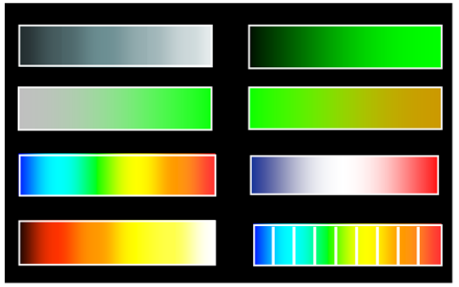
Terrain Visualization

Oregon State University
Mike Bailey
mjb@cs.oregonstate.edu

Reminder: Color Scale Transfer Functions

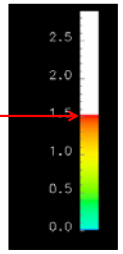


The biggest rule here is to design something that is ***intuitive***. The "snapshot rule" definitely applies!

Sometimes elevation is represented by a color transfer function, like one of these. Sometimes elevation is represented by the color of what exists at that elevation (sand, dirt, grass, trees, snow, etc.) Remember Tufte's *Do No Harm* admonition.

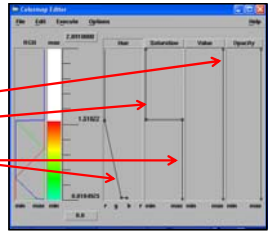
A Possible Color Scale Transfer Function for Oregon

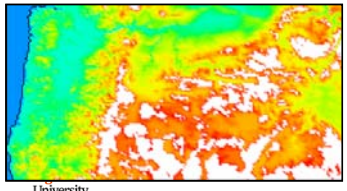
Assume snow level is at 1.5 miles

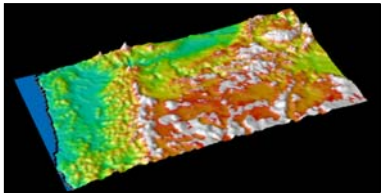


Sculpting the transfer function in HSV.

Full value, no saturation above 1.5 miles, full-saturation hue change below 1.5 miles.

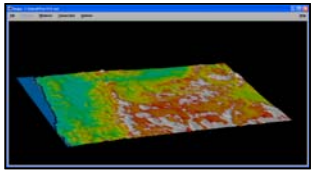




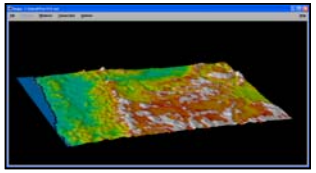


Height Exaggeration

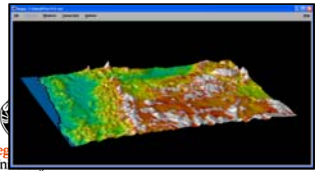
Most terrain visualization applications require height exaggeration to see any elevation changes. Why? Consider Oregon for example. Oregon is about 360 x 260 miles horizontally, and has an elevation range of about 2.5 miles vertically. This makes the elevation range less than 1% of the horizontal dimensions – hardly noticeable. However, be careful of going overboard.



