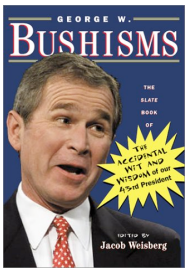
Opacity



Def. 1: Quality of a body that makes it impervious to light
Def. 2: Obscurity of sense: UNINTELLIGIBLENESS
Def. 3: The quality or state of being mentally obtuse.

Misunderstimated?
Sublimable?
Hopefuller?

"I know how hard it is for you to put food on your family."

"I know the human being and fish can coexist peacefully."

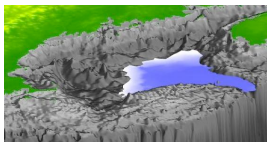
Outline

- Announcements:
  - Homework III due Friday by 5, by e-mail
  - Homework II: answers on web
- Homework II
- Gulf of Maine Example II
- Controlling opacity and using it for science
- Gulf of Maine Example III

Homework II

- Nice job!
- #2 Seemed to be the hardest:
  - Problem: you want to compare two color-scaled plots (pcolor). How do you make sure red=same value in each figure?
    - Make sure they have the same clims:
      - figure(1);lim(1,:)=get(gca,'clim');
      - figure(2);lim(2,:)=get(gca,'clim');
      - lim=[min(lim(:,1)),max(lim(:,2))];
      - figure(1);set(gca,'clim',lim);
      - figure(2);set(gca,'clim',lim);

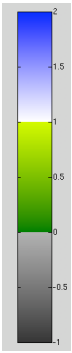
GOM Example II



- Showed on Friday how to produce bathymetry
- Now, will add temperature data
  - Data consists of discrete observations of T (measured as % of ideal water mass) at arbitrary locations (lon, lat, depth)
  - I interpolated on to a triangular mesh using objective analysis
    - Gives value at each vertex and
    - Variance (uncertainty) at each vertex
  - If variance exceeded some threshold or values were less then 20%, I replaced with NaN
  - Plotted as patch with color proportional to %

GOM Example II

- 3 Colormaps:
  - Blue--mapped temps [20 100] to Cdata [1 2];
  - Green-mapped altitude [0 4.5] to Cdata [0 1] ;
  - Gray--mapped depth [-6000 0] to Cdata [-1 0];
- Set Clim to [-1 2]



OpenGL™

- OpenGL is *the* graphics library
  - Started as proprietary library on Silicon Graphics workstations
  - Now available everywhere (standard with most systems)
  - Tightly coupled with graphics cards
  - Underlying system for most games and scientific visualization systems

- Built-in primitives to draw points, lines, polygons
- Can easily transform objects in 3D
  - scale, rotate, translate, change viewpoint
- Can control opacity of objects, not just color
- Can add textures to objects
- For more info:
  - www.opengl.org or
  - CS417

& Matlab

- In Version 6.0, Matlab added OpenGL as a possible renderer
  - Property of figure, controls how objects are translated to pixels
  - OpenGL is now the default renderer for surfaces and patches
    - When you add a surface or patch, Matlab switches to OpenGL
- Adding OpenGL improves graphics performance
- Adding OpenGL adds some new functionality

Controlling Opacity

- Opacity is controlled in a similar way to color
  - Uses "Alpha" fields
  - An alpha is a number between 0 and 1
    - 0==transparent, 1==opaque

Controlling Opacity

	Colors		Alphas	
Object	Property	Options	Property	Options
surface	FaceColor	none, flat, interp, or a color	FaceAlpha	Flat, interp, or an alpha (1)
	Cdata	Matrix specifying color data (for flat or interp)	AlphaData	Matrix specifying alpha data (for flat or interp)
Patch	FaceColor	Same	FaceAlpha	same
	FaceVertexColorData	Color values at vertices (taken from Cdata, if necessary)	FaceVertexAlphaData	Alpha values at vertices (no AlphaData for patches!)
Figure	ColorMap	Matrix of rgb values (jet)	AlphaMap	Vector of alphas (linspace(0,1,64))
Axes	Clim	Controls mapping of Cdata values to colors	Alim	Controls mapping of AlphaData (or FaceVertexAlphaData) to alphas

Controlling Opacity

- So, for patches & surfaces we can specify opacity either
  - Directly--by setting facealpha to a value, or
  - Indirectly--by setting facealpha to flat or interp and filling AlphaData (or FaceVertexAlphaData) with data values
    - Can control the appearance by changing figure's AlphaMap and axes' Alim

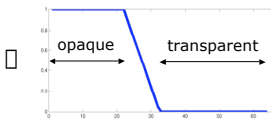
So what?

- Why would you want to control opacity?
  - See inside closed surfaces
  - Represent another dimension of data (next example)
  - It's cool

Making transparency useful

- Statistical interpolation techniques (like objective analysis) give you a distribution of values and an estimate of their accuracy (error variance)
- Most people will simply plot the interpolated data and ignore the error maps
- Ideally, we would incorporate error into the image so that it is easy to tell which values we believe

New GOM Figure



- Want to incorporate error in a less arbitrary manner
  - Let transparency be proportional to error
  - 1) create a surface at Z=-100 m with color proportional to temp
  - 2) set its FaceAlphaVertexData to Err & FaceAlpha to interp
  - 3) set figure's alphamap to
    - amap=interp1([0.2 0.6 0.8 1.4],[1 1 0 0],linspace(0.2,1.4,64));
  - 4) set axes' Alim to [0.2 1.4]