Subject: how to I make this array (without for loops)? EXTRACT_VOLUME
Posted by mmiller3 on Thu, 25 Sep 2003 16:09:59 GMT
View Forum Message <> Reply to Message

Dear IDLers,

Ok, I've thoroughly confused myself. I'm trying to make arrays of homogeneous coordinates on a uniform grid. For example, to get a grid of points in a plane, I do this:

```
; d = pixel size (d by d)
; lx = slice width
; ly = slice height
; center = three element array giving slice center coordinate

width = long(lx/d)
height = long(ly/d)

Npixels = width*height

; x and y points along plane
xloc = center[0] - lx/2.0 + d*findgen(width) + d/2.0
yloc = center[1] - ly/2.0 + d*findgen(height) + d/2.0

; points in space (mm)
p0 = [ $
    [reform([xloc # replicate(1.0,height)], Npixels)], $
    [reform([replicate(1.0,width) # yloc], Npixels)], $
    [replicate(center[2],Npixels)], $
    [replicate(1.0,Npixels)] $
    ]
```

This generates an Npixels by 4 array of points that I can then use to interpolate an array, and is more or less lifted from EXTRACT_SLICE.

Now I want to generate a regular 3D grid of points using some equally (or more) efficient method, but I've confused my self badly with various attempts along these lines. The idea is to have an EXTRACT_VOLUME routine similar to EXTRACT_SLICE. I'm confused and if any of this makes sense (or even if it doesn't ;) I'd appreciate any light that yous can shed for me.

Here's my latest non-working attempt:

```
;width = volume width (x)
;height = volume height (y)
;depth = volume depth (z)
```

```
;Nwidth = number of voxels along width
;Nheight = number of voxels along height
;Ndepth = number of voxels along depth
;center = three element array giving slice center coordinate

Npixels = Nwidth*Nheight*Ndepth

wsize = width/Nwidth
hsize = height/Nheight
dsize = depth/Ndepth

; x and y points along plane
wloc = center[0] - width/2.0 + wsize*( findgen(Nwidth) + 0.5 )
hloc = center[1] - height/2.0 + hsize*( findgen(Nheight) + 0.5 )
dloc = center[2] - depth/2.0 + dsize*( findgen(Ndepth) + 0.5 )
stop

; points in space (mm)
p0 = [ $
    [reform(wloc # [replicate(1.0,Nheight*Ndepth)], Npixels)], $
    [reform([reform(replicate(1.0,Nwidth) # hloc, Nwidth*Nheight) # replicate(1.0,Ndepth)],
Npixels)], $
    [reform([ replicate(1.0,Nwidth*Nheight) # dloc], Npixels)], $
    [replicate(1.0,Npixels)] $
    ]
```

Thanks, Mike