
Subject: correlation between images

Posted by [Wout De Nolf](#) on Thu, 02 Apr 2009 12:33:24 GMT

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

Hi all,

I have two images and I want to check (statistically) whether they are "the same" and show it to others so they will believe me :-).

The first thing I thought of was using the Pearson correlation coefficient:

```
R=correlate(img1,img2)
```

However, R close to 1 doesn't necessarily means they are "the same" or in other words, img1 vs. img2 is linear.

(Some examples of R=0.8:

<http://upload.wikimedia.org/wikipedia/commons/b/b6/Anscombe.svg>)

I can off course show the scatter plot

```
plot,img1,img2
```

```
with LADFIT(img1,img2) to draw a line through it and  
correlate(img1,img2) to add a correlation coefficient.
```

I'm still not satisfied though. Does anyone use another statistical method to compare images? (I tried IDL's `c_correlate` but I can't figure out what the result means...)

Thanks

Wox

Subject: Re: correlation between images

Posted by [Brian Larsen](#) on Thu, 02 Apr 2009 14:38:24 GMT

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

Now the details of actually how to do this escape me for now. But a few ways that come to mind as statistically convincing are:

- compute the cross correlation of the images in chunk showing that each chunk is cross correlated
- Create a difference image and compare the fluctuation in the difference to the fluctuations of the image itself or if they are statistically different from "chunk" to "chunk"
- one might be able to dream about a Kolmogorov–Smirnov test showing that the distributions of the 2 images are the same, or that the difference is normal or something like that.

But really I think you are best off beating your head on c_correlate until it makes sense.

Cheers,

Brian

Brian Larsen
Boston University
Center for Space Physics
<http://people.bu.edu/balarsen/Home/IDL>

Subject: Re: correlation between images
Posted by [Brian Larsen](#) on Thu, 02 Apr 2009 14:45:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

I got a bit interested on why I wasn't getting how to do this since I have done this before and found a couple of sites that looked helpful to me:

<http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/correlate/>
<http://mulab.physiol.upenn.edu/crosscorrelation.html>

Brian

Brian Larsen
Boston University
Center for Space Physics
<http://people.bu.edu/balarsen/Home/IDL>

Subject: Re: correlation between images
Posted by [Wout De Nolf](#) on Thu, 02 Apr 2009 16:14:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, 2 Apr 2009 07:45:43 -0700 (PDT), Brian Larsen <balarsen@gmail.com> wrote:

> I got a bit interested on why I wasn't getting how to do this since I

> have done this before and found a couple of sites that looked helpful
> to me:
>
> <http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/correlation/>
> <http://mulab.physiol.upenn.edu/crosscorrelation.html>

Ok, thanks. I'll have a look.

The KS-test (and co.) is another idea. However I'll have to think about whether it makes sense. Wouldn't it always pass when the images contain the same objects, only in other areas of the image?

Subject: Re: correlation between images
Posted by [Brian Larsen](#) on Thu, 02 Apr 2009 17:54:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

> The KS-test (and co.) is another idea. However I'll have to think
> about whether it makes sense. Wouldn't it always pass when the images
> contain the same objects, only in other areas of the image?

Yea, its sure might. I guess that I dont know enough about the images you are taking about to say for certain. One thought might be to difference the focal image and one that is the "same" and KS test that again the difference between the focal and one that is "different". But maybe I'm way off here...

Cheers,

Brian

Brian Larsen
Boston University
Center for Space Physics
<http://people.bu.edu/balarsen/Home/IDL>

Subject: Re: correlation between images
Posted by [Brian Larsen](#) on Thu, 02 Apr 2009 20:56:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK, so I was on much the same telecom three times today and needed to occupy myself...

This is an example "related" to what you are working on.

Run this as I think its pretty fun.