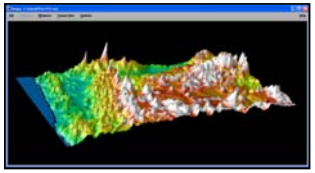
Height Exaggeration = 1.0



Height Exaggeration = 2.0

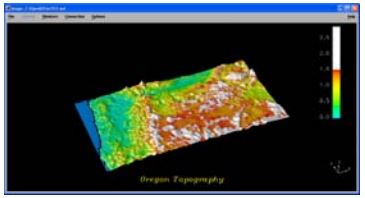


Height Exaggeration = 3.0

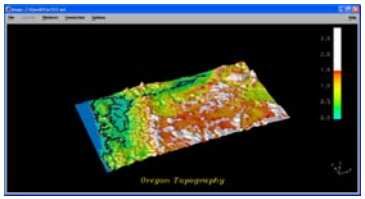


Height Exaggeration = 10.0

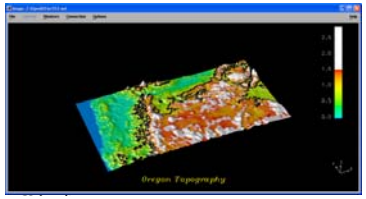
Different Contour Lines



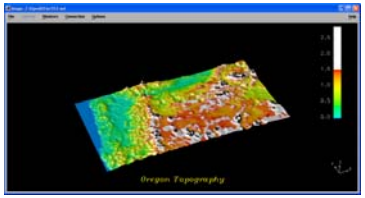
S* = 0 miles



S* = 0.17 miles

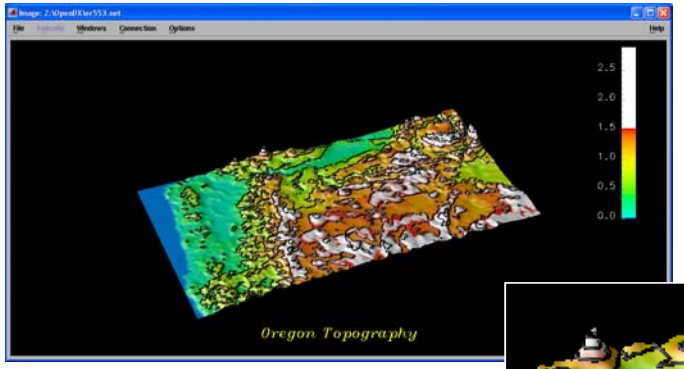


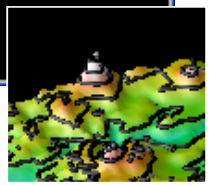
S* = 1.00 miles



S* = 2.00 miles

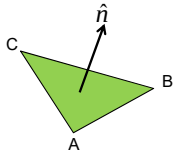
Multiple Contour Lines





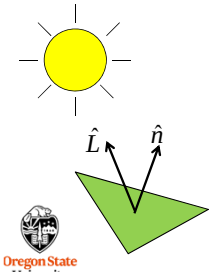
Oregon State University
Computer Graphics

Lighting



To do effective lighting of terrain surfaces, you need a surface normal for each triangle. You can get this with the cross product and unitizing:

$$\hat{n} = \frac{AB \times AC}{\left\| AB \times AC \right\|}$$



You can use this unitized normal directly in the OpenGL glNormal3f( ) call to do dynamic OpenGL lighting.

You can also do pseudo-lighting, where you assume that the sun is in a fixed direction from the scene. The diffuse portion of the lighting model is then:

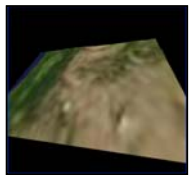
$$I_d = \hat{n} \cdot \hat{L}$$

If you assume that the sun is directly overhead, then this reduces to just the vertical component of the unit surface normal.

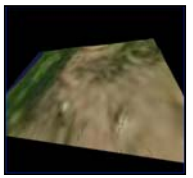
Oregon State University
Computer Graphics

Lighting Height Exaggeration

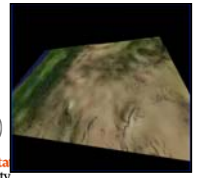
At times it is helpful to exaggerate the height for the lighting computations, but not for the height display. This ends up exaggerating the lighting effects, which is usually a good thing.



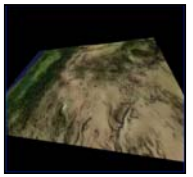
Lighting Height Exaggeration = 1.0



Lighting Height Exaggeration = 5.0



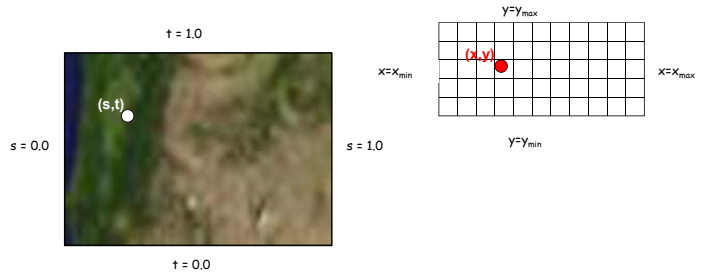
Lighting Height Exaggeration = 10.0



Lighting Height Exaggeration = 20.0

Oregon State University
Computer Grap

Computing (s,t) Texture Coordinates from Longitude (x) and Latitude (y)

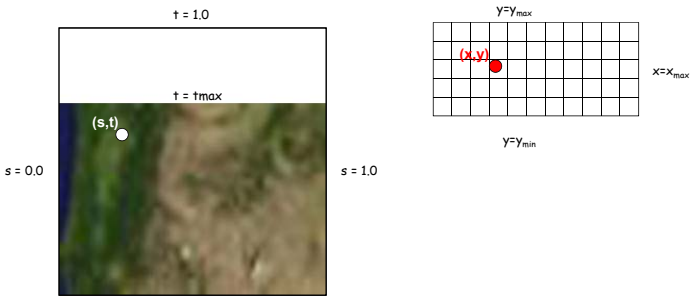


$$\frac{s-0.}{1.-0.}=\frac{x-x_{min}}{x_{max}-x_{min}} \qquad \frac{t-0.}{1.-0.}=\frac{y-y_{min}}{y_{max}-y_{min}}$$

$$s=\frac{x-x_{min}}{x_{max}-x_{min}} \qquad t=\frac{y-y_{min}}{y_{max}-y_{min}}$$

Oregon State University
Computer Graphics

Computing (s,t) Texture Coordinates: What if the Texture doesn't occupy the entire Image?

