Lecture 9: Registration

Allison Okamura
Stanford University
Updates

Assignment 3 - Due today, 4 pm

Assignment 4 - Posted today

For Assignment 4, sign up for teams/ultrasound after 11 am today at:

http://tinyurl.com/2016ultrasound

Intuitive Surgical Tour: Friday, Nov. 11 (leaving at 2 pm)

31 people signed up so far. This evening I will open the doodle poll (http://doodle.com/poll/upmkmwraamtmsu3sv) to students in ME/CS 571. Sign up before then if you want to come (max 40 attendees). Thanks to all the drivers!
registration

the alignment of multiple data sets into a single coordinate system such that the spatial locations of corresponding points coincide

i.e., establish a quantitative relationship between different reference frames

essential for the quantitative integration of various sources of information:
1. pre-, intra-, and post-operative images
2. trackers (tracking data not useful unless registered)
3. robots (cannot use images unless registered)

examples
suitability for image-guided procedures

- **Accuracy**: Target registration error (TRE) quantifies how far the predicted position of the anatomical target is from its actual position. Typically satisfactory at ~1 mm.

- **Speed**: How long does it take the algorithm to produce the solution intraoperatively, need on the order of seconds to minutes.

- **Robustness**: How well does the algorithm deal with noise and outliers, depends on how noisy the data is.
navigation and registration example

http://www.youtube.com/watch?v=jYCiKOERYD8
http://www.youtube.com/watch?v=c58etXWWoDA
a typical robot registration problem

Preoperative model from a CT scan gives the location of a tumor in the brain...

Now we are in the OR and want to operate with a robot based on this preoperative image

registration example and graphics provided by Russell H. Taylor, JHU
a typical robot registration problem

What is $T_{\text{reg}}$?
coordinate systems and transformations

e.g., robot base coordinate system

e.g., image or patient coordinate system
coordinate systems and transformations

\[ v_A = F(v_B) \]
\[ v_A = T v_B \]
\[ v_A = R v_B + p \]

\[ T(\mathbf{R}, \mathbf{p}) = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{p}_{3 \times 1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

e.g., robot base coordinate system

e.g., image or patient coordinate system
coordinate systems and transformations

$$v_A = F(v_B)$$

$$v_A = Tv_B$$

$$v_A = Rv_B + p$$

$$T(R, p) = \begin{bmatrix} R_{3 \times 3} & p_{3 \times 1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
coordinate systems and transformations

\[ v_A = F(v_B) \]
\[ v_A = T v_B \]
\[ v_A = R v_B + p \]

then translate

\[
T(R, p) = \begin{bmatrix}
R_{3 \times 3} & p_{3 \times 1} \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
interesting properties of $T$ and $R$

forward transformation

$$v_A = T v_B$$
$$v_A = R v_B + p$$

inverse transformation

$$T^{-1} v_A = v_B$$
$$v_B = R^{-1} (v_A - p)$$
$$v_B = R^{-1} v_A - R^{-1} p$$

composition of transformations

$$T_3 = T_1 T_2$$
$$R_3 = R_1 R_2$$
$$p_3 = R_1 p_2 + p_1$$

rotation matrices are orthogonal matrices

$$R^T = R^{-1}$$
$$R^T R = RR^T = I$$

and members of the group $\text{SO}(3)$

$$\det(R) = +1$$
composition of transformations

\[ F_{GH} = F_{BG}^{-1} F_{BE} F_{EH} \]
feature-based registration

preoperative model

intraoperative reality

tracked pointer/robot end-effector
features used for feature-based registration

- Point fiducials (markers)
- Point anatomical landmarks
- Ridge curves
- Contours
- Surfaces
- Line fiducials

Given two features (either the same features in two different reference frames, or two different features in the same reference frame), we need to know the “distance” between them.
what the computer knows

preoperative model

intraoperative reality
identify corresponding points

preoperative model  intraoperative reality

\[\vec{b}_1, \vec{b}_2, \vec{b}_3\]  \[\vec{a}_1, \vec{a}_2, \vec{a}_3\]
how to identify correspondences

manual

vs.

part of the optimization/minimization process

state of the art is the “iterative closest point” (ICP) algorithm

... but we will assume manual is possible
find the best rigid transformation

preoperative model

\[ \begin{align*}
\min_{T_{\text{reg}}} & \sum_{i} w_i D(T_{\text{reg}}a_i, b_i) \\
\text{minimize over all possible values of } T_{\text{reg}}
\end{align*} \]

intraoperative reality

weighting depending on which points are being considered

\[ \text{a disparity function} \]
disparity function, $D$

a metric for the error between two feature sets

$$\min_{T_{\text{reg}}} \sum_{i} w_i D(T_{\text{reg}} a_i, b_i)$$

minimize over all possible values of $T_{\text{reg}}$

weighting depending on which points are being considered

sum of squares of residuals is common:  $$\min_{T_{\text{reg}}} \sum_{i} w_i \| (T_{\text{reg}} a_i - b_i) \|^2$$

other $D$ possibilities include:
- maximum distance
- median distance
- cardinality depending in threshold
how to do the optimization for rigid registration

Given points in two different coordinate systems (e.g., a set of points \( \{a_i\} \) and a set of points \( \{b_i\} \))

Find the transformation matrix \( T(R, p) \)

That minimizes \( \sum_i e_i^T e_i \)

Where \( e_i = (Ra_i + p) - b_i \)

this is tricky because of \( R \)

Options:
• global vs. local
• numerical vs. direct (analytical)
• ways of dealing with local minima
minimizing registration errors

