

Displacement Textures



Oregon State
University
Mike Bailey

mjb@cs.oregonstate.edu



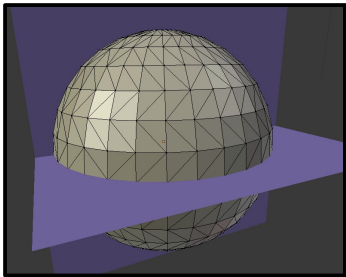
This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/)



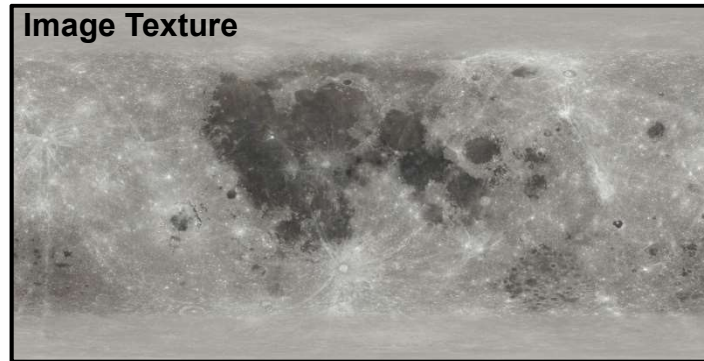
Oregon State
University
Computer Graphics

Displacement Textures as a Special Way to Model 3D Geor

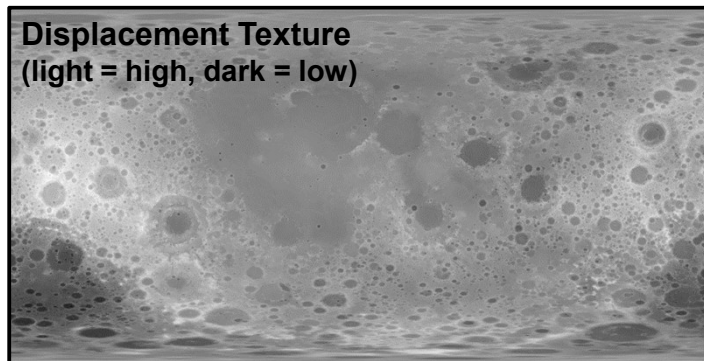
Vertex-described Object

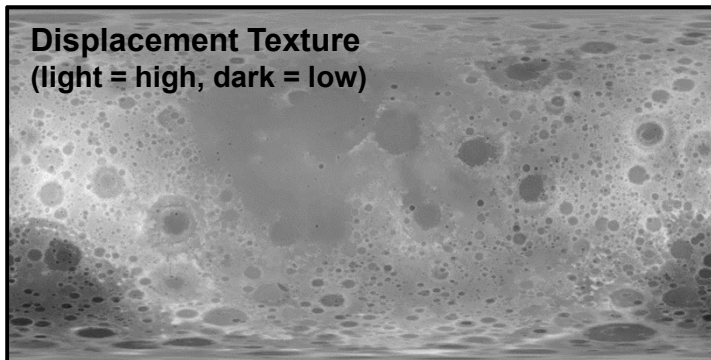
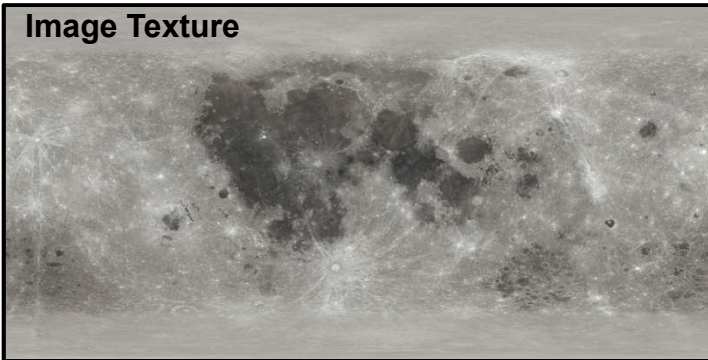


+



=





moondisp.vert

```
uniform float          uScale;
uniform sampler2D      uDispUnit;

out vec2              vST;
out vec3              vNormal;

void
main( )
{
    vec2 st = gl_MultiTexCoord0.st;
    vST = st;          // to send to fragment shader

    vec3 norm = normalize( gl_Normal );
    vNormal= normalize( gl_NormalMatrix * gl_Normal );

    float disp = texture( uDispUnit, st ).r;
                // in half-meters, relative to a radius of 1,727,400 meters
    disp *= uScale;

    vec3 vert = gl_Vertex.xyz;
    vert += norm * disp;

    gl_Position = gl_ModelViewProjectionMatrix * vec4( vert, 1. );
}
```



moondisp.frag

```
#version 330 compatibility

uniform float          uLightX, uLightY, uLightZ;
uniform float          uKd;
uniform sampler2D      uColorUnit;

in vec2                vST;
in vec3                vNormal;

void main( )
{
    vec3 light = normalize( vec3( uLightX, uLightY, uLightZ ) );
    float intensity = uKd * abs( dot( vNormal, light ) );
    intensity += (1.-uKd); // ambient
    vec3 newcolor = texture( uColorUnit, vST).rgb;
    gl_FragColor = vec4( newcolor*intensity, 1. );
}
```