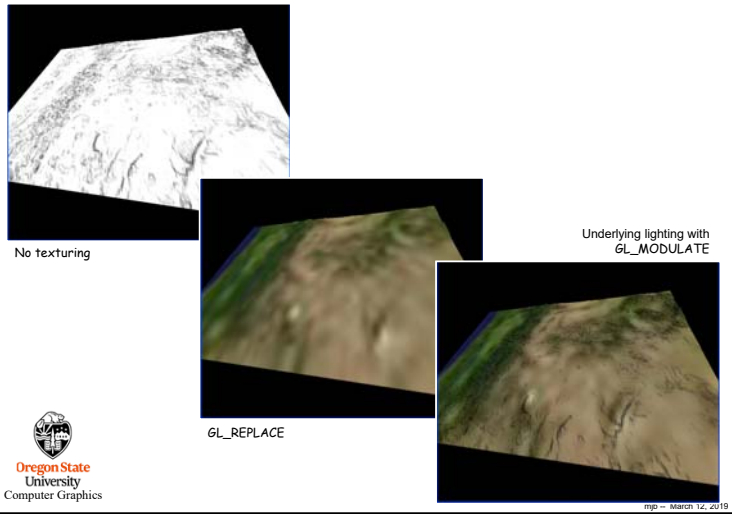


$$\frac{s-0.}{1.-0.}=\frac{x-x_{min}}{x_{max}-x_{min}} \qquad \frac{t-0.}{t_{max}-0.}=\frac{y-y_{min}}{y_{max}-y_{min}}$$

$$s=\frac{x-x_{min}}{x_{max}-x_{min}} \qquad t=\frac{t_{max}(y-y_{min})}{y_{max}-y_{min}}$$

Oregon State University
Computer Graphics

OpenGL Texture Environments

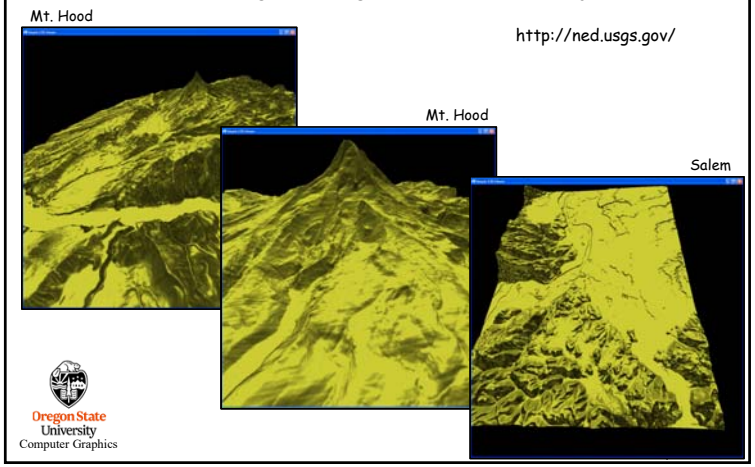


Oregon State University
Computer Graphics

USGS National Elevation Database Program

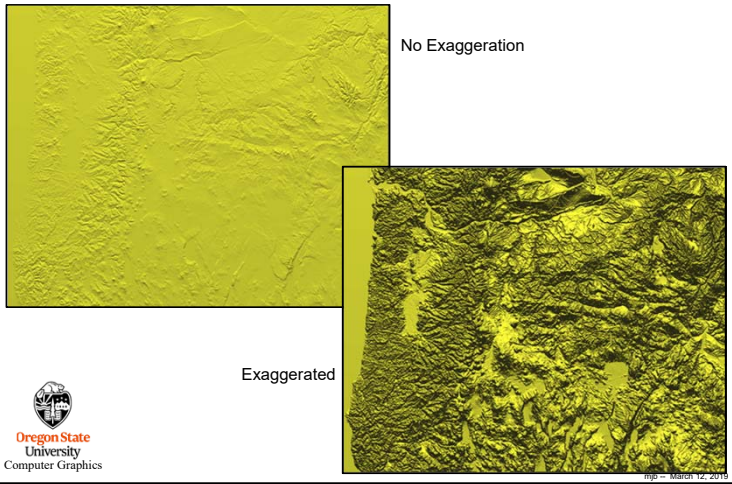
Continental US Data available free at 10m resolution.

