Calibration

600.445/446 Computer Integrated Surgery

Calibrate (vt): 1. to determine the caliber of (as a thermometer tube); 2. to determine, rectify, or mark the gradations of (as a thermometer tube); 3. to standardize (as a measuring instrument) by determining the deviation from a standard so as to ascertain the proper correction factors; 4. ADJUST, TUNE
Calibration

Reality

Measuring device

Model of Reality

Error

Calibration

Reality

Actuation device

Model of Reality

Error
Basic Techniques

- Parameter Estimation
- Mapping the space

Parameter Estimation

- Compare observed system performance to reference standard ("ground truth")
- Compute parameters of mathematical model that minimizes residual error.
Pointing device calibration

\[ F_i = (R_i, p_i) \]
Pointing device calibration

\[ F_i = (R_i, p_i) \]
Pointing device calibration

\[ F_i = (R_i, p_i) \]

Parameter estimation

Typically, try to find the minimum of a convex function such as

\[ q^* = \arg \min_q E \left\{ f(\tilde{x}, q), \tilde{p}, q \right\} \]

for a function \( f(\tilde{x}; q) \) and observations \( \tilde{p}_k = \tilde{f}(\tilde{x}_k) \)

There are many methods for solving this problem. You can consult any good numerical methods text, such as Numerical Methods in C / C++ / xyz.

Most often \( E \left\{ f(\tilde{x}_k; q), \tilde{p}_k \right\} \) is a sum of squares

\[ E \left( f(\tilde{x}_k; q), \tilde{p}_k \right) = \sum_k \left| f(\tilde{x}_k; q) - \tilde{p}_k \right|^2 \]

However, other functions are also used, e.g.

\[ E \left( f(\tilde{x}_k; q), \tilde{p}_k \right) = \sum_k \left| f(\tilde{x}_k; q) - \tilde{p}_k \right|_1 \]
Linear Parameter Estimation

\[ p_{norm} = f(q) \quad \text{where} \quad q = [q_1, ..., q_n]^T \quad \text{are parameters} \]

\[ p^* = f(q + \Delta q) \approx f(q) + \sum \frac{\partial f}{\partial q_j}(q) \Delta q_j \]

\[ \equiv f(q) + J(q)\Delta q \]

Parameter estimation: least squares adjustment

Generally, these are iterative methods. One typical example is:

\[ q^* = \arg \min \sum (f(x_i, q) - p_i)^2 \]

Step 0 Make an initial guess \( q^{(0)} \) of the parameter vector \( q \). Set \( i \leftarrow 0 \).

Step 1 Solve the least squares problem

\[ \begin{bmatrix} J_i(q^{(i)}) \end{bmatrix} \Delta q^{(i+1)} \approx \begin{bmatrix} p_i^* - f(q^{(i)}) \end{bmatrix} \]

to find \( \Delta q^{(i+1)} \)

Step 2 \( q^{(i+1)} \leftarrow q^{(i)} + \Delta q^{(i+1)} \); evaluate \( \{ e_i \leftarrow p_i^* - f(q^{(i+1)}) \}; \quad \{ \zeta^{(i+1)} \leftarrow \sum e_i e_i^T \} \)

Step 3 If \( \zeta^{(i+1)} \) is small enough, or otherwise converged, then stop.

Else set \( i \leftarrow i + 1 \) and go back to Step 1.
Linear Least Squares

- Most commonly used method for parameter estimation
- Many numerical libraries
- See the web site
- Here is a quick review

Example: 2 link robot calibration
Example: 2 link robot calibration

$$p = \begin{bmatrix} a_1 \sin \theta_1 + a_2 \sin(\theta_{12}) \\ 0 \\ a_1 \cos \theta_1 + a_2 \cos(\theta_{12}) \end{bmatrix}$$

where $\theta_{12} = \theta_1 + \theta_2$

$$p' = \begin{bmatrix} (a_1 + \Delta a_1)\sin(\theta_1 + \Delta \theta_1) + (a_2 + \Delta a_2)\sin(\theta_{12} + \Delta \theta_1 + \Delta \theta_2) \\ 0 \\ (a_1 + \Delta a_1)\cos(\theta_1 + \Delta \theta_1) + (a_2 + \Delta a_2)\cos(\theta_{12} + \Delta \theta_1 + \Delta \theta_2) \end{bmatrix}$$
Example: 2 link robot calibration

