Subject: Re: Low pass filter - Problem with kernel
Posted by Vidhya on Tue, 10 Oct 2006 13:08:35 GMT
View Forum Message <> Reply to Message

Thank you, James. Now I have overcome the problem of kernel selection,
I indeed rebin my kernel to suit to the column_log dimensions as a 1*5
kernel.

Thanks very much.

On Oct 4, 5:23 pm, kuy...@wizard.net wrote:
> Vidhya wrote:
>> Dear All,
>> Sorry for not giving a detailed description of the program. The
>> following is the program, which tries to apply a low pass filter to an
>> image of size, 766*374 with 62 bands.
>
>> PRO vnoise
>> filename='image.hdf'
>> hdfid=hdf_sd_start(filename, /READ)
>
>> varid=hdf_sd_select(hdfid, 0)
>> hdf_sd_getdata, varid, image
>
>> varid=hdf_sd_select(hdfid, 1)
>> hdf_sd_getdata, varid, mask
>
>> hdf_sd_end_access, varid
>
>> hdf_sd_end, hdfid
>
>> column_average=rebin(image, 1, 374, 62)
>
>> ;applying the log to the average radiance of the image
>> column_log = alog10(column_average)
>
>> ksize = [3,3]
>
>> kernel = replicate((1.0/(ksize[0]*ksize[1])), ksize[0], ksize[1])
>
>> filtered_image = convol(float(column_log), kernel, /CENTER,
>> /EDGE_TRUNCATE)The problem here is that column_log is a 1x374x62 array, while kernel
> is a 3x3 array. The number of dimensions of the two arguments has to be
> the same, and each dimension of the kernal should, in general, be no
> larger than the corresponding dimension of the first argument.
>
> I'm not quite sure what you're trying to do here. Which dimensions do

> you want to do your convolution on? If it's the two spatial dimensions
> of the image, than you should be passing a 776x374 image to convol. If
> you only want to convolve along the X direction, you should use a
> kernel that has a length of only 1 in the other two dimensions.
>
>> ;filtered_image = convol(float(column_log), kernel, /CENTER,
>> /EDGE_TRUNCATE)
>> ;% CONVOL: Incompatible dimensions for Array and Kernel.
>> ;% Execution halted at: $MAIN$
>
>> And this is where I get the error message about the dimensions.
>
>> What I am trying to do is to rebin the image column-wise, apply a log
>> to the average, and then apply the kernel to the image.If you want to convolve the image, why are you convolving the
> column_log instead? What role do you want the column_log to play in the
> convolution of the image? If you do in fact want to convolve the
> column_log instead of the image itself, then you need to choose a
> kernel which has the right shape to be used for that purpose.