http://ned.usgs.gov/



Oregon State University
Computer Graphics

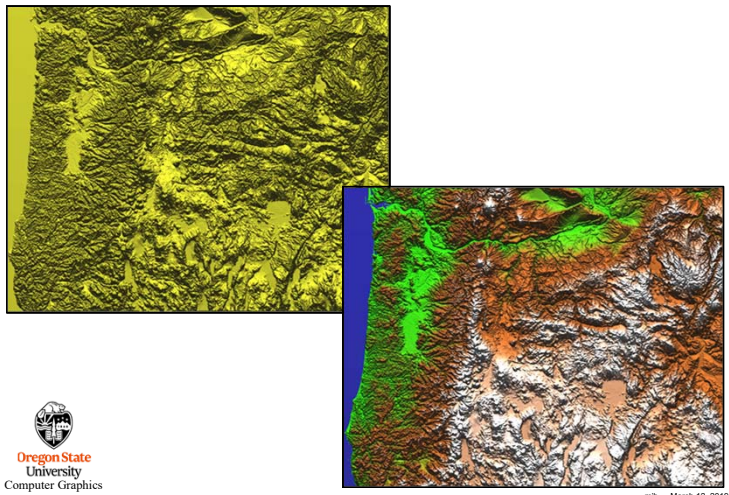
Terrain Height Bump-mapping: Don't Actually Supply the Geometry, Just Manipulate the Normals



Oregon State University Computer Graphics

mjb -- March 12, 2019

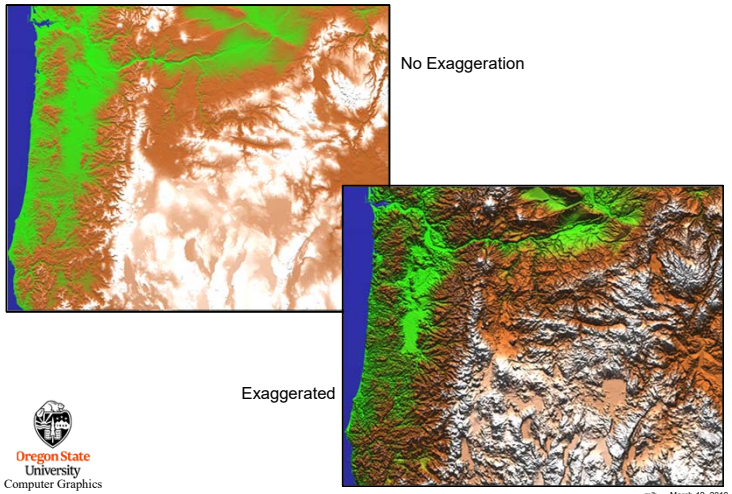
Terrain Height Bump-mapping: Coloring by Height



Oregon State University Computer Graphics

mjb -- March 12, 2019

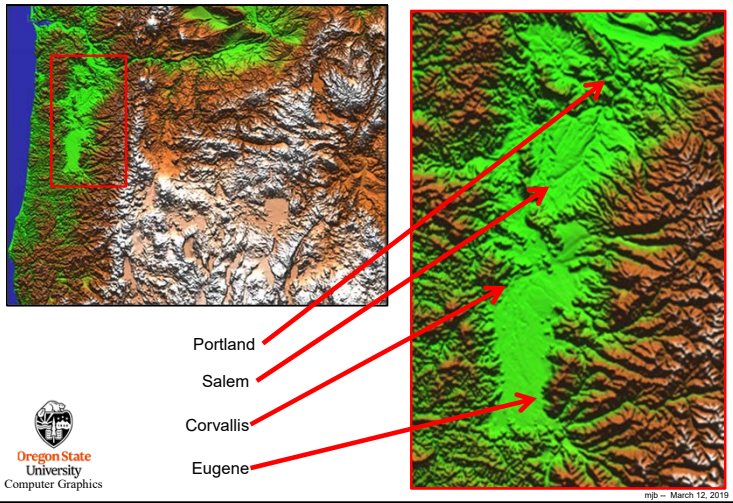
Terrain Height Bump-mapping: Coloring by Height



