CSci 6974 and ECSE 6966 Math. Tech. for Vision, Graphics and Robotics
Lecture 15&16, March 20&23, 2006
Representing and Estimating Rotations in $\mathbb{R}^3$

Preliminary Notes

- This week we will complete our discussion of the perspective camera and rotations in $\mathbb{R}^3$.
- Next week we will cover some topics in differential geometry.
- If you have the 2nd edition of Hartley&Zisserman, the chapter references in the Lecture 14 notes are off by one chapter.

Problem Overview

Problem: Given corresponding sets of point vectors $\{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ and $\{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ in $\Re^3$, find the similarity transformation that best aligns the two sets.

- This problem arises in a number of applications in photogrammetry and computer vision.
- The solution given here is based on a paper by Horn, which will be distributed in class.
- As part of the solution we will learn about *quaternions* as a means of representing rotations.
- Following the presentation of this solution, we will introduce a second technique that uses small angle approximations to solve the same problem.

Least-Squares Objective Function

The following is the least-squares objective function for the "exterior orientation" problem:

$$\sum_i \|\mathbf{p}_i - s\mathbf{R}\mathbf{q}_i - \mathbf{t}\|^2 \tag{1}$$

- s is a scale factor.
- $\mathbf{t}$ is the translation vector.
- $\mathbf{R}$ is the rotation matrix, which we will eventually represent using quaternions.

Center the Data

The first step is to calculate the center of mass of each set of points and rewrite the objective function

- Let $\overline{\mathbf{p}} = \sum_i \mathbf{p}_i / n$.
- Write $\mathbf{p}'_i = \mathbf{p}_i - \overline{\mathbf{p}}$, so that $\mathbf{p}_i = \overline{\mathbf{p}} + \mathbf{p}'_i$.
- Similarly, let $\overline{\mathbf{q}} = \sum_i \mathbf{q}_i / n$.
- Write $\mathbf{q}'_i = \mathbf{q}_i - \overline{\mathbf{q}}$, so that $\mathbf{q}_i = \overline{\mathbf{q}} + \mathbf{q}'_i$.
- The objective function can then be rewritten:

$$\sum_i \|\mathbf{p}'_i - s\mathbf{R}\mathbf{q}'_i - \mathbf{t}'\|^2 \tag{2}$$

where

$$\mathbf{t}' = \mathbf{t} - \overline{\mathbf{p}} + s\mathbf{R}\overline{\mathbf{q}}. \tag{3}$$

Solving for Translation

- Minimizing this equation requires that

$$\mathbf{t}' = \mathbf{0}. \tag{4}$$

- Therefore, after we find s and $\mathbf{R}$, we can find $\mathbf{t}$ as

$$\mathbf{t} = \overline{\mathbf{p}} - s\mathbf{R}\overline{\mathbf{q}} \tag{5}$$

- The objective function now becomes

$$\sum_i \|\mathbf{p}'_i - s\mathbf{R}\mathbf{q}'_i\|^2 \tag{6}$$

Solving for Scale

We can solve for scale separately as well:

- Expanding the objective function and using the orthonormality of $\mathbf{R}$, we get:

$$\sum_i \mathbf{p}'^{\top}_i \mathbf{p}'_i - 2s \sum_i \mathbf{p}'^{\top}_i \mathbf{R}\mathbf{q}'_i + s^2 \sum_i \mathbf{q}'^{\top}_i \mathbf{q}'_i. \tag{7}$$

- The first thing to note about this is that it is minimized with respect to $\mathbf{R}$ when

$$\sum_i \mathbf{p}'^{\top}_i \mathbf{R}\mathbf{q}'_i \tag{8}$$

is maximized. This is what we will solve to find $\mathbf{R}$.

