CSE6338: Assignment 2

Burton Ma

Posted: Sat Oct 27, 2012
Due: Wed Nov 14, 2012

1. The ICP algorithm repeatedly searches for a point on the model surface that is closest to each registration data point. For surfaces defined by a mesh of triangles, the search requires solving for the point in a triangle closest to the data point. One way to solve for the closest point is to first find the point $\mathbf{x}$ on the plane that contains the triangle that is closest to the data point, and then determine if $\mathbf{x}$ is inside the triangle.

   Suupose that you are given a triangle defined by the points $\mathbf{a} = [a_x\ a_y\ a_z]^T$, $\mathbf{b} = [b_x\ b_y\ b_z]^T$, and $\mathbf{c} = [c_x\ c_y\ c_z]^T$, and lying in the plane with unit normal vector $\mathbf{n} = [n_x\ n_y\ n_z]^T$.

   (a) Find an expression for $\mathbf{x}$, the point on the plane (containing the triangle) closest to the data point $\mathbf{y}$.

   (b) The points inside the triangle are defined by the following equation:

   $$\mathbf{a} + u(\mathbf{b} - \mathbf{a}) + v(\mathbf{c} - \mathbf{a})$$

   subject to the following constraints:

   $$\begin{aligned} 0 \leq u \leq 1 \\ 0 \leq v \leq 1 \\ u + v \leq 1 \end{aligned}$$

   If any of the constraints are broken, then the point is not inside the triangle, but it is on the plane containing the triangle.

   Given a point $\mathbf{x} = [x_x\ x_y\ x_z]^T$, solve for u and v. Make sure that your solution works for all reasonable triangles.

2. The equation of a line defined by two points $\mathbf{a}$ and $\mathbf{b}$ is

   $$\mathbf{a} + u(\mathbf{b} - \mathbf{a})$$

   where u is a scalar value. Given a measured point $\mathbf{x}$ having measurement covariance $\mathbf{\Sigma}$, find the point on the line closest to $\mathbf{x}$ in terms of the Mahalanobis distance.

3. Deriving the spatial stiffness matrix for surface-based registration is tedious because the rotation part of the displacement requires matrix multiplications involving three different rotation matrices; however, if we discard the rotation, then the derivation is straightforward.

Derive the upper-left 3×3 block of the spatial stiffness matrix for shape-based registration, ignoring the rotation part of the displacement. In this case, the potential energy for one spring is given by

$$U_i = \frac{1}{2}((\mathbf{q}_i - \mathbf{p}_i) \cdot \mathbf{n}_i)^2$$

where $\mathbf{q}_i$ is $\mathbf{p}_i$ displaced by a translation $\mathbf{t} = [t_x \; t_y \; t_z]^T$; i.e., $\mathbf{q}_i = \mathbf{p}_i + \mathbf{t}$. You need to compute the Hessian matrix

$$\mathbf{H}_i = \begin{bmatrix} \frac{\partial^2 U_i}{\partial t_x^2} & \frac{\partial^2 U_i}{\partial t_x \partial t_y} & \frac{\partial^2 U_i}{\partial t_x \partial t_z} \\ & \frac{\partial^2 U_i}{\partial t_y^2} & \frac{\partial^2 U_i}{\partial t_y \partial t_z} \\ & & \frac{\partial^2 U_i}{\partial t_z^2} \end{bmatrix}$$

4. Implement a variation of the pivot calibration method described in the lecture from Oct 17. For the purposes of this question, a full calibration is not required; you only need to find $\mathbf{p}^c$, the location of the pivot point in camera coordinates.

   The technique described in class used only 2 poses to obtain a calibration; your implementation must somehow extend the technique so that it uses information from n poses (where n is greater than 2).

   Test your implementation using the actual calibration data stored in the file `a2.mat` (available from the assignment web page). You can load the file in Matlab using the command

   ```
   load a2
   ```

   The file will load 4 Matlab variables `m1`, `m2`, `m3`, and `m4`. Each variable is a $3 \times n$ array of marker points (`m1` are points for marker 1, `m2` are points for marker 2, and so on).