CS 4620 Preliminary Exam #1

Tuesday 5 October 2010—50 minutes

Explain your reasoning for full credit.
You are permitted a double-sided sheet of notes.
Calculators are allowed but unnecessary.

Problem 1: 2D Transformations (15 pts)

(i) Estimate the 2D affine transformation matrix, $\mathbf{T} = \begin{bmatrix} \boldsymbol{F} & \boldsymbol{v} \\ 0^T & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3}$, given its action on three homogeneous points:

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \xrightarrow{\boldsymbol{T}} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \qquad \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} \xrightarrow{\boldsymbol{T}} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \qquad \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} \xrightarrow{\boldsymbol{T}} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}.$$

- **Answer:** You need to estimate $\boldsymbol{F}$ and $\boldsymbol{v}$ from its observed transformation of some 2D points,

$$\begin{bmatrix} \boldsymbol{F} & \boldsymbol{v} \\ 0^T & 1 \end{bmatrix} \begin{pmatrix} \boldsymbol{p}_k \\ 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}'_k \\ 1 \end{pmatrix} \qquad \Longleftrightarrow \qquad \boldsymbol{F}\boldsymbol{p}_k + \boldsymbol{v} = \boldsymbol{p}'_k,$$

where we will denote

$$\boldsymbol{p}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \longrightarrow \boldsymbol{p}'_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \tag{1}$$

$$\boldsymbol{p}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \longrightarrow \boldsymbol{p}'_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \tag{2}$$

$$\boldsymbol{p}_3 = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \longrightarrow \boldsymbol{p}'_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{3}$$

There are numerous ways to solve for $\boldsymbol{F}$ and $\boldsymbol{v}$, with the least easy being setting it up as a large 6x6 system of equations. Another way is to solve for $\boldsymbol{F}$ first (then compute the translation $\boldsymbol{v}$) by subtracting one equation from the other two

$$\boldsymbol{F}(\boldsymbol{p}_k - \boldsymbol{p}_j) + \boldsymbol{v} = (\boldsymbol{p}'_k - \boldsymbol{p}'_j),$$

to obtain the matrix form,

$$\boldsymbol{F}\,\big[(\boldsymbol{p}_2 - \boldsymbol{p}_1) \quad (\boldsymbol{p}_3 - \boldsymbol{p}_1)\big] = \big[(\boldsymbol{p}'_2 - \boldsymbol{p}'_1) \quad (\boldsymbol{p}'_3 - \boldsymbol{p}'_1)\big] \qquad \Longleftrightarrow \qquad \boldsymbol{F}\boldsymbol{P} = \boldsymbol{P}',$$

and then solve the 2x2 linear system for $\boldsymbol{F}$,

$$\boldsymbol{F} = \boldsymbol{P}'\boldsymbol{P}^{-1}.$$

Finally you get the translation from any equation

$$\boldsymbol{v} = \boldsymbol{p}'_k - \boldsymbol{F}\boldsymbol{p}_k.$$

The simplest way is to observe that the data makes this problem easy: the first two points (which differ by the y coordinate) map to the same place, $\boldsymbol{p}'_1 = \boldsymbol{p}'_2$, so the transformation $\boldsymbol{F}$ must have a zero "y column." For example, in a column-oriented form you can let $\boldsymbol{F} = [\boldsymbol{t}\,\boldsymbol{u}]$ where $\boldsymbol{t}, \boldsymbol{u} \in \mathbb{R}^2$, and let $\boldsymbol{p}_k = (x_k, y_k)^T$, so that we need to solve for the 3 vectors $\boldsymbol{t}, \boldsymbol{u}$ and $\boldsymbol{v}$:

$$x_k \boldsymbol{t} + y_k \boldsymbol{u} + \boldsymbol{v} = \boldsymbol{p}'_k, \quad k = 1, 2, 3.$$

This leads to the system of equations

$$\begin{aligned} \boldsymbol{t} + \boldsymbol{u} + \boldsymbol{v} &= \begin{pmatrix} 2 \\ 1 \end{pmatrix} & (4) \\ \boldsymbol{t} - \boldsymbol{u} + \boldsymbol{v} &= \begin{pmatrix} 2 \\ 1 \end{pmatrix} & (5) \\ -\boldsymbol{t} - \boldsymbol{u} + \boldsymbol{v} &= \begin{pmatrix} 0 \\ 1 \end{pmatrix}. & (6) \end{aligned}$$

Subtracting the first two equations yields

$$\boldsymbol{u} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

afterwhich adding the last two equations yields

$$\boldsymbol{v} = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

and so

$$\boldsymbol{t} = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

and the transformation matrix is

$$\mathbf{T} = \begin{bmatrix} \boldsymbol{F} & \boldsymbol{v} \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

(ii) What kind of transformation does this matrix represent?

- **Answer:** It applies a nonuniform scale of $(s_x, s_y) = (1, 0)$ (which projects away all y components) then translates by $(1, 1)$.

Problem 2: Affine Transformations (10 pts)

Show that affine transformations preserve parallel lines.
(Hint: Recall the explicit parameterization of a line.)

- **Answer:** Consider two parallel lines whose points can be explicitly represented as

$$\boldsymbol{p}_k + t\boldsymbol{u}, \quad k = 1, 2,$$

where each is parameterized by some $t \in \mathbb{R}$, and both point in the same $\boldsymbol{u}$ direction (by virtue of being parallel) but have different base points $\boldsymbol{p}_k$. Applying an affine transformation $\begin{bmatrix} \boldsymbol{F} & \boldsymbol{v} \\ 0^T & 1 \end{bmatrix}$ to such points produces new line points

$$\boldsymbol{F}(\boldsymbol{p}_k + t\boldsymbol{u}) + \boldsymbol{v}, = (\boldsymbol{F}\boldsymbol{p}_k + \boldsymbol{v}) + t\boldsymbol{F}\boldsymbol{u}, = \boldsymbol{p}'_k + t\boldsymbol{u}'.$$

However, since both transformed lines point in the same direction $\boldsymbol{u}' = \boldsymbol{F}\boldsymbol{u}$, they are still parallel.