- Now, the objective function (7) can be written

$$S_p - 2sD + s^2 S_q. \tag{9}$$

- This is clearly maximized when

$$s = D/S_q = \sum_i \mathbf{p}_i'^\top \mathbf{R}\mathbf{q}_i' \;/\; \sum_i \mathbf{q}_i'^\top \mathbf{q}_i'. \tag{10}$$

- Unfortunately, this doesn't treat the points symmetrically. Instead, we can rewrite the objective function as

$$\sum_i \|\frac{1}{\sqrt{s}}\mathbf{p}_i' - \sqrt{s}\mathbf{R}\mathbf{q}_i'\|^2 \tag{11}$$

  and solve for s to obtain

$$s = \left[\sum_i \mathbf{p}_i'^\top \mathbf{p}_i' / \sum_i \mathbf{q}_i'^\top \mathbf{q}_i'\right]^{1/2}. \tag{12}$$

  One of your practice homework problems is to derive this result and provide a geometric interpretation.

- With the revised objective function in (11), the translation becomes

$$\mathbf{t} = \frac{1}{\sqrt{s}}\overline{\mathbf{p}} - \sqrt{s}\mathbf{R}\overline{\mathbf{q}} \tag{13}$$

We are left with the problem of maximizing

$$\sum_i \mathbf{p}_i'^\top \mathbf{R}\mathbf{q}_i' \tag{14}$$

with respect to $\mathbf{R}$.

Representations of Rotation

The problem is that in 3-space $\mathbf{R}$ is a 3×3 matrix but has only 3 degrees of freedom: it must be orthonormal. Thus we have to parameterize it somehow. Many methods have been proposed in the research literature. The solution here is based on unit quaternions. Following presentation of the solution, we will describe the use of small angle approximations.

Quaternions

This is a summary and restatement of the material in Section 3 of the Horn paper.

- A *quaternion* is a 4-component vector, or a special type of complex number having 1 real part and 3 imaginary parts:

$$\dot{\mathbf{q}} = (q_0, q_x, q_y, q_z)^\top = q_0 + q_x i + q_y j + q_z k. \tag{15}$$

  In effect, we will treat quaternions as vectors, but also develop special operations on them that don't apply to vectors. The most important of these will be multiplication of two quaternions.

- For a *unit quaternion*, we must have $q_0^2 + q_x^2 + q_y^2 + q_z^2 = 1$.

Multiplication of Quaternions

- Using the rules

$$i^2 = -1, \qquad j^2 = -1, \qquad k^2 = -1, \tag{16}$$

$$ij = k, \qquad jk = i, \qquad ki = j, \tag{17}$$

$$ji = -k, \qquad kj = -i, \qquad ik = -j, \tag{18}$$

  we can create a rule for multiplication of quaternions. Note that this is an algebraic extension of the notion of a vector, which has no multiplication rule:

$$\begin{aligned} \dot{\mathbf{p}}\dot{\mathbf{q}} = {} & (p_0 q_0 - p_x q_x - p_y q_y - p_z q_z) \\ & + (p_0 q_x + p_x q_0 + p_y q_z - p_z q_y) i \\ & + (p_0 q_y - p_x q_z + p_y q_0 + p_z q_x) j \\ & + (p_0 q_z + p_x q_y - p_y q_x + p_z q_0) k \end{aligned} \tag{19}$$

- This can be written in matrix multiplication form. We have

$$\dot{\mathbf{p}}\dot{\mathbf{q}} = \begin{pmatrix} p_0 & -p_x & -p_y & -p_z \\ p_x & p_0 & -p_z & p_y \\ p_y & p_z & p_0 & -p_x \\ p_z & -p_y & p_x & p_0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{pmatrix} = \mathcal{P}\dot{\mathbf{q}} \tag{20}$$

  and

$$\dot{\mathbf{q}}\dot{\mathbf{p}} = \begin{pmatrix} p_0 & -p_x & -p_y & -p_z \\ p_x & p_0 & p_z & -p_y \\ p_y & -p_z & p_0 & p_x \\ p_z & p_y & -p_x & p_0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{pmatrix} = \tilde{\mathcal{P}}\dot{\mathbf{q}} \tag{21}$$

- Note that both $\mathcal{P}$ and $\tilde{\mathcal{P}}$ are orthogonal (orthonormal for unit quaternions).