Download this image:

http://technabob.com/blog/wp-content/uploads/2008/03/apple_logo_rainbow_fruit.jpg

then run the following code:

```
; .run image_registration
read_jpeg, 'apple_logo_rainbow_fruit.jpg', img, /true
tv, img[1, *, *]
size = size(img, /dimen)
cut_size = 50
sub = img[* , 140:190, 300:350]
ans = fltarr(size[1:2])
FOR i = 0UL, size[1]-cut_size-1, 10 DO BEGIN
  FOR j = 0UL, size[2]-cut_size-1, 10 DO BEGIN
    ans[i, j] = c_correlate(sub, img[* , i:i+cut_size, j:j+cut_size],
0)
    tv, sub, i, j, /true
    tv, img[1,i:i+cut_size, j:j+cut_size], i, j
  ENDFOR
ENDFOR
ind=where(ans eq max(ans))
wheretomulti, ans, ind, col, row
tv, img[1, *, *]
tv, sub, col, row, /true
END
```

This is fun as it finds just the right place. To be smart one would iteratively move different numbers of pixels. As I new the answer I was able to take large steps and get the right answer, but this gave the same answer taking 1 pixel steps but takes too long.

Cheers,

Brian

Brian Larsen
Boston University
Center for Space Physics
<http://people.bu.edu/balarsen/Home/IDL>

Subject: Re: correlation between images
Posted by [Brian Larsen](#) on Thu, 02 Apr 2009 21:02:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Oops you also need this helper routine:
http://www.dfanning.com/tip_examples/wheretomulti.pro

Cheers,

Brian

Brian Larsen
Boston University
Center for Space Physics
<http://people.bu.edu/balarsen/Home/IDL>

Subject: Re: correlation between images
Posted by [Wout De Nolf](#) on Fri, 03 Apr 2009 07:45:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, 2 Apr 2009 13:56:56 -0700 (PDT), Brian Larsen
<balarsen@gmail.com> wrote:

```
> Run this as I think its pretty fun.  
> Download this image:  
> http://technabob.com/blog/wp-content/uploads/2008/03/apple\_logo\_rainbow\_fruit.jpg  
>  
> then run the following code:  
> ; .run image_registration  
> read_jpeg, 'apple_logo_rainbow_fruit.jpg', img, /true  
> tv, img[1, *, *]  
> size = size(img, /dimen)  
> cut_size = 50  
> sub = img[*, 140:190, 300:350]  
> ans = fltarr(size[1:2])  
> FOR i = 0UL, size[1]-cut_size-1, 10 DO BEGIN  
>   FOR j = 0UL, size[2]-cut_size-1, 10 DO BEGIN  
>     ans[i, j] = c_correlate(sub, img[*, i:i+cut_size, j:j+cut_size],  
> 0)  
>     tv, sub, i, j, /true  
>     tv, img[1,i:i+cut_size, j:j+cut_size], i, j  
>   ENDFOR  
> ENDFOR  
> ind=where(ans eq max(ans))  
> wheretomulti, ans, ind, col, row
```

```
> tv, img[1, *, *]  
> tv, sub, col, row, /true  
> END
```

I like it :-). `c_correlate` starts making sense to me.

Nevertheless, `c_correlate` has essentially the same problem as `correlate` doesn't it (i.e. the Anscombe problem).

Subject: Re: correlation between images
Posted by [Wout De Nolf](#) on Fri, 03 Apr 2009 12:01:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

So this is what I came up with so far (see code below) thanks to Brian's suggestions.

To say whether 2 images are "the same", I subdivided "img2" in subimages. For each subimage I try to find its best correlating position in "img1" with a loop over `c_correlate` while shifting the subimage over `img1`.

The criteria for "images are the same" are:

1. A correlation tolerance level ($R_{tol}=0.9$): each subimage must correlate better than R_{tol} with some part of `img1`
2. The subimage positions in `img1` must be regularly aligned with respect to each other within a pixel tolerances ($shifftol=0.01*imgsize$)

I have a question for you: when the criteria above are met, would you believe the images are the same? If not, do you have a better suggestion?

If you wonder what images I'm talking about, it ain't all roses like in the example. I have a distribution map of the copper content in a fragment of a painting. The second distribution map represents the azurite content (azurite is a copper containing pigment). Both should be the same (correlated) if there are no other copper containing pigments present (or if the other pigments are mixed with azurite, so having the same distribution). I want to prove that this is the case.

