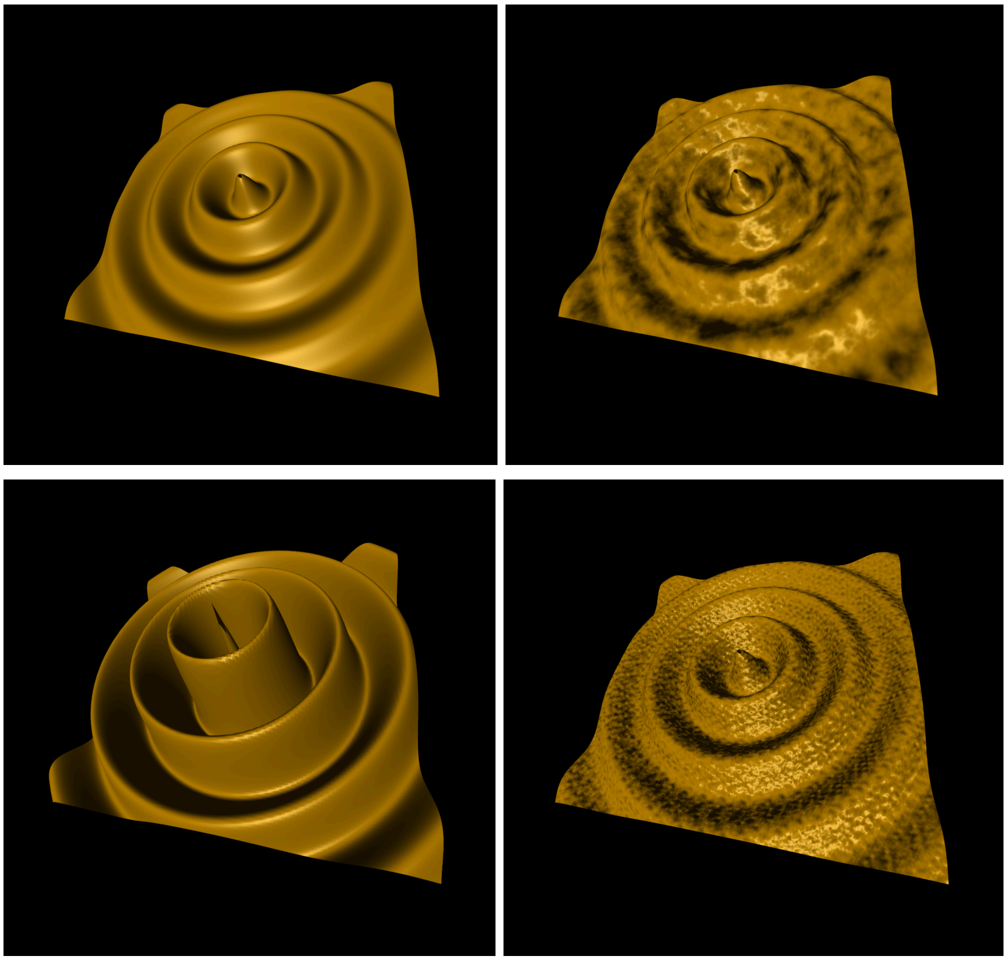
# CS 457 Project #3
# Displacement Mapping, Bump Mapping, and Lighting

Annette Tongsak
tongsaan@oregonstate.edu



Video link

To create this display, I worked in this order:

**rock.glib**

1) Set parameters for the following variables so they appear as sliders in GLman:
    a) uA - amplitude
    b) uB - period
    c) uC - phase shift
    d) uD - decay rate
    e) uNoiseAmp - noise amplitude
    f) uNoiseFreq - noise frequency
    g) uKa
    h) uKd

i) uKs
j) uShininess
k) uLightX - light position x-coordinate
l) uLightY - light position y-coordinate
m) uLightZ - light position z-coordinate
n) uColor - color of the quad
o) uSpecularColor