Dot Products and Conjugates

- The dot product of two quaternions is the same as if the quaternions were treated as 4-component vectors in $\mathbb{R}^4$:

$$\dot{\mathbf{p}} \cdot \dot{\mathbf{q}} = \dot{\mathbf{p}}^\top \dot{\mathbf{q}} = p_0 q_0 + p_x q_x + p_y q_y + p_z q_z. \tag{22}$$

- Returning to the complex number analogy, the *conjugate* of a quaternion is

$$\dot{\mathbf{q}}^* = q_0 - q_x i - q_y j - q_z k. \tag{23}$$

  and it is easy to see that

$$\dot{\mathbf{q}}\dot{\mathbf{q}}^* = \dot{\mathbf{q}} \cdot \dot{\mathbf{q}}, \tag{24}$$

  i.e. the "complex" components of $\dot{\mathbf{q}}\dot{\mathbf{q}}^*$ are 0.

- We also have

$$\dot{\mathbf{p}}^* \dot{\mathbf{q}} = \begin{pmatrix} p_0 & p_x & p_y & p_z \\ -p_x & p_0 & p_z & -p_y \\ -p_y & -p_z & p_0 & p_x \\ -p_z & p_y & -p_x & p_0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{pmatrix} = \mathcal{P}^\top \dot{\mathbf{q}} \tag{25}$$

  and, similarly,

$$\dot{\mathbf{q}}\dot{\mathbf{p}}^* = \tilde{\mathcal{P}}^\top \dot{\mathbf{q}} \tag{26}$$

- Finally, combining results on the dot product with the orthogonality of $\mathcal{P}$ produces:

$$\mathcal{P}\tilde{\mathcal{P}} = (\dot{\mathbf{p}} \cdot \dot{\mathbf{p}})\mathbf{I}. \tag{27}$$

Other Properties

- Using the matrix multiplication description of quaternion multiplication, it is fairly easy to see that

$$(\dot{\mathbf{r}}\dot{\mathbf{p}}) \cdot (\dot{\mathbf{r}}\dot{\mathbf{q}}) = (\dot{\mathbf{r}} \cdot \dot{\mathbf{r}})(\dot{\mathbf{p}} \cdot \dot{\mathbf{q}}). \tag{28}$$

  This is just $\dot{\mathbf{p}} \cdot \dot{\mathbf{q}}$ when $\dot{\mathbf{r}}$ is a unit quaternion.

- We also have the rather bizarre looking

$$(\dot{\mathbf{p}}\dot{\mathbf{q}}) \cdot \dot{\mathbf{r}} = \dot{\mathbf{p}} \cdot (\dot{\mathbf{r}}\dot{\mathbf{q}}^*). \tag{29}$$

  This can also be shown using the matrix multiplication description of quaternion multiplication and the properties of the conjugate.

3-Space Vectors and Rotations

So far we haven't shown anything about rotations and solving our original problem. We're almost there, though.

- A 3-space vector $\mathbf{p} = (p_x, p_y, p_z)^\top$ may be represented as a "purely imaginary" (don't you love it!) quaternion:

$$\dot{\mathbf{p}} = (0, p_x, p_y, p_z)^\top. \tag{30}$$

- Heading toward rotations, one of the major properties of a rotation matrix is length (and dot product) preservation. With unit quaternions, we already have $(\dot{\mathbf{r}}\dot{\mathbf{p}}) \cdot (\dot{\mathbf{r}}\dot{\mathbf{q}}) = \dot{\mathbf{p}} \cdot \dot{\mathbf{q}}$. This is getting us close.

- It turns out that we need to use two multiplications to represent a rotation:

$$\dot{\mathbf{p}} = \dot{\mathbf{r}}\dot{\mathbf{q}}\dot{\mathbf{r}}^* \tag{31}$$

  is the rotation of $\dot{\mathbf{q}}$ into $\dot{\mathbf{p}}$. Here, we assume that $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$ represent 3-space vectors and $\dot{\mathbf{r}}$ is a unit quaternion.

