Subject: creating a 2D mask for image filtering
Posted by David Higgins on Wed, 17 Aug 2011 10:47:20 GMT
View Forum Message <> Reply to Message

Hi all

I need to create a 2D mask to filter data in the frequency domain (apodization). I would like to leave 0.8 of the centre-to-edge of the data untouched (i.e. a circle of untouched data), and then a Hanning-type shape to smooth down to zero at the edges. (Think of an upside-down frying pan, kinda.) I can apply a Hanning filter with

apod_fn = HANNING(kx_res, ky_res, alpha=0.5)

but of course the centre area which I would like to have untouched doesn't exist, and the filter is too aggressive. I'd go without the Hanning shape requirement if I could get more-or-less the right shape. I see also DIGITAL_FILTER, but can't seem to widen the filter with my changes to it's arguments.

Thanks for any pointers.

Subject: Re: creating a 2D mask for image filtering
Posted by David Fanning on Wed, 17 Aug 2011 17:44:58 GMT
View Forum Message <> Reply to Message

Dave Higgins writes:

>
> Thanks for the advice, very much appreciated. I followed your general method as follows, but actually it's not the filter shape I want:
>
> FUNCTION CIRCLE, xcenter, ycenter, radius
> points = (2 * !PI / 99.0) * FINDGEN(100)
> x = xcenter + radius * COS(points)
> y = ycenter + radius * SIN(points)
> RETURN, TRANSPOSE([[x],[y]])
> END
>
> PRO apod_filter
> image = dist(512)
> WINDOW, 0, TITLE = 'Test data before filtering'
> SHADE_SURF, image
> s = Size(image, /Dimensions)
> hf = Hanning(s[0], s[1], ALPHA=0.5)
> maxRadius = Min(s)/2;
> WINDOW, 1, TITLE = 'temporary window', xsize=s[0], ysize=s[1]

> ; Use of data coords for "circle" ok since data extent may not be square
> POLYFILL, CIRCLE(s[0]/2-1, s[1]/2-1, 0.8*maxRadius), color=1
> circleMask = TVRD()
> WDELETE, 1
> indices = Where(circleMask EQ 1)
> hf[indices] = 1
> WINDOW, 2, TITLE = 'Filter to be applied'
> SHADE_SURF, hf
> WINDOW, 3, TITLE = 'Apodized image'
> SHADE_SURF, hf*image
> WDELETE, 0, 3
> END
>
> This produces a step down at the edge of the circle, to where the Hanning window was before the circleMask was applied. But this step-down causes Gibbs ringing in a FT of the data.
> I was aiming for a smooth "S" shaped (or similar) reduction of the filter values from the edge of the circle to the edge of the data.
> I was wondering if I applied some sort of smoothing to circleMask, it would blur the edge of the circle and achieve the smooth decent at the circle edge:
>
> PRO apod_filter
> image = dist(512)
> WINDOW, 0, TITLE = 'Test data before filtering'
> SHADE_SURF, image
> s = Size(image, /Dimensions)
> maxRadius = Min(s)/2;
> WINDOW, 1, TITLE = 'temporary window', xsize=s[0], ysize=s[1]
> ; Use of data coords for "circle" ok since data extent may not be square
> POLYFILL, CIRCLE(s[0]/2-1, s[1]/2-1, 0.8*maxRadius), color=1
> circleMask = float(TVRD())
> WDELETE, 1
> help, circlemask
> circleMask = SMOOTH(circleMask, 100, /EDGE_TRUNCATE, MISSING=0.0)
> indices = Where(circleMask GT 0.01)
> filter = fltarr(s[0], s[1])
> filter[indices] = circleMask[indices]
> WINDOW, 2, TITLE = 'Filter to be applied'
> SHADE_SURF, filter
> WINDOW, 3, TITLE = 'Apodized image'
> SHADE_SURF, filter*image
> WDELETE, 0, 3
> END
>
> ...but this smoothing eats back into my leave-it-alone circle of data; I'd like to start the descent to zero at the edge of the originally defined circleMask.
>
> Thanks for any further help.

