

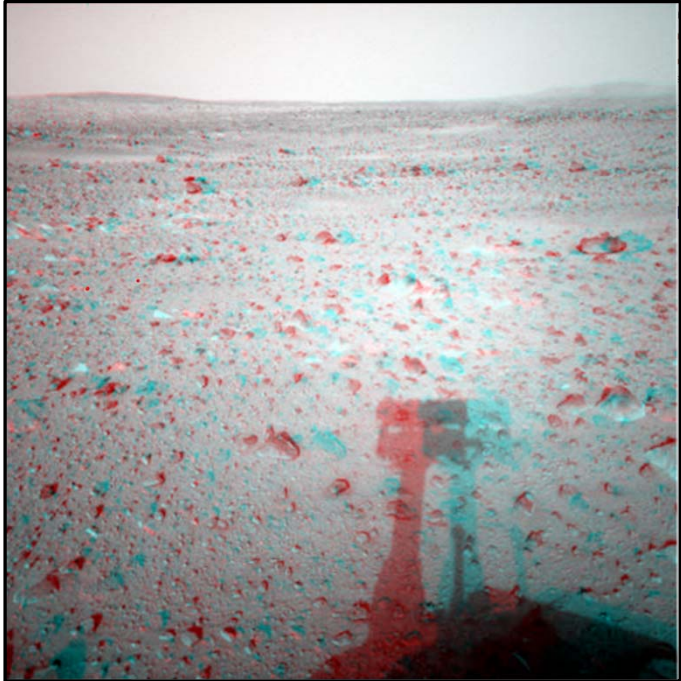
# Red-Cyan Stereographics



**Oregon State**  
University

**Mike Bailey**

mjb@cs.oregonstate.edu



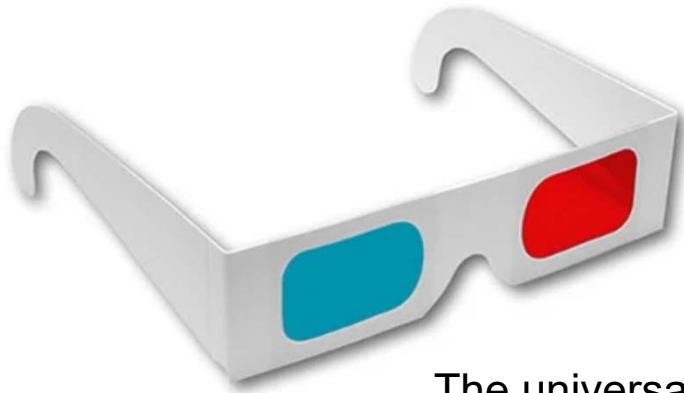
This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/)



**Oregon State**  
University

Computer Graphics

## Red-Cyan Glasses



No, they are not *red-blue* glasses!  
No, they are not *red-green* glasses!

The universal convention is:

- Red goes over the left eye
- Cyan goes over the right eye

If you want your own red-cyan glasses, one of the many places you can go is:  
<https://www.3dglasseonline.com/products/anaglyphic/>



Oregon State  
University

Computer Graphics

## anaglyph.glib

```

##OpenGL GLIB
Ortho -5. 5. -5. 5.
LookAt 0 0 1 0 0 0 0 1 0

Texture2D 5 mars_left.bmp
Texture2D 6 mars_right.bmp

Vertex anaglyph.vert
Fragment anaglyph.frag
Program Anaglyph
    uOffsetS <-.25 0.0000 .25> \
    uOffsetT <-.25 0.0000 .25> \
    uRed <0. 1. 5.> \
    uGreen <0. 1. 5.> \
    uBlue <0. 1. 5.> \
    uLeftUnit 5 uRightUnit 6

QuadXY .2 5.

```



## anaglyph.vert

```
#version 330 compatibility

out vec2 vST;

void
main( )
{
    vST = gl_MultiTexCoord0.st;
    gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
```



## anaglyph.frag

```

#version 330 compatibility

uniform sampler2D uLeftUnit, uRightUnit;
uniform float uOffsetS, uOffsetT;
uniform float uRed, uGreen, uBlue;

in vec2 vST;

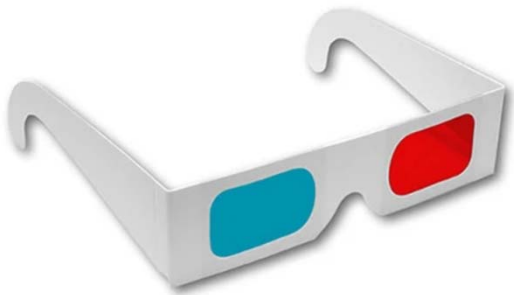
void
main()
{
    vec4 left = texture( uLeftUnit, vST );
    vec4 right = texture( uRightUnit, vST+vec2(uOffsetS,uOffsetT) );

    vec3 color = vec3( left.r, right.gb );
    color *= vec3( uRed, uGreen, uBlue );
    color = clamp( color, 0., 1. );

    gl_FragColor = vec4( color, 1. );
}

