Subject: Back-projection of scatter-plot
Posted by Alex Schuster on Fri, 04 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Hi!

I have two images, img1 and img2, and a scatter-plot of them: plot, img1, img2. Now I want so select a region in the plot, and get back all pixels contained in that region. I wrote some code that works, but it is quite slow, so I'm asking if someone here has a better idea.

What I did so far:
x and y are arrays with the coordinates of the polygon I drew in the scatter-plot window. First, I convert them into the scale of [0, resolution], with resolution being a value of about 100:

```
x = round( x * resolution / ( !p.lclip[2]-!p.clip[0]+1 )
y = round( y * resolution / ( !p.lclip[3]-!p.clip[1]+1 )
```

Now I get the subscripts for this region:

```
subs = polyfillv( x, y, resolution, resolution )
```

I convert those back to x and y vectors:

```
xx = subs mod resolution
yy = subs  /  resolution
```

I scale the images to the same range:

```
img1 = fix( img1 * resolution / !x.crange )
img2 = fix( img2 * resolution / !y.crange )
```

And then I loop through all elements, using WHERE() to find all pixels correcponding to the actual pixel:

```
result = 0B * img1
for i = 0L, n_elements( xx ) do begin
  index = where( xx[i] eq img1 and yy[i] eq img2 )
  if ( index[0] ne -1L ) then result[index] = 1B
endfor
```

I thought of improving speed by not checking single pixels, but ranges: index = where(xmin ge img1 and xmax le img1 and ymin ge img2 and ymax le img2). This would back-project a rect region at once, and by filling the drawn region with such squares speed might be improved a lot. But filling a complex region recursevely with rectangles is complicated, and maybe there is an elegant, tricky way, or maybe someone already did that in IDL.

Alex
--
Alex Schuster Wonko@weird.cologne.de PGP Key available

alex@pet.mpin-koeln.mpg.de