- To show this, note that

$$\dot{\mathbf{r}}\dot{\mathbf{q}}\dot{\mathbf{r}}^* = \big(\mathcal{R}\dot{\mathbf{q}}\big)\dot{\mathbf{r}}^* = \tilde{\mathcal{R}}^\top \mathcal{R}\dot{\mathbf{q}}, \tag{32}$$

  and recall that both $\mathcal{R}$ and $\tilde{\mathcal{R}}^\top$ are both orthonormal.

- Writing out $\tilde{\mathcal{R}}^\top\mathcal{R}$ gives

$$\tilde{\mathcal{R}}^\top\mathcal{R} = \begin{pmatrix} \dot{\mathbf{r}}\cdot\dot{\mathbf{r}} & 0 & 0 & 0 \\ 0 & (r_0^2 + r_x^2 - r_y^2 - r_z^2) & 2(r_xr_y - r_0r_z) & 2(r_xr_z + r_0r_y) \\ 0 & 2(r_xr_y + r_0r_z) & (r_0^2 - r_x^2 + r_y^2 - r_z^2) & 2(r_yr_z - r_0r_x) \\ 0 & 2(r_xr_z - r_0r_y) & 2(r_yr_z + r_0r_x) & 2(r_0^2 - r_x^2 - r_y^2 + r_z^2). \end{pmatrix} \tag{33}$$

  The lower 3×3 submatrix is exactly the rotation matrix needed when considering $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$ as vectors. Also, multiplication by the whole 4×4 matrix preserves the "pure imaginary" property of representing 3-space vectors with quaternions.

- Finally, it can be shown that for unit quaternion $\dot{\mathbf{r}}$ representing the rotation,

$$r_0 = \cos\Big(\frac{\theta}{2}\Big) \qquad \text{and} \qquad (r_x, r_y, r_z)^\top = \sin\Big(\frac{\theta}{2}\Big)\hat{\omega}, \tag{34}$$

  where θ is the angle of rotation and $\hat{\omega}$ is the axis of rotation.

Finding the Rotation

Now we can return to the problem of finding the rotation that maximizes

$$\sum_i \mathbf{p}_i'^\top \mathbf{R} \mathbf{q}_i'. \tag{35}$$

This is the rotation that minimizes our original objective function.

- Writing in terms of quaternions, we have

$$\sum_i \dot{\mathbf{p}}_i' \cdot (\dot{\mathbf{r}} \mathbf{q}_i' \dot{\mathbf{r}}^*). \tag{36}$$

- Manipulating using equation 29,

$$\begin{aligned}
\sum_i \dot{\mathbf{p}}_i' \cdot (\dot{\mathbf{r}} \dot{\mathbf{q}}_i' \dot{\mathbf{r}}^*) &= \sum_i (\dot{\mathbf{p}}_i' \dot{\mathbf{r}}) \cdot (\dot{\mathbf{r}} \dot{\mathbf{q}}_i') \\
&= \sum_i (\mathcal{P}_i' \dot{\mathbf{r}}) \cdot (\tilde{\mathcal{Q}}_i' \dot{\mathbf{r}}) \\
&= \dot{\mathbf{r}}^\top \left[\sum_i \mathcal{P}_i'^\top \tilde{\mathcal{Q}}_i' \right] \dot{\mathbf{r}} \\
&= \dot{\mathbf{r}}^\top \mathbf{M} \dot{\mathbf{r}}
\end{aligned}$$

- Maximizing this subject to the constraint that $\dot{\mathbf{r}}$ is a unit vector is a problem we know how to solve!