Aren't you smoothing the wrong thing? I think you want this:

```
PRO apod_filter
   image = dist(512)
   WINDOW, 0, TITLE = 'Test data before filtering'
   SHADE_SURF, image
   s = Size(image, /Dimensions)
   hf = Hanning(s[0], s[1], ALPHA=0.5)
   maxRadius = Min(s)/2;
   WINDOW, 1, TITLE = 'temporary window', xsize=s[0], ysize=s[1]

   POLYFILL, CIRCLE(s[0]/2-1, s[1]/2-1, 0.8*maxRadius), color=1
   circleMask = TVRD()
   WDELETE, 1
   indices = Where(circleMask EQ 1)
   hf[indices] = 1
   WINDOW, 2, TITLE = 'Filter to be applied'
   SHADE_SURF, Smooth(hf, 50, /edge_truncate)
   WDELETE, 0
END
```

Also, TVCircle (from the Astronomy Library) will produce
a real circle no matter what the aspect ratio of the window.
One of the reasons it's one of my favorite programs. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: creating a 2D mask for image filtering
Posted by David Higgins on Thu, 18 Aug 2011 14:28:24 GMT
View Forum Message <> Reply to Message

Ah, yes, that is definitely better. The step-down is still smoothed out, but with better preservation of the central to-be-left-alone section.

I had another idea, which was to plot concentric circles as seen in
http://www.idlcoyote.com/tips/make_circle.html
to approximate the filter shape a little better before smoothing, with the result that the smoothing needn't be as aggressive, and the central section is even better preserved.

```
PRO apod_filter
   SET_PLOT, 'WIN'
   image = DIST(512)
   WINDOW, 0, TITLE = 'Test data before filtering'
   SHADE_SURF, image
   siz = SIZE(image, /DIMENSIONS)
   minRadius = MIN(siz)/2;

   thisDevice = !D.Name
   SET_PLOT, 'Z'
   DEVICE, SET_RESOLUTION=siz
   ERASE, COLOR=0

   ; 20-point Hanning-shape step-down from the edge of minRadius,
   ; using concentric circles, starting at 0.8*minRadius, to zero at minRadius
   steps = REVERSE(FINDGEN(20))
   filter_width = 0.8
   radius_frac = filter_width + steps*((1-filter_width)/N_ELEMENTS(steps))
   gap_idx =  (minRadius/2*(1-(radius_frac-filter_width)/(1-filter_width)) )
   Hanning_shape_colorvals = FIX(100* (0.5*(1-cos((2*!PI*gap_idx)/minRadius))) )
   FOR i=0,N_ELEMENTS(Hanning_shape_colorvals)-1 DO BEGIN
      points = (2*!PI/99.0) * FINDGEN(100)
      x_c = siz[0]/2-1 + FIX(ROUND(radius_frac[i]*minRadius)) * COS(points)
      y_c = siz[1]/2-1 + FIX(ROUND(radius_frac[i]*minRadius)) * SIN(points)
      circ = TRANSPOSE([[x_c],[y_c]])
      POLYFILL, circ, color=Hanning_shape_colorvals[i]
      ; Possible problem with ellipses instead of a circle; dependent on image
      ; aspect ratio, see http://www.idlcoyote.com/tips/make_circle.html
   ENDFOR
   filter = smooth(float(TVRD())/100, 10, /EDGE_TRUNCATE)

   SET_PLOT, thisDevice
   WINDOW, 3, TITLE = 'Filter to be applied'
   SHADE_SURF, filter
   WINDOW, 4, TITLE = 'Filtered image'
   SHADE_SURF, filter*image
END
```

I see your point about TVCircle to keep circles circular. Thanks. I'll look into that next.

Subject: Re: creating a 2D mask for image filtering
Posted by David Fanning on Thu, 18 Aug 2011 14:37:35 GMT
View Forum Message <> Reply to Message

Dave Higgins writes:

> Ah, yes, that is definitely better. The step-down is still smoothed out, but with better

preservation of the central to-be-left-alone section.
>
> I had another idea, which was to plot concentric circles as seen in
> http://www.idlcoyote.com/tips/make_circle.html
> to approximate the filter shape a little better before smoothing, with the result that the smoothing needn't be as aggressive, and the central section is even better preserved.

And, you realize that this:

```
SHADE_SURF, filter*image
```

Is NOT how you apply this filter to an image, right?

The Hanning filter is a frequency domain filter and
it should be applied to the frequency image created
with the FFT function.