CODE:

```
;-----
```

```

pro test
path = Filepath(Subdir=['examples', 'data'], 'rose.jpg')
read_jpeg, path, img, /true
img = 0.3*Reform(img[0,*,*]) + 0.59*Reform(img[1,*,*]) +
0.11*Reform(img[0,*,*])

kernelSize = [10,10]
kernel = REPLICATE((1./((kernelSize[0]*kernelSize[1])), $
  kernelSize[0], kernelSize[1])
img2= CONVOL(img, kernel, /CENTER, /EDGE_TRUNCATE)

print,image_equal(img,img2,/outid)
end;pro test
;-----
function
  image_equal,img1,img2,npix=npix,shifftol=shifftol,Rtol=Rtol, outid=outid
; Image offsets or scales don't matter
; npix: subimage pixels for cross-correlation
; shifftol: subimage shift tolerance
; Rtol: cross-correlation tolerance

s1=size(img1,/dim)
s2=size(img2,/dim)
msize=s1[0]<s1[1]<s2[0]<s2[1]

if not keyword_set(npix) then npix=fix(msize*0.4)>10 ; 40% of the size
npix<=msize

if n_elements(shifftol) eq 0 then shifftol=(msize*0.01)>1 ; 1% of the
size
if not keyword_set(Rtol) then Rtol=0.9

; Subimages in img2
nsub=s2/npix
nx=nsub[0]
ny=nsub[1]
x0=npix*indgen(nx)
x1=[x0[1:.*],s2[0]]-1
y0=npix*indgen(ny)
y1=[y0[1:.*],s2[1]]-1

; img2 subimages in img1
xoff=lonarr(nsub)
yoff=xoff
xyccor=fltarr(nsub)
if keyword_set(outid) then img2recon=img1*0

; Cross-correlate subimages of img2 with img1

```

```

for i=0,nx-1 do $
  for j=0,ny-1 do begin
    sub=img2[x0[i]:x1[i],y0[j]:y1[j]]
    ssub=size(sub,/dim)-1

    ; Number of sub-shifts in img1
    noffx=s1[0]-ssub[0]
    noffy=s1[1]-ssub[1]
    ccor=fltarr(noffx,noffy)

    ; Correlate sub with img1
    for k=0,noffx-1 do $
      for l=0,noffy-1 do $
        ccor[k,l]=c_correlate(sub,img1[k:k+ssub[0],l:l+ssub[1]],0)

    ; Sub image offset and cross-correlation
    mccor=max(ccor,moff)
    k=moff mod noffx
    l=moff/noffx

    xoff[i,j]=k
    yoff[i,j]=l
    xyccor[i,j]=mccor

    if keyword_set(outid) then begin
      img2recon[k,l]=sub
      print,'Progress: ',(i*ny+j+1.)/(nx*ny)*100,'% '
    endif
  endfor

; Check whether img2 and img1 are equal
bsame=total(xyccor lt Rtol,/pres) eq 0
bsame and= total(rebin(total(xoff,2)/ny,nx,ny)-xoff gt shifftol,/pres)
eq 0
bsame and= total(rebin(reform(total(yoff,1),1,ny)/nx,nx,ny)-yoff gt
shifftol,/pres) eq 0

if keyword_set(outid) then begin
  window
  tvscl,img1,0
  tvscl,img2,1
  tvscl,img2recon,2
  tvscl,img2-img2recon,3

  xyouts,0.1,0.7,'img1',/normal,color=100
  xyouts,0.3,0.7,'img2',/normal,color=100
  xyouts,0.5,0.7,'reconstructed img2',/normal,color=100
  xyouts,0.7,0.7,'img1 - reconstructed img2',/normal,color=100

```

```
isurface,xyccor
endif
```

```
return,bsame
end;function image_equal
```

Subject: Re: correlation between images
Posted by [C.E. Ordonez](#) on Fri, 03 Apr 2009 14:43:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wox wrote:

```
> Hi all,
>
> I have two images and I want to check (statistically) whether they are
> "the same" and show it to others so they will believe me :-).
>
>
> I'm still not satisfied though. Does anyone use another statistical
> method to compare images? (I tried IDL's c_correlate but I can't
> figure out what the result means...)
>
>
> Thanks
>
> Wox
```

Another possibility is the use of "mutual information" in a manner as applied to image registration. For example, see F. Maes, et al., "Multimodality image registration by maximization of mutual information," IEEE Trans. Med. Imaging, Vol. 16, No. 2, 1997, pp187-198.

