# Re: need some help with custom filter in Photoshop

***Source:*** http://sci.tech−archive.net/Archive/sci.astro.amateur/2005−02/3006.html

---

***From:*** Blaine Waddington (*v_at_z.m*)
***Date:*** 02/27/05

Date: Sun, 27 Feb 2005 22:51:28 GMT

Hi Guys:

Boy, this seems to be some tough math here but I think I'm beginning to understand it. Let me try to phrase what I think is happening:

If I have a 7x7 kernal, for example, and apply that to a 500x500 pixel image, each pixel will be changed according to the configuration of the 7x7, or would this be each 7x7 pixel area being changed?

I inputted the values for vertical lines and such, as well as tried selected blurs in only certain directions. I think I'm beginning to understand the process. Let me try to understand a bit more with the following question:

Suppose I have a tiny 5x5 pixel image for 25 square pixels. I apply a 7x7 matrix. Is this matrix too large for this tiny image? Should my image have been at least a 7x7 size as well?

Thanks,
Blaine

"Anders Thulin" <ath_no_spam_please@algonet.se> wrote in message news:jKfUd.18369$d5.142993@newsb.telia.net...
*> Blaine Waddington wrote:*
*>*
*> > I keep running into these "custom" or kernal filters in Photoshop and* other
*> > programs but, try as I might, I have been unable to find good tutorials* as
*> > to how they work with some practical input values.*
*>*
*> A textbook in image processing perhaps? *Real* image processing, not*
*> image editing or synthesizing. (I remember Gonzales & Wintz from way back,* but
*> Gonzales & Woods probably replaces that nowadays.)*
*>*
*> Kernels are square arrays (typically of odd sizes, like 3x3) containing*

a
> *number of specially selected coeffiecents. The size is the 'input area',*
> *the coefficients the weighting of the corresponding pixels in the input*
image,
> *and the actual function being performed over these pixels is typically*
> *summing follwoed by a scaling operation implicit in the kernel. Output is*
> *one single pixel orresponding to the center of the kernel.*
>
> *Processing is done by placing the kernel in all possible positions on*
the
> *input image, and creating an output image from the input pixels. (image*
> *edges get special treatment.) For example (3x3 kernel, with coordinate*
[0,0] in the center):
>
> *out[x,y] := sum of (in[x−1,y−1]*kernel[−1,−1], ...,*
in[x+1,y+1]*kernel[1,1]);
>
> *Each pixel gets multiplied with the corresponding kernel coefficient, and*
the result is
> *summed.*
>
> *If the kernel is all 1, and you add a division by 9 at the end, this is*
a simple
> *averaging filter (remember, pixels are from input image only). By changing*
the kernel,
> *you can weight pixels to influence the outcome more (though you may need*
to change the
> *9 accordingly). For instance, corner pixels may get weight 1, orthogonal*
pixels weight
> *4, central pixel weight 9 −− this filter is slightly more spatially*
sensitive than
> *the 'all 1' one.*
>
> *You can also use kernels for feature detection, such as edge detection.*
>
> *If kernel is [[−1,−1,−1],[−1,8,−1],[−1,−1,−1]], you'll get a*
> *resulting image that is all edges from the original image, i.e. places*
> *where a low pixel value was next to a high pixel, regardless of direction*
> *(though within the spatial limitations of a 3x3 kernel).*
>
> *If you change that to [−1,8,−1][−1,8,−1][−1,8,−1], you'll*
> *enhance narrow vertical lines. Typically, the results from several*
> *such 'convolutions' are combined for further processing, and possibly*
> *even used as input images for higher−level convolutions.*
>
> *3x3 pixels are a bit too primitive, though −− 5x5 or 7x7 often makes*
for
> *better results, but of course requires much more processing power.*
> *For these sizes, you need hardware assist ... or specialized hardware.*
> *In a past life, I worked with a company where 11x11 kernels were thought*
> *of as barely useful −− 19x19 and above were more like real life.*

> 
> *If you want to go deeper, look for the term 'convolution' in the*
context
> *of image processing. You may need some math background, though.*
> 
> 
> – –
> *Anders Thulin ath*algonet.se http://www.algonet.se/~ath*