Subject: Re: Image warping in IDL
Posted by Wox on Tue, 21 Nov 2006 09:18:02 GMT
View Forum Message <> Reply to Message

On Mon, 20 Nov 2006 13:17:03 -0700, JD Smith <jdsmith@as.arizona.edu>
wrote:

> Yes, as I see you figured out. Nice implementation. As you found,
> explicitly loop from 1 to omax in your histogram of repeat counts is
> fine, and solves the problem without any monkeying of indices. In
> fact, the snippet j=1,omax4-omin4 works only when omin4 is zero (which
> it seems to be always for you).

For H4, min=1, so omin4 is always 1. (I should have used j=1,omax4-1 )

So
j=0 => repeat count 1 (handle separate)
j=1 => repeat count 2 (init loop)
...

This way we skip the 0, which is what we want. These are the "empty
pixels" that need some interpolation from it's neighbours afterwards.

> j=1,omax should work. If you want to
> handle the j=1 case seperately for efficiency (as you've done), just do so
> and start the loop at 2. Also, I couldn't quite understand the
> rebin([3,2,3,2],4,npix) for selecting which 4 of the 9 output pixels
> actually receive any data. It seems like those are fixed offsets, which
> wouldn't work when the offset direction rotates around. Maybe something
> about your mapping lets you get away with that.

This is because I added a "boarder" of two pixels to the output image.

```
interimg=MAKE_ARRAY(imgs[1]+4,imgs[2]+4,type=size(*img,/type ))
```

I did this for the pixels that "fall-off". I just have to use < and >
as in:

```
off_x=0>(rebin(outpix[0,*],4,npix)+off_x)<(imgsinter[1]-1)
off_y=0>(rebin(outpix[1,*],4,npix)+off_y)<(imgsinter[2]-1)
```

After that, I cut off the 2 pixel boarder that accumulated all
fall-off pixels. I thought this was the most efficient way. Otherwise
I had to use if statements or something.