Also,

http://ranger.uta.edu/~hchen/CSE6392_2005spring/references/M I_survey_maes.pdf

-Caesar

Subject: Re: correlation between images
Posted by [Wout De Nolf](#) on Mon, 06 Apr 2009 09:56:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Fri, 03 Apr 2009 09:43:28 -0500, "C.E. Ordonez" <nerdzoo@gmail.com> wrote:

```
> Another possibility is the use of "mutual information" in a manner as
> applied to image registration. For example, see F. Maes, et al.,
> "Multimodality image registration by maximization of mutual
```

> information," IEEE Trans. Med. Imaging, Vol. 16, No. 2, 1997, pp187-198.
> Also,
> http://ranger.uta.edu/~hchen/CSE6392_2005spring/references/M I_survey_maes.pdf

Interesting. But if I use the "mutual information" (0 when image1 and image2 are indepent and 1 when they are dependent, right?) instead of the cross-correlation coefficient, how do I benefit from that?

Subject: Re: correlation between images
Posted by [Mike\[2\]](#) on Mon, 06 Apr 2009 14:35:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

What about just using the RMS difference between the two images?

Mike

Subject: Re: correlation between images
Posted by [Wout De Nolf](#) on Mon, 06 Apr 2009 15:13:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Mon, 6 Apr 2009 07:35:46 -0700 (PDT), Mike
<Michael.Miller5@gmail.com> wrote:

> What about just using the RMS difference between the two images?

Whenever I use cross-correlation, RMS difference, mutual information, Kullback-Leibler distance,...(there seem to be thousands) it always comes down to this: you have a measure for image difference (i.e. a number) which should be ideally close to 1 (or some other value, dependent on what you used). When it's "close enough", the images match.

But what is "close enough"? I guess nobody knows. Therefor I tried Brian's approach of dividing in subimages and not only put a threshold on the correlation coeff (or RMS or whatever) but also check whether the subimages are located (+/- x pixels) at their original position after the cross-correlation loop. (This can go wrong when parts of the images a just noise.)

Another way to solve the "close enough" problem is statistical hypothesis testing of the Pearson correlation coeff.
(<http://davidmlane.com/hyperstat/B62955.html>)
However this would only allow me to say that R is significantly different from zero (if p-value <= 0.05 that is) so there is some correlation. Fisher's z' transformation of R should allow testing H0:

$R=x$ but this doesn't work for $x=1$ because $z=\text{Inf}$ and $p\text{-value}=0$ (i.e. R always significantly different from 1). So this approach doesn't work (or I'm missing something).

I'm just trying to find out whether there is a more robust/objective image comparison method.

Subject: Re: correlation between images
Posted by [David Fanning](#) on Mon, 06 Apr 2009 15:46:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wox writes:

> I'm just trying to find out whether there is a more robust/objective
> image comparison method.

I know there is, although I don't know what it is.

My son is an intellectual property specialist for a company that represents photographers and sells images. They hire an Israeli company to search the web for images that have not been paid for.

Whatever software they are using (and they don't say, naturally) is VERY good. They can find images that have been flipped, re-colored, resized, clipped, and manipulated in other various ways. I'm pretty sure they are not relying totally on RMS difference. ;-)

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: correlation between images
Posted by [Mike\[2\]](#) on Mon, 06 Apr 2009 16:59:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Apr 6, 11:13 am, Wox <s...@nomail.com> wrote:

> But what is "close enough"? I guess nobody knows.

