Subject: Re: how to make a mask from a picture and how to put... Posted by David Fanning on Thu, 04 Mar 2004 17:25:03 GMT

View Forum Message <> Reply to Message

Thomas Nehls writes:

- > thank you that is interesting, but:
- >
- > I especially want to avoid drawing by hand! I already have my black and
- > white image bw (1,400,400) which results by some calculations from the
- > original image "org"(3,400,400).
- > Now I want to multiply these two images in that way, that everywhere
- > where my "bw" is black (or white, whatever) should be black in the
- > resulting image "res". the following loop did not work...
- > for x=400, Y=400
- > for X = 1:400
- for Y = 1:400
- res(Z=0:3,X,Y) = bw(X,Y) * org(Z=0:3,X,Y)
- end
- > end
- > end
- > Do you understand what I want to do?

Yes, but I was hoping you would be able to read between the lines a bit. (I don't know why I thought this. I haven't had a bit of luck this week! But there you go, an eternal optimist!)

OK, I would REFORM your B&W image into a 2D array, not a 3D. You will just get confused with that extra 1 dimension hanging around. (Or, at least, I do.)

```
image = Reform(bw)
```

I'm going to assume black image pixels are 0, everything else is something other than 0.

```
mask = image GT 0
```

Mask now contains a 1 where you want the "light" to shine through and 0 where you want to block it. If your situation is the other way around, subtract 1 from mask.

Now you have to apply the mask to the three image planes. Let's rearrange your pixel interleaved image into a band interleaved so we don't have that pesky 1 dimension to deal with:

```
maskedImage = Transpose(res, [1,0,2])
```

Now apply the mask to each color plane:

FOR j=0,2 DO maskedImage[0,0,j] = maskedImage[0,0,j]*mask

If you *have* to have a pixel-interleaved image:

maskedImage = Transpose(maskedImage, [2,0,1]) TV, maskedImage, True=1

Cheers,

David

David Fanning, Ph.D. Fanning Software Consulting Coyote's Guide to IDL Programming: http://www.dfanning.com/