lastSlash = STRPOS(file, '\', /REVERSE_SEARCH)
  situation_num = strmid(file, lastSlash+1, 3)
  path = strmid(file, 0, lastSlash+1)

  ;read MODIS and create valid mask to do the cross corelation
  mo_file = FILE_SEARCH(path, situation_num + '2MO' + '*.tif', COUNT=ccc)
  im_mo = float(READ_TIFF(mo_file))
  S = REPLICATE(1, 2*d_search_area+1, 2*d_search_area+1)
  mask = read_tiff(file)
  mask = im_mo gt 0.
  mask = erode(mask, S) * (im_mo gt 0)
  in_size = size(mask)
  indx_mask = where(mask eq 1, count_mask)    ;compute ccr only for these pixels
  if count_mask eq 0 then return, ''

  ;Restore SEVIRI images
  ;file SEVIRI 1
  s1_file = FILE_SEARCH(path, situation_num + '2S1' + '*.tif', COUNT=ccc)
  im_s1 = float(read_tiff(s1_file)) ;+ 1.
  s1_file = strmid(s1_file, lastSlash+1, lastSlash+50)
  tmp = strlen(s1_file)
  s1_file = strmid(s1_file, 6, tmp-10)


  ;Seach max correlation for all OK pixels
  shift_s1_col = long(mask*0)
  shift_s1_lin = shift_s1_col
  cross_cor1 = make_array(in_size[1], in_size[2], value=-1.)

  ;prepare the indexes for the moving window
```

```
  tmp = lindgen(in_size[1], in_size[2])
  tmp = tmp - tmp[d_window,d_window]
  indx_mw = tmp[0:2*d_window, 0:2*d_window]
  tmp = !null
  indx_mw = rebin(reform(indx_mw, 1, (2*d_window+1)^2), max_pixels, (2*d_window+1)^2)

  ;because of the memory issues do not run everything at once
  for i=0L,count_mask-1,max_pixels do begin
    print, i
    if (i+max_pixels) gt (count_mask-1) then begin
      max_pixels = count_mask - i
      indx_mw = indx_mw[0:max_pixels-1, *]
    endif
    ;initialize partial results
    tmp_cor1 = make_array(max_pixels, value=-1.)   ;partial output cross-correlation
    tmp_cor2 = tmp_cor1
    tmp_s1_col = make_array(max_pixels) ;partial output shift for SEVIRI 1 columns
    tmp_s1_lin = tmp_s1_col
    ;select indexes
    indx_i = indx_mask[i:i+max_pixels-1]                    ;indexes of pixels to be processed
    indx_all = rebin(indx_i, max_pixels, (2*d_window+1)^2) + indx_mw  ;indexes of these pixels
and their vicinities
    ;prepare cross-correlation
    y = im_mo[indx_all]               ;MODIS with its vicinity
    ym = total(y, 2) / (2*d_window+1)^2 ;average of MODIS within the pixel vicinity
    ym = rebin(ym, max_pixels, (2*d_window+1)^2)
    sum_y2 = total((y - ym)^2, 2)
    sign_col = 1
    for sh_lin=0,2 do BEGIN
      for sh_col_sign=0,5 do BEGIN
        sign_col = sign_col * (-1)
        sh_col = sh_col_sign / 2 * sign_col
        print, sh_lin, sh_col
        ;SEVIRI 1
        T = systime(1)
        im_shift = shift(im_s1, sh_col, sh_lin))
        x = im_shift[indx_all]
        xm = total(x, 2) / (2*d_window+1)^2   ;average of SEVIRI within the pixel vicinity
        xm = rebin(xm, max_pixels, (2*d_window+1)^2)
        T = systime(1)
        sum_x2 = total((x - xm)^2, 2)
        T = systime(1)
        sum_xy = total((x - xm)*(y-ym), 2)
        tmp = sum_xy / sqrt(sum_x2 * sum_y2)  ;cross correlation
        indx_max = where(tmp gt tmp_cor1, count_max)
        if count_max gt 0 then begin
          tmp_s1_col[indx_max] = sh_col
          tmp_s1_lin[indx_max] = sh_lin
```

```
      tmp_cor1[indx_max] = tmp[indx_max]
    endif

  endfor

  ;For documentation
  write_tiff, path+situation_num+'shift_s1_col.tif', shift_s1_col, /float, COMPRESSION=1
  write_tiff, path+situation_num+'shift_s1_lin.tif', shift_s1_lin, /float, COMPRESSION=1
  write_tiff, path+situation_num+'cross_cor1.tif', cross_cor1, /float, COMPRESSION=1
  write_tiff, path+situation_num+'shift_s2_col.tif', shift_s2_col, /float, COMPRESSION=1
  write_tiff, path+situation_num+'shift_s2_lin.tif', shift_s2_lin, /float, COMPRESSION=1
  write_tiff, path+situation_num+'cross_cor2.tif', cross_cor2, /float, COMPRESSION=1

END
```