

Border replication option of imfilter

Posted by **Steve Eddins**, June 15, 2007

Loren wrote about the 2007 MATLAB World Tour (<http://blogs.mathworks.com/loren/2007/04/05/2007-matlab-world-tour/>) in her Art of MATLAB (<http://blogs.mathworks.com/loren/>) blog recently. Several other MathWorks developers also attended some of the events.

One message that they brought back to us is that many of the good features we put into our products simply have not yet been discovered by many of our users. That has inspired me to think about a new kind of post on this blog - I think of it as the "Didja Know?" post. (Many of the MATLAB Desktop (<http://blogs.mathworks.com/desktop/>) blog posts are Didja Know? posts.) In my Didja Know? posts, I'll try to highlight useful functions or function options that you might not have discovered yet.

First up: the border replication option of imfilter (<http://www.mathworks.com/help/images/ref/imfilter.html>).

When you perform convolution on an image, one question to consider is: How do you compute the convolution sum for pixels along the image border? By default, imfilter assumes that image pixels "outside" the image are zero. Mathematically this is kind of reasonable, but it tends to produce a dark strip along the edge of the output image. Here's an example to show what I mean:

```
I = imread('cameraman.tif');  
imshow(I)  
imcredit('Image courtesy of MIT')
```



Image courtesy of MIT

```
J = imfilter(I, ones(5,5)/25);  
imshow(J)
```



But you can tell imfilter to handle image borders by replicating the border pixel values. That often produces a more desirable result:

```
K = imfilter(I, ones(5,5)/25, 'replicate');  
imshow(K)
```



There are other border options as well. See the `imfilter` (<http://www.mathworks.com/help/images/ref/imfilter.html>) documentation for more information.

Get the MATLAB code

Published with MATLAB® 7.4

These postings are the author's and don't necessarily represent the opinions of MathWorks.