Small Angle Approximations — Overview

We will introduce small angle approximation to solve the problem of minimizing

$$\sum_i \|\mathbf{p}_i - \mathbf{R}\mathbf{q}_i\|^2 \tag{37}$$

(I've dropped the **'** on $\mathbf{p}_i$ and $\mathbf{q}_i$ to simplify notation.) Above we found that this was equivalent to maximizing (14),

$$\sum_i \mathbf{p}_i{}^\top \mathbf{R}\mathbf{q}_i, \tag{38}$$

but here we stick with the original form in using a second representation of rotation — the small angle approximation.

More on Rotations

- Any rotation can be obtained as a composition of rotations about the X, Y, and Z axes. These axis rotations can be written

$$\mathbf{R}_X = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}, \qquad \mathbf{R}_Y = \begin{pmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{pmatrix}, \tag{39}$$

  and

$$\mathbf{R}_Z = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{40}$$

  where θ, ϕ, and ψ are the angles of rotation.

- Any rotation matrix $\mathbf{R}$ can be written as

$$\mathbf{R} = \mathbf{R}_Z \mathbf{R}_Y \mathbf{R}_X \tag{41}$$

  for some appropriate choice of rotation angles.

- It is important to observe that by reordering the multiplications, e.g. $\mathbf{R}_Z \mathbf{R}_X \mathbf{R}_Y$, the rotation matrix changes (in general).

The Approximation

- The power of using these angles comes when approximating $\mathbf{R}$ assuming small angles.

- Recall that the Taylor series expansions of the sine and cosine functions are

$$\cos t = 1 - \frac{t^2}{2} + \frac{t^4}{4!} - \ldots$$
$$\sin t = t - \frac{t^3}{3!} + \frac{t^5}{5!} - \ldots$$

Using a 1^{st} order approximation, $\cos t = 1$ and $\sin t = t$. Also, note that in a 1^{st} order approximation, the Taylor series expansion produces $\sin u \sin v = 0$.

- We can use these facts to show that up to a 1^{st} (small angle) approximation,

$$\mathbf{R} = \begin{pmatrix} 1 & -\psi & \phi \\ \psi & 1 & -\theta \\ -\phi & \theta & 1 \end{pmatrix} \tag{42}$$

Note that this matrix is **not** orthogonal.

Solving the Rotation Problem Using the Approximation

- Substituting the approximation into (37) yields

$$\sum_i \|\mathbf{p}_i - \begin{pmatrix} 1 & -\psi & \phi \\ \psi & 1 & -\theta \\ -\phi & \theta & 1 \end{pmatrix} \mathbf{q}_i\|^2 \tag{43}$$

- This may easily be rearranged into

$$\sum_i \|\mathbf{p}_i - \mathbf{q}_i - \mathbf{Q}_i \mathbf{r}\|^2 \tag{44}$$

where $\mathbf{r}^\top = (\theta, \phi, \psi)$ and

$$\mathbf{Q}_i = \begin{pmatrix} 0 & q_{i,z} & -q_{i,y} \\ -q_{i,z} & 0 & q_{i,x} \\ q_{i,y} & -q_{i,x} & 0 \end{pmatrix} \tag{45}$$

for $\mathbf{q}_i^\top = (q_{i,x}, q_{i,y}, q_{i,z})$.

- This is a straight-forward, unconstrained, least-squares regression problem.

- Substituting the resulting angles back into

$$\begin{pmatrix} 1 & -\psi & \phi \\ \psi & 1 & -\theta \\ -\phi & \theta & 1 \end{pmatrix} \tag{46}$$

does not yield a rotation matrix. We still need to convert this matrix, which we will denote by $\mathbf{R}^*$, into a rotation.

- We do this using (once again!) the SVD. If

$$\mathbf{R}^* = \mathbf{U}\mathbf{W}\mathbf{V}^\top \tag{47}$$

is the SVD of $\mathbf{R}^*$, then we can show that

$$\mathbf{R} = \mathbf{U}\mathbf{I}\mathbf{V}^\top \tag{48}$$

is the closest rotation matrix to $\mathbf{R}^*$ in a Frobenius norm (least-squares distance between the individual components of the matrices) sense.

