Interactive rotation

Oliver W. Layton

CS251: Data analysis and visualization

Lecture 10, Spring 2019

Wednesday February 27

Plan

- Interactive rotation
- Rotation matrices

Convert click-and-drag to rotation angle

1. Save the screen position of the initial mouse click (P_0).
2. Save initial (*pre-rotation*) observer's axes (U, VUP, VPN): $\overrightarrow{U_0}, \overrightarrow{VUP_0}, \overrightarrow{VPN_0}$.
3. For each mouse movement i:
   - Calculate the current horizontal screen displacement relative to the initial click (pixels): $\Delta x = P_{ix} - P_{0x}$
   - Convert to rotation about VUP, where dragging mouse across entire app viewing window brings us a full-circle around the data: $\theta_h = \dfrac{-2\pi\Delta x}{s_x}$
   - Calculate the current vertical screen displacement relative to the initial click (pixels): $\Delta y = P_{iy} - P_{0y}$
   - Convert to rotation about U, where dragging mouse across entire app viewing window brings us a full-circle around the data: $\theta_v = \dfrac{2\pi\Delta y}{s_y}$

Circling rotation implementation

Compute joint x-y rotation $X(\theta_h, \theta_v)$:

$$T\left(VRP + \frac{E_z}{2}VPN\right) R_{XYZ}^T R_x(\theta_v) R_y(\theta_h) R_{XYZ} T\left(-VRP - \frac{E_z}{2}VPN\right)$$

where

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) & 0 \\ 0 & sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\theta) & 0 & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$