```

Note: .gb is cyan



# Mars



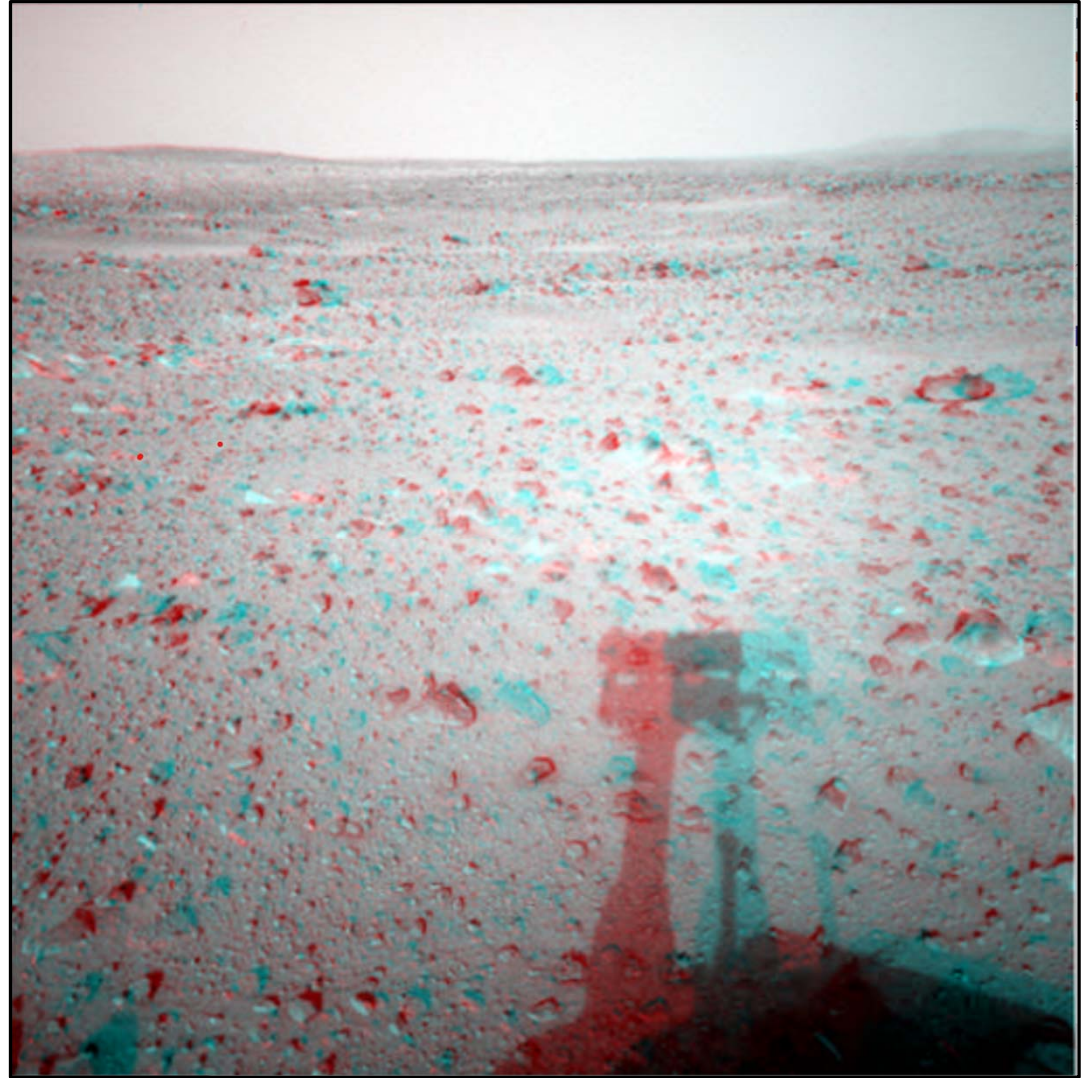
Left



Right



# Mars



Oregon State  
University

Computer Graphics

# Garden



Left



Right





# Garden