There are robust methods for evaluating statistical parametric maps (SPMs). Very generally, all of these methods involve using a set of images to calculate, at each point in the images, the statistical parameter appropriate for the null hypothesis under question. Applying a Bonferroni correction is required to keep the error rate at an acceptable level. The result is a map of the statistical parameter (or p-value) that is thresholded at a significance level. In order to account for correlations within each image, the data are often smoothed.

There is a nice online bibliography at <http://www.fil.ion.ucl.ac.uk/spm/doc/biblio/>. A good starting point might be J Comp Assisted Tomography 19 (1995) 788, "Estimating Smoothness in Statistical Parametric Maps: Variability of p Values." Or just google for SPM.

If you have only two images, you will always have trouble calculating an SPM. You could try treating the data as repeated measures of the same object. Then you could calculate a single paired t-test for the entire data set. If the test is significant, the hypothesis that the mean difference is zero could be rejected. In the sort of tomographic imaging that I'm familiar with, this is dangerous because individual points in each image are correlated with other points as a consequence of the image reconstruction methods. Another simple statistic is a z-score map (difference between test image and mean of a standard data set)/(std dev of a standard data set). That is an easy way to see if an image is consistent with a calibration data set, but again will require the proper Bonferroni corrections to avoid high error rates.

Mike

Subject: Re: correlation between images
Posted by [Mike\[2\]](#) on Mon, 06 Apr 2009 17:12:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Apr 6, 11:46 am, David Fanning <n...@dfanning.com> wrote:

- > My son is an intellectual property specialist for
- > a company that represents photographers and sells
- > images. They hire an Israeli company to search the
- > web for images that have not been paid for.
- >
- > Whatever software they are using (and they don't say,
- > naturally) is VERY good. They can find images that
- > have been flipped, re-colored, resized, clipped,
- > and manipulated in other various ways. I'm pretty
- > sure they are not relying totally on RMS difference. ;-)

That reminded me that the open source program gqview includes a tool for finding duplicate images based on a similarity measure. Maybe the code would shed some light...

Mike

Subject: Re: correlation between images
Posted by [Wout De Nolf](#) on Wed, 08 Apr 2009 08:04:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Mon, 6 Apr 2009 10:12:07 -0700 (PDT), Mike
<Michael.Miller5@gmail.com> wrote:

> That reminded me that the open source program gqview includes a tool
> for finding duplicate images based on a similarity measure. Maybe the
> code would shed some light...

The code can be found in "similar.c"

It seems that both images are subdivided in 32x32 subimages. For each subimage the average value is calculated. Then the (normalized) sum of differences between the two resulting 32x32 arrays is taken as similarity measure.

So this is similar to the RMS approach.

As for the SPMs: it seems I'll have to invest a lot of time to understand and implement this. You seem to have doubts about its use in my case. Do you think it's worth the effort?

Subject: Re: correlation between images
Posted by [Mike\[2\]](#) on Wed, 08 Apr 2009 14:20:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Apr 8, 4:04 am, Wox <s...@nomail.com> wrote:

> On Mon, 6 Apr 2009 10:12:07 -0700 (PDT), Mike

>

> <Michael.Mill...@gmail.com> wrote:

>> That reminded me that the open source program gqview includes a tool

>> for finding duplicate images based on a similarity measure. Maybe the

>> code would shed some light...

>

> The code can be found in "similar.c"

>

- > It seems that both images are subdivided in 32x32 subimages. For each
- > subimage the average value is calculated. Then the (normalized) sum of
- > differences between the two resulting 32x32 arrays is taken as
- > similarity measure.
- >
- > So this is similar to the RMS approach.
- >
- > As for the SPMs: it seems I'll have to invest a lot of time to
- > understand and implement this. You seem to have doubts about its use
- > in my case. Do you think it's worth the effort?

That depends on the details of how what you'd be using it for. The section on experimental design in www.fil.ion.ucl.ac.uk/spm/doc/intro/ might help you answer that question.

As for implementing it for a single experiment, it's not that hard. Making a general system would be complicated, but you could use the (matlab-based (please forgive my heresy)) SPM software.

Mike