Oregon State University Computer Graphics

mjb -- March 12, 2019

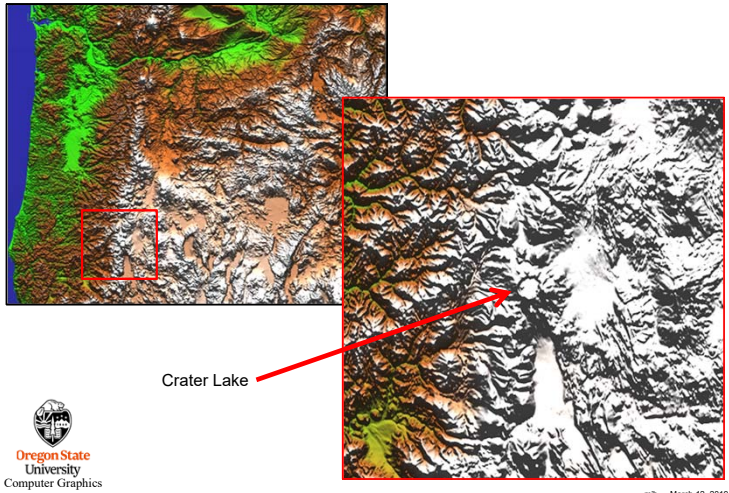
Terrain Height Bump-mapping: Zooming In



Oregon State University Computer Graphics

mjb -- March 12, 2019

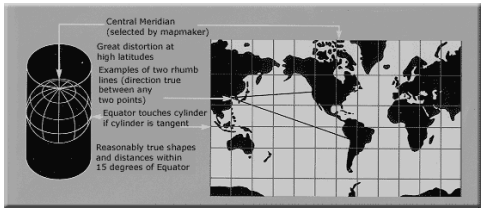
Terrain Height Bump-mapping: Zooming In



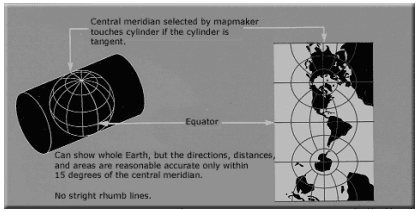
Oregon State University
Computer Graphics

Map Projections

Mercator



Transverse Mercator

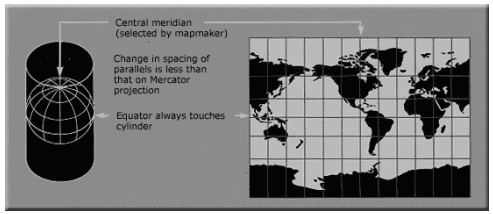


Oregon State University
Computer Graphics

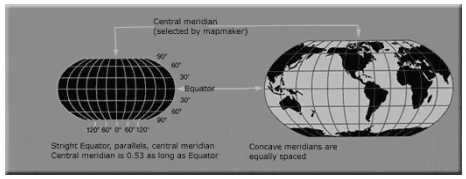
http://egsc.usgs.gov/isb/pubs/MapProjections/projections.html

Map Projections

Miller Cylindrical



Robinson

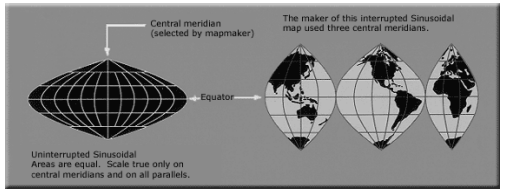


Oregon State University
Computer Graphics

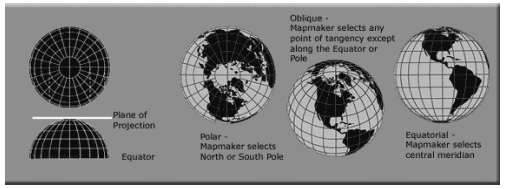
http://egsc.usgs.gov/isb/pubs/MapProjections/projections.html

Map Projections

Sinusoidal Equal Area



Orthographic

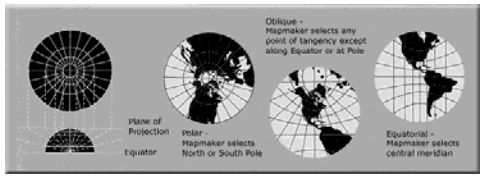


Oregon State University
Computer Graphics

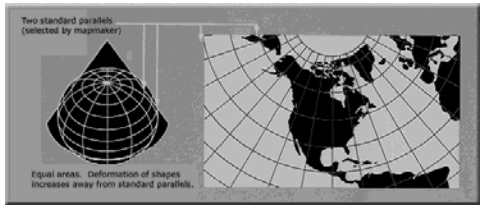
http://egsc.usgs.gov/isb/pubs/MapProjections/projections.html

Map Projections

Gnomonic



Albers Equal Area Conic



Oregon State University
Computer Graphics

http://egsc.usgs.gov/isb/pubs/MapProjections/projections.html

mjb -- March 12, 2019