- The correction from $\mathbf{R}^*$ to $\mathbf{R}$ is larger for large angles. Therefore, sometimes it is necessary to estimate an incremental rotation from

$$\sum_i \|\mathbf{p}_i - \Delta\mathbf{R} \cdot \mathbf{R}\mathbf{q}_i\|^2 = \sum_i \|\mathbf{p}_i - \Delta\mathbf{R}\mathbf{q}_i{}'\|^2 \tag{49}$$

where here $\mathbf{q}_i' = \mathbf{R}\mathbf{q}_i$.

- The new estimate is

$$\mathbf{R} = \Delta\mathbf{R} \cdot \mathbf{R}. \tag{50}$$

- These iterations can be repeated as many times as desired, but more than once or two is rarely necessary to obtain sufficient accuracy.

Discussion

- The quaternion solution is straight-forward, simple and elegant, but it depends heavily on the ability to separate translation and scale from rotation.
- The small-angle solution is just as straight-forward, but not quite as elegant. It works much better, however, for problems where separability is not possible.

Practice Problems / Potential Test Questions

1. Prove that Equation (2) is minimized with respect to $\mathbf{t}'$ when $\mathbf{t}' = \mathbf{0}$.
2. Give a derivation of Equation (12) from (11) and from provide a geometric interpretation that intuitively justifies the resulting scale estimate.
3. Prove that

$$\dot{\mathbf{q}}\dot{\mathbf{q}}^* = \begin{pmatrix} \dot{\mathbf{q}} \cdot \dot{\mathbf{q}} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

4. Prove the relation in Equation (29).
5. Show that in the small angle approximation

$$\mathbf{R}_X\mathbf{R}_Y = \mathbf{R}_Y\mathbf{R}_X,$$

but that this is not true in general.

6. The Frobenius norm of a matrix is the square root of the sum of the squares of the matrix entries. In other words, if

$$\mathbf{A} = (a_{i,j})$$

then the Frobenius norm of $\mathbf{A}$ is

$$\|\mathbf{A}\|_F = \left(\sum_{i,j} a_{i,j}^2 \right)^{1/2}.$$

The Frobenius norm distance between two matrices, $\mathbf{A}$ and $\mathbf{B}$, of the same dimensions is

$$\|\mathbf{A} - \mathbf{B}\|_F.$$

Let $\mathbf{A}$ be a square matrix and let the SVD of $\mathbf{A}$ be $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$. Show that the matrix

$$\mathbf{B} = \mathbf{U}\mathbf{V}^\top$$

is the closest orthogonal matrix to $\mathbf{A}$ in the Frobenius norm.

Homework 8 Problem

The solution to the following programming problem, together with solutions to the Homework 7 problems listed at the end of the Lecture 14 notes, is due at the start of class on Monday, March 27. Each of the two parts of the problem is worth 20 points toward your homework grade.

Please write Matlab or C++ programs to compute the best least-squares estimate to the rotation matrix aligning two sets of N points, $\mathcal{A}$ and $\mathcal{B}$ — i.e. the rotation from the coordinate system of $\mathcal{A}$ to the coordinate system of $\mathcal{B}$. For the first part, use unit quaternions and for the second part use the small-angle approximation. The input will be a file containing N lines of numbers. Each line will have 6 floating point numbers, with the first 3 numbers being the coordinates of a point from $\mathcal{A}$ and the last 3 numbers being the coordinates of the corresponding point from $\mathcal{B}$.

In your solutions, you may only use vectors, matrices, the spectral decomposition, and the SVD. You may not use any special quaternion implementations. For each of your two solutions, turn in the source code and the results/output on the data sets posted (soon) on the course web site. Both code and output should be clearly documented. The output should contain both the computed rotation and an analysis of the remaining error following application of the rotation. In particular, if $\mathbf{R}$ is the estimated rotation, represented as a matrix, then the error is

$$\left[\sum_{i=1}^{N} \|\mathbf{b}_i - \mathbf{R}\mathbf{a}_i\|^2 / N \right]^{1/2}.$$