
Subject: Re: texture mapping: arbitrary orientation impossible?

Posted by [Rick Towler](#) on Fri, 03 May 2002 23:16:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

"Harald von der Osten-Woldenburg" <hvdosten@lb.netic.de> wrote

- > it seems that it is only possible to combine pictures with 3-D-surfaces
- > if they are orientated parallel to the coordinate axes. Is this correct?
- > Or does perhaps anyone know a tricky method combining pictures of
- > arbitrary orientation with surfaces? For example a picture rotated by 30
- > degrees...?

You can map a texture onto a surface pretty much any way you want. The key is getting the correct texture coordinates (and understanding what OpenGL will do with them.) The texcoords map a normalized texel value to a vertex on your polygon (in your case the 3d surface). If you want to rotate the texture on the surface you change the texture coords accordingly.

I would personally work with the image data and rotate that. But if you need to dynamically rotate your texture then you could transform the texture coords. I haven't played with this a lot but I did take David Fanning's Texture Surface example (www.dfanning.com) and arbitrarily rotated the texture around. Assume you have a 2xN array of texture coordinates that map your image onto your surface. Call the array "texcoords".

```
;first we need to translate our texcoords so they  
;are centered about our local origin. This ensures  
;the texture is rotated about the center.
```

```
texcoords = texcoords - 0.5
```

```
;get the transform matrix - use IDLgrModel because  
;it does the math for us. You could easily build this  
;by hand. We'll rotate the model 30 degs about the z  
;axis.
```

```
model = obj_new('idlgrmodel')  
model -> rotate, [0,0,1], 30.  
ctm = model -> getctm()
```

```
;since we're dealing with a 2d transform, get rid of  
;the extra dimensions on our 4x4 transform matrix
```

```
ctm = ctm[0:1,0:1]
```

```
;transform the texcoords so they are rotated 30 degs  
;about 0,0  
for n=0, (size(texcoords, /dimensions))[1] do $  
    texcoords[* ,n] = texcoords[* ,n] # ctm
```

;now we translate our texcoords back

```
texcoords = texcoords + 0.5
```

Now texcoords will map your image onto your surface rotated 30 degs.

Good Luck!

-Rick Towler
