Subject: help needed in understanding correl_images
Posted by gunvicsin11 on Mon, 19 Sep 2016 08:19:21 GMT
View Forum Message <> Reply to Message

Hello all,
I have taken two images
image1(701,501)
image2(701,501)

I need to find the xshift in these two images and have used the
correl_images it gives a 15 by 15 pixel output.

But couldnt understand what this output array is.

Can anybody pls explain this.
thanks

Subject: Re: help needed in understanding correl_images
Posted by Helder Marchetto on Mon, 19 Sep 2016 10:11:16 GMT
View Forum Message <> Reply to Message

On Monday, September 19, 2016 at 10:19:25 AM UTC+2, sid wrote:
> Hello all,
> I have taken two images
> image1(701,501)
> image2(701,501)
>
> I need to find the xshift in these two images and have used the
> correl_images it gives a 15 by 15 pixel output.
>
> But couldnt understand what this output array is.
>
> Can anybody pls explain this.
> thanks

Hi,
my guess, is that what you *really* want is correl_optimize. This uses correl_images and corrmat_analyze to give you the x,y-shift.
The alternative is to use phase correlation (https://en.wikipedia.org/wiki/Phase_correlation).

Helder

Subject: Re: help needed in understanding correl_images
Posted by gunvicsin11 on Tue, 20 Sep 2016 01:51:56 GMT
View Forum Message <> Reply to Message

On Monday, September 19, 2016 at 3:41:19 PM UTC+5:30, Helder wrote:
> On Monday, September 19, 2016 at 10:19:25 AM UTC+2, sid wrote:
>> Hello all,
>> I have taken two images
>> image1(701,501)
>> image2(701,501)
>>
>> I need to find the xshift in these two images and have used the
>> correl_images it gives a 15 by 15 pixel output.
>>
>> But couldnt understand what this output array is.
>>
>> Can anybody pls explain this.
>> thanks
>
> Hi,
> my guess, is that what you *really* want is correl_optimize. This uses correl_images and corrmat_analyze to give you the x,y-shift.
> The alternative is to use phase correlation (https://en.wikipedia.org/wiki/Phase_correlation).
>
> Helder

Thank you for the idea. But the problem is, I have a scattered region so this correlation method doesn't work properly. It gives some very high value of xoffset and yoffset which is not expected. So can you please let me know is there any other method to find the shift between images.

Is it a good idea to make use of centroid.

thanks

Subject: Re: help needed in understanding correl_images
Posted by Helder Marchetto on Tue, 20 Sep 2016 08:00:59 GMT
View Forum Message <> Reply to Message

On Tuesday, September 20, 2016 at 3:51:58 AM UTC+2, sid wrote:
> On Monday, September 19, 2016 at 3:41:19 PM UTC+5:30, Helder wrote:
>> On Monday, September 19, 2016 at 10:19:25 AM UTC+2, sid wrote:
>>> Hello all,
>>> I have taken two images
>>> image1(701,501)
>>> image2(701,501)
>>>
>>> I need to find the xshift in these two images and have used the
>>> correl_images it gives a 15 by 15 pixel output.
>>>
>>> But couldnt understand what this output array is.
>>>

>>> Can anybody pls explain this.
>>> thanks
>>
>> Hi,
>> my guess, is that what you *really* want is correl_optimize. This uses correl_images and corrmat_analyze to give you the x,y-shift.
>> The alternative is to use phase correlation (https://en.wikipedia.org/wiki/Phase_correlation).
>>
>> Helder
>
> Thank you for the idea. But the problem is, I have a scattered region so this correlation method doesn't work properly. It gives some very high value of xoffset and yoffset which is not expected. So can you please let me know is there any other method to find the shift between images.
>
> Is it a good idea to make use of centroid.
>
> thanks

Hi,
I don't know what your images *really* look like (signal/noise,...), but what I use for *my* images is the phase correlation approach.
Here is roughly what I do:
- select from the images the sub-region you want to compare
- window these images (multiply by a function with 1 in the middle and zero at the edges) with either a sin function or something like a hamming window.
- make the fft of both
- multiply the fft of the first by the conj() of the fft of the second
- find the position of the max of the real_part()
- convert this to 2d array indices

It might seem tricky, but it's good to learn with.

Helder