Step 1: compute means and residuals of known points

\[ \bar{a} = \frac{1}{N} \sum_{i=1}^{N} a_i \]

\[ b = \frac{1}{N} \sum_{i=1}^{N} b_i \]

\[ \tilde{a}_i = a_i - \bar{a} \]

\[ \tilde{b}_i = b_i - \bar{b} \]

Step 2: Find \( R \) that minimizes \( \sum_i (R\tilde{a}_i - \tilde{b}_i)^2 \)

\[ p = \bar{b} - R\bar{a} \]

Step 4: Desired transformation is

\[ T(R, p) = \begin{bmatrix} R & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \]
minimization substeps: direct method
(there is also an iterative method)

Step 1: Compute

\[ H = \sum_i \begin{bmatrix}
\tilde{a}_i \tilde{b}_i, x & \tilde{a}_i \tilde{b}_i, y & \tilde{a}_i \tilde{b}_i, z \\
\tilde{a}_i \tilde{b}_i, x & \tilde{a}_i \tilde{b}_i, y & \tilde{a}_i \tilde{b}_i, z \\
\tilde{a}_i \tilde{b}_i, x & \tilde{a}_i \tilde{b}_i, y & \tilde{a}_i \tilde{b}_i, z \\
\end{bmatrix} \]

outer product between a and b

Step 2: Compute the SVD of \( H = USV^t \)

Step 3: \( R = VU^t \)

Step 4: Verify \( Det(R) = 1 \). If not, then algorithm may fail.

Extra Slides:
An Iterative Method for Finding the Rotation Matrix

(I don’t recommend using this, though. The SVD method is much simpler.)
iterative method: solving for $R$

Goal: given paired point sets $\{a_i\}$ and $\{b_i\}$, find

$$R = \arg \min \sum_i (R\tilde{a}_i - \tilde{b}_i)^2$$

Step 0: Make an initial guess $R_0$

Step 1: Given $R_k$, compute $\hat{b}_i = R_k^{-1}\tilde{b}_i$

Step 2: Compute $\Delta R$ that minimizes $\sum_i (\Delta R\tilde{a}_i - \hat{b}_i)^2$

Step 3: Set $R_{k+1} = R_k \Delta R$

Step 4: Iterate Steps 1-3 until residual error is sufficiently small (or other termination condition)
more mathematical preliminaries

matrix representation of cross product

\[ a = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T \]

\[ \text{skew}(a) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \]

\[ a \times v = \text{skew}(a)v \]

representations/constructions of rotation matrices

angle-axis: \( \text{rot}(a, \alpha) \) is a rotation \( \alpha \) about axis \( a \)

Rodrigues’ formula: \( c = \text{rot}(a, \alpha)b = b \cos(\alpha) + a \times b \sin(\alpha) + a(a^Tb)(1 - \cos(\alpha)) \)

exponential: \( \text{rot}(a, \alpha) = e^{\text{skew}(a)\alpha} = I + \alpha \text{skew}(a) + \frac{\alpha^2}{2!}\text{skew}(a)^2 + \ldots \)

\( \text{rot}(a, \alpha) \approx I + \alpha \text{skew}(a) \)

more notation: \( \text{rot}(a, \alpha) = R_a(\alpha) \)

\( R(a) = \text{rot}(a, ||a||) \)
“small” transformations
useful for linear approximations to
represent small pose shifts $\Delta T v = \Delta R v + \Delta p$

$\Delta R$ a small rotation

$R_a(\Delta \alpha)$ a rotation by a small angle $\Delta \alpha$ about axis $a$

$\text{rot}(a, ||a||) b \approx a \times b + b$ for $||a||$ sufficiently small

$\Delta R(a)$ a rotation that is small enough so that any error introduced by this approximation is negligible

$\Delta R(\lambda a) \Delta R(\mu b) \approx \Delta R(\lambda a + \mu b)$ linearity for small rotations

exercise: work out the linearity proposition by substitution
“small” transformations

\[ \Delta T v = \Delta R(a) v + \Delta p \]

\[ \Delta T v \approx v + a \times v + \Delta p \]

\[ a \times v = \text{skew}(a)v = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \]

\[ \text{skew}(a)a = 0 \]

\[ \Delta R(a) \approx I + \text{skew}(a) \]

\[ \Delta R(a)^{-1} \approx I - \text{skew}(a) = I + \text{skew}(-a) \]
iterative method: solving for $R$

Goal: given paired point sets $\{a_i\}$ and $\{b_i\}$, find

$$R = \arg \min \sum_i (R\tilde{a}_i - \tilde{b}_i)^2$$

Step 0: Make an initial guess $R_0$

Step 1: Given $R_k$, compute

$$\hat{b}_i = R_k^{-1}\tilde{b}_i$$

Step 2: Compute $\Delta R$ that minimizes

$$\sum_i (\Delta R\tilde{a}_i - \hat{b}_i)^2$$

Step 3: Set $R_{k+1} = R_k \Delta R$

Step 4: Iterate Steps 1-3 until residual error is sufficiently small (or other termination condition)
iterative method: solving for $\Delta R$

Approximate $\Delta R$ as $(I + \text{skew}(\bar{\alpha}))$

Which is equivalent to $\Delta Rv \approx v + \bar{\alpha} \times v$

remember: multiplying by a skew-symmetric matrix is equivalent to taking cross product

Then our least squares problem becomes

$$\min_{\Delta R} \sum_i (\Delta R \tilde{a}_i - \hat{b}_i)^2 \approx \min_{\bar{\alpha}} \sum_i (\tilde{a}_i - \hat{b}_i + \bar{\alpha} \times \tilde{a}_i)^2$$

This is a linear least squares problem in $\bar{\alpha}$

Then compute $\Delta R(\bar{\alpha})$