Since you were talking about "ringing" yesterday, I
thought you understood this, but this is the second
time I've seen a plot with "Filtered Image" as a
title and the *incorrect* filtered image. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: creating a 2D mask for image filtering
Posted by David Higgins on Thu, 18 Aug 2011 14:50:18 GMT
View Forum Message <> Reply to Message

Yes, you're right of course.
I am actually already working in the frequency domain, but have named my test data set badly!

image = DIST(512)
ought to have been defined
test_k_data = DIST(512)

and so on from there.

Subject: Re: creating a 2D mask for image filtering
Posted by orifox2003 on Wed, 20 Feb 2013 19:26:32 GMT
View Forum Message <> Reply to Message

On Thursday, 18 August 2011 10:50:18 UTC-4, Dave Higgins wrote:
> Yes, you're right of course.
> I am actually already working in the frequency domain, but have named my test data set badly!
>
> image = DIST(512)
> ought to have been defined
> test_k_data = DIST(512)
>
> and so on from there.
>
> Dave Higgins

Hi guys,

Not sure if this post is still active, but I have a couple follow-up questions.
I'm trying to deconvolve an astronomical image by the telescope's PSF (PSF_tele) and then convolve by a gaussian (PSF_gauss). This is all done in fourier space, so the code looks something like this:

```
xx = fft(psf_guass)
yy = fft(psf_tele)
fftratio=xx/yy
s = Size(fftratio, /Dimensions)
hf = Hanning(s[0], s[1], ALPHA=0.5)
maxRadius = Min(s)/2
TVCircle, maxRadius*0.8, s[0]/2-1., s[1]/2-1., COLOR=1, /FILL
circleMask = TVRD()
indices = Where(circleMask EQ 1)
hf[indices] = 1
hf=smooth(hf,50,/edge_truncate)
kernel = fft(xx/yy*hf,/inverse)
```

The problem is that the resulting image has significant ringing to it. I will try to attach some images of the original PSF's and resulting image below. The images are named accordingly.

/Users/ofox/Desktop/before_after.jpg
/Users/ofox/Desktop/xx_divide_yy_times_hfilter.tiff
/Users/ofox/Desktop/hfilter.tiff
/Users/ofox/Desktop/xx_divide_yy.tiff

Thanks,

Ori

Subject: Re: creating a 2D mask for image filtering
Posted by orifox2003 on Wed, 20 Feb 2013 19:30:45 GMT
View Forum Message <> Reply to Message

On Wednesday, 20 February 2013 14:26:32 UTC-5, orifo...@gmail.com wrote:
> On Thursday, 18 August 2011 10:50:18 UTC-4, Dave Higgins wrote:
>
>> Yes, you're right of course.
>
>> I am actually already working in the frequency domain, but have named my test data set badly!
>
>>
>
>> image = DIST(512)
>
>> ought to have been defined
>
>> test_k_data = DIST(512)
>
>>
>
>> and so on from there.
>
>>
>
>> Dave Higgins
>
>
>
> Hi guys,
>
>
>
> Not sure if this post is still active, but I have a couple follow-up questions.
>
> I'm trying to deconvolve an astronomical image by the telescope's PSF (PSF_tele) and then convolve by a gaussian (PSF_gauss). This is all done in fourier space, so the code looks something like this:
>
>
>
> xx = fft(psf_guass)
>
> yy = fft(psf_tele)

>
> fftratio=xx/yy
>
> s = Size(fftratio, /Dimensions)
>
> hf = Hanning(s[0], s[1], ALPHA=0.5)
>
> maxRadius = Min(s)/2
>
> TVCircle, maxRadius*0.8, s[0]/2-1., s[1]/2-1., COLOR=1, /FILL
>
> circleMask = TVRD()
>
> indices = Where(circleMask EQ 1)
>
> hf[indices] = 1
>
> hf=smooth(hf,50,/edge_truncate)
>
> kernel = fft(xx/yy*hf,/inverse)
>
>
>
> The problem is that the resulting image has significant ringing to it. I will try to attach some images of the original PSF's and resulting image below. The images are named accordingly.
>
>
>
> /Users/ofox/Desktop/before_after.jpg
>
> /Users/ofox/Desktop/xx_divide_yy_times_hfilter.tiff
>
> /Users/ofox/Desktop/hfilter.tiff
>
> /Users/ofox/Desktop/xx_divide_yy.tiff
>
>
>
> Thanks,
>
> Ori

Sorry, no success with adding the pictures. Doesn't seem to work.