Volumetric Visualization

Outline

- Announcements
  - PS III due today
  - Contest entries by tomorrow
  - PS IV online today--GUI vs. VV
  - Demos on Friday
- What is VV?
- Slices
- Isosurfaces
- Controlling transparency
- Example: CT data

What is VV?

- \( y = f(x) \) => lines
- \( z = f(x,y) \) => surfaces
- \( v = f(x(t),y(t),z(t)) \) => surfaces with color \( z \)
- \( v = f(x,y,z) \) => true 3D data?
Representing \( V = f(x,y,z) \)

- we have a value of \( V \) for every point in a volume
- \( V \) is a cube of data (m-by-n-by-p)
- We need \( X, Y, Z \) of the same size to indicate positions of data
  - Typically, we have a regular grid defined by vectors \( x, y, \text{ & } z \)
  - \([X,Y,Z]=	ext{meshgrid}(x,y,z)\) produces 3D arrays needed by Matlab's \( V \) functions
    - \( X, Y, Z \) are m-by-n-by-p
    - for all \( j \) & \( k \), \( X(j,:,k)=x, Y(:,j,k)=y, Z(j,k,:)=z \)

Example: CT data

- CT scans ("Computerized Tomography" a.k.a. CAT scans) produce a series of cross-sectional x-rays
- Several slices can be stacked together to form a volume

Example: CT data

- CT scans of head and thorax from dogs provided by Dr. Ned Dykes at NYSCVM
  - Each slice is a separate tiff file
    - loaded each tiff with imread
    - stacked into array head
  - Thinned the data set
    - \([X,Y,Z,\text{Head}]=	ext{reducevolume}(X,Y,Z,\text{Head},[4,4,1]);\)
  - Cropped data
    - Head\_reduce4.4.1\_crop.mat
**Visualizing V**

- Simplest way is to look at a particular row/column/layer of V:
  - `pcolor(x,y,V(:,:,k))` -- layer k
  - `pcolor(x,z,squeeze(V(:,:,k)))` -- column k
  - `pcolor(y,z,squeeze(V(:,:,k)))` -- row k
- `squeeze` removes singleton dimensions
  - `v(:,:,k)` is 1-by-n-by-p
  - `squeeze(v(:,:,k))` is n-by-p

**General Slicing**

- `h=slice(X,Y,Z,V,xs,ys,zs)`
  - slices V at multiple planes
  - `slice(X,Y,Z,V,[20 30],[],[10])` produces 3 slices:
    - x=20, x=30, z=10
  - What if a slice falls between a row or column?
- `h=slice(X,Y,Z,V,Xsurf,Ysurf,Zsurf)`
  - slices V with a surface defined by Zs = f(Xs,Ys)

**Slicing the dog**

- `clearslice(Xs,Ys,Zs,Heads,xs,ys,zs,thresh)`
  - same as `slice(Xs ...)` but values of Heads below the threshold are set to clear
Controlling Opacity

- The opacity of an object is controlled like color
  - can specify edgealpha and facealpha of a surface object
    - can be set to a particular alpha level
      \(0=\text{transparent, } 1=\text{opaque}\)
    - can be set to flat or interp just like colors
    - in this case, Matlab uses determines opacity from
      - alphadata of surface (like cdata)
      - alphamap of figure (like colormap, but n-by-1)
      - alim of axes (like clim)

Isosurfaces

- Before perspective plots and color mapping, people plotted \(z=f(x,y)\) with countours:
  - curves of constant \(z\)
- Isosurfaces are analogous methods in 3D
  - find \(X,Y,Z\) s.t. \(f(X,Y,Z)=v\)

Isosurfaces

- \(fv=\text{isosurface}(X,Y,Z,V,v)\);
- \(fv\) is a struct describing a patch (or surface) object on a triangular mesh
- \(fv\text{.vertices}(j,:)=\text{position of jth vertex}[x\text{, } y\text{, } z]\)
- \(fv\text{.faces}(j,:)=1\text{-by-}3\text{-index to 3 vertices forming triangle (like tri)}\)
- \(h=\text{patch}(fv)\) will display the surface
  - set \(h\text{'edgecolor'},'none','facecolor',\text{Colorspec,'facealight'\text{'phong'}}\)
**Isosurfacing the Dog**

- `visiso(Xs, Ys, Zs, Hs, v)` -- makes a pretty isosurface, adds a light source