Problem 3: Quaternions (15 pts)

Rotate the point $\boldsymbol{p} = (1, 1, 1)$ using the rotation specified by the quaternion $q = \langle d; \boldsymbol{u} \rangle = \langle 1; 1, 1, 1 \rangle$.

- **Answer:** In short, the point is unchanged since it lies on the rotation axis. You can show this by using the definition of a unit quaternion. First, you need to make a unit quaternion and apply the definition

  $$\hat{q} = \frac{q}{\|q\|} = \frac{q}{\sqrt{4}} = < \frac{1}{2}; \frac{1}{2}, \frac{1}{2}, \frac{1}{2} > = < \cos\frac{\theta}{2}; \sin\frac{\theta}{2}\boldsymbol{v} >$$

  so that the unit axis of rotation, $\boldsymbol{v}$, is

  $$\sin\frac{\theta}{2}\boldsymbol{v} = \frac{1}{2}(1, 1, 1) \quad \Longrightarrow \quad \boldsymbol{v} = \frac{1}{\sqrt{3}}(1, 1, 1).$$

  Clearly since $\boldsymbol{p} = (1, 1, 1)$ lies on the rotation axis, the rotated version will not change under rotation, $\boldsymbol{p}' = \boldsymbol{p}$. Arguing thus was sufficient to receive full credit.

  Alternately many people proceeded directly to apply the quaternion multiplication formulas, which would also work. We can represent the point using a quaternion vector, which before and after rotation is

  $$\tilde{\boldsymbol{p}} = < 0; \boldsymbol{p} >, \quad \text{and} \quad \tilde{\boldsymbol{p}}' = < 0; \boldsymbol{p}' > .$$

  Then the quaternion rotation formula is (using the unit quaternion version for convenience)

  $$\begin{aligned} \tilde{\boldsymbol{p}}' &= \hat{q}\,\tilde{\boldsymbol{p}}\,\hat{q}^* & (7)\\ &= \left(\frac{1}{2} < 1; 1, 1, 1 >\right) < 0; \boldsymbol{p} > \left(\frac{1}{2} < 1; -1, -1, -1 >\right) & (8)\\ &= \frac{1}{4} < 1; 1, 1, 1 > < 0; 1, 1, 1 > < 1; -1, -1, -1 > & (9) \end{aligned}$$

  then using the formula for quaternion multiplication

  $$< d; \boldsymbol{u} > < d'; \boldsymbol{u}' > = < dd' - \boldsymbol{u} \cdot \boldsymbol{u}';\; d\boldsymbol{u}' + d'\boldsymbol{u} + \boldsymbol{u} \times \boldsymbol{u}' >$$

  we obtain

  $$\begin{aligned} \tilde{\boldsymbol{p}}' &= \frac{1}{4} < 1; 1, 1, 1 > < 0; 1, 1, 1 > < 1; -1, -1, -1 > & (10)\\ &= \frac{1}{4} < -3; 1, 1, 1 > < 1; -1, -1, -1 > & (11)\\ &= \frac{1}{4} < 0; 4, 4, 4 > & (12)\\ &= < 0; 1, 1, 1 > & (13)\\ &= < 0; \boldsymbol{p} > & (14) \end{aligned}$$

Problem 4: SLERP (10 pts)

When interpolating with SLERP between two unit quaternions, $\boldsymbol{x}$ and $\boldsymbol{y}$, we use:
SLERP$(\boldsymbol{x}, \boldsymbol{y}, \alpha)$, if $\boldsymbol{x} \cdot \boldsymbol{y} > 0$, and SLERP$(\boldsymbol{x}, -\boldsymbol{y}, \alpha)$ otherwise.

(i) Why is this method better than just SLERP$(\boldsymbol{x}, \boldsymbol{y}, \alpha)$? What is the difference between $+\boldsymbol{y}$ and $-\boldsymbol{y}$ here?

- **Answer:** Recall that scaling a quaternion by a nonzero constant does not change the rotation it represents, and therefore $\pm \boldsymbol{y}$ both represent the same physical rotation. The only difference is the size of the interpolation performed, which is related to the angle Ω between $\boldsymbol{x}$ and $\boldsymbol{y}$, given by $\Omega = \cos^{-1}(\boldsymbol{x} \cdot \boldsymbol{y})$. The two interpolation paths differ by which way they travel around the great circle, and the formula just takes the shorter path—as many students illustrated with a figure.

(ii) When interpolating unit normal vectors, $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$, for lighting calculations, should we also use

$\text{SLERP}(\boldsymbol{n}_1, \boldsymbol{n}_2, \alpha)$, if $\boldsymbol{n}_1 \cdot \boldsymbol{n}_2 > 0$, and $\text{SLERP}(\boldsymbol{n}_1, -\boldsymbol{n}_2, \alpha)$ otherwise?

- **Answer:** No, because, unlike quaternions for which $\pm q$ correspond to the same physical rotations, $\pm \boldsymbol{n}$ correspond to different normals for shading/lighting calculations. Using this formula would lead to interpolated normals which vary insufficiently when the two normals are at angles greater than 90^o.