2) Specified the quad and its sub-quads so there are enough vertices to create a smooth displacement function

**rock.vert (Displacement Mapping)**

3) Defined M_PI
4) Defined output vec3 vMC
5) Defined uniform float variables related to light position (uLightX, uLightY, uLightZ) and the ripple equation (uA, uB, uC, uD)
6) Defined vec3 LightPosition with aforementioned variables controlled by GLman sliders
7) Found variables with known equations
   a) Defined float r using gl_Vertex.x and gl_Vertex.y, considering $r = \sqrt{x^2 + y^2}$ in the ripple equation
   b) Defined float z using the ripple equation and variables uA, uB, uC, uD, r, and M_PI
8) Differentiated $r^2 = x^2 + y^2$ with respect to x and y, respectively, to find $\frac{dr}{dx}$ and $\frac{dr}{dy}$

$$r^2 = x^2 + y^2$$
$$\frac{d}{dx}(r^2) = \frac{d}{dx}(x^2 + y^2)$$
$$2r\frac{dr}{dx} = 2x$$

$$r^2 = x^2 + y^2$$
$$\frac{d}{dy}(r^2) = \frac{d}{dy}(x^2 + y^2)$$
$$2r\frac{dr}{dy} = 2y$$

So now we have the relationships

$$2r\frac{dr}{dx} = 2x \qquad 2r\frac{dr}{dy} = 2y$$

Divide both sides by 2r to solve for $\frac{dr}{dx}$ and $\frac{dr}{dy}$

$$\frac{dr}{dx} = \frac{2x}{2r} \qquad \frac{dr}{dy} = \frac{2y}{2r}$$
$$\frac{dr}{dx} = \frac{x}{r} \qquad \frac{dr}{dy} = \frac{y}{r}$$

9) Differentiated the ripple equation $Z = A * cos(2\Pi Br + C) * e^{-Dr}$ with respect to r
   a) First used product rule and chain rule to obtain the equation for $\frac{dz}{dr}$
   b) Set float dzdr to the equation with passed in variables

10) Used found variables dzdr ($\frac{dz}{dr}$), drdx ($\frac{dr}{dx}$), and drdy ($\frac{dr}{dy}$) to ultimately get dzdx ($\frac{dz}{dx}$) and dzdy ($\frac{dz}{dy}$)

→ Recall: $\frac{dz}{dx} = \frac{dz}{dr} \cdot \frac{dr}{dx}$ $\qquad$ $\frac{dz}{dy} = \frac{dz}{dr} \cdot \frac{dr}{dy}$

So now: $\frac{dz}{dx} = \frac{dz}{dr} \cdot \frac{x}{r}$ $\qquad$ $\frac{dz}{dy} = \frac{dz}{dr} \cdot \frac{y}{r}$

11) Formed vec3 tangent vectors Tx and Ty using dzdx and dzdy, respectively
12) Normalized the cross product of Tx and Ty
13) Computed the vector from point to light
14) Computed the vector from point to eye
15) Defined vec4 new_Vertex as equal to gl_Vertex and set its z value to the z we computed in step 6b
16) Transformed newVertex from object space to clip space using model-view-projection matrix and assigned it to gl_Position
17) Set newVertex.xyz to output vec3 vMC to be passed to the fragment shader

**rock.frag (Bump-Mapping)**

Important: uColor and uSpecularColor must be passed in as vec4 and accessed with .rgb before being set to a vec3 for use

18) Used noise texture to get 2 noise values, referencing sample code
19) Rotated the normal using given function RotateNormal()
20) Multiplied the normal by the gl_NormalMatrix and normalized it
21) Set color of the quad to passed in uColor rgb values
22) Applied per-fragment lighting
    a) Ambient reflection
    b) Diffuse reflection
    c) Specular reflection
23) Assigned combination of the ambient, diffuse, and specular reflection items to gl_FragColor