\[ p_k = f(q_k) = f(a_1, a_2, \theta_1, \theta_2) = \begin{bmatrix} a_1 \sin \theta_{1k} + a_2 \sin(\theta_{12k}) \\ 0 \\ a_1 \cos \theta_{1k} + a_2 \cos(\theta_{12k}) \end{bmatrix} \]  

where \( \theta_{12} = \theta_1 + \theta_2 \)

so we solve the least squares problem

\[
\begin{bmatrix}
\vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial f}{\partial a_1}(q_k) & \frac{\partial f}{\partial a_2}(q_k) & \frac{\partial f}{\partial \theta_1}(q_k) & \frac{\partial f}{\partial \theta_2}(q_k) & \Delta a_1 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial f}{\partial a_1}(q_k) & \frac{\partial f}{\partial a_2}(q_k) & \frac{\partial f}{\partial \theta_1}(q_k) & \frac{\partial f}{\partial \theta_2}(q_k) & \Delta a_2 \\
\Delta \theta_1 & \Delta \theta_2 & \vdots & \vdots \\
\end{bmatrix} \approx \begin{bmatrix}
\vdots \\
p_k - a_1 \text{Rot}(y, \theta_{1k}) \\
\vdots \\
+ a_2 \text{Rot}(y, \theta_{2k}) \\
\end{bmatrix}
\]

Example: 2 link robot calibration

Here

\[ J_l(q_k) = \begin{bmatrix}
\sin \theta_{1k} & \sin \theta_{12k} & a_1(\cos \theta_{1k} + \cos \theta_{12k}) & a_2 \cos \theta_{12k} \\
0 & 0 & 0 & 0 \\
\cos \theta_{1k} & \cos \theta_{12k} & -a_1(\sin \theta_{1k} + \sin \theta_{12k}) & -a_2 \cos \theta_{12k} \\
\end{bmatrix} \]

so

\[
\begin{bmatrix}
\sin \theta_{1k} & \sin \theta_{12k} & a_1(\cos \theta_{1k} + \cos \theta_{12k}) & a_2 \cos \theta_{12k} \\
\cos \theta_{1k} & \cos \theta_{12k} & -a_1(\sin \theta_{1k} + \sin \theta_{12k}) & -a_2 \cos \theta_{12k} \\
\vdots & \vdots & \vdots & \vdots \\
\end{bmatrix} \begin{bmatrix}
\Delta a_1 \\
\Delta a_2 \\
\Delta \theta_1 \\
\Delta \theta_2 \\
\end{bmatrix} \approx \begin{bmatrix}
\Delta x' - a_1 \sin \theta_{1k} + a_2 \sin(\theta_{12k}) \\
\Delta z' - a_1 \cos \theta_{1k} + a_2 \cos(\theta_{12k}) \\
\vdots \\
\end{bmatrix}
\]
Example: Undistorted fluoroscope calibration
Calibration if no distortion (version 1)

Assume no distortion. For the moment also assume that you have N point calibration features (e.g., small steel balls) at known positions \( \{a_0, \cdots, a_{N-1}\} \) relative to the detector. Assume further that the points create images at corresponding points \( \{d_0, \cdots, d_{N-1}\} \) on the detector. Estimate the position \( s \) of the x-ray source relative to the detector.
Projection of a point feature

\[ s = \lambda (a - d) + d \]
\[ \lambda = \frac{(a - d) \cdot (s - d)}{(a - d) \cdot (a - d)} \]
\[ d = \mu (a - s) + s \]
\[ \mu = \frac{(a - s) \cdot (d - s)}{(a - s) \cdot (a - s)} \]

Approach

\[ (a - d) \times (s - d) = 0 \]
\[ \text{skew}(a - d) \cdot s = (a - d) \times d = a \times d - d \times d = a \times d \]

\[
\begin{bmatrix}
0 & d_z - a_z & a_y - d_y \\
(a_z - d_z) & 0 & d_x - a_x \\
(d_y - a_y) & (a_x - d_x) & 0
\end{bmatrix} \]

\[ s = a \times d \]
Approach

Solve least squares problem

\[
\begin{bmatrix}
\text{skew}(a_0 - d_0) \\
\vdots \\
\text{skew}(a_{N-1} - d_{N-1})
\end{bmatrix}
\begin{bmatrix}
s_x \\
s_y \\
s_z
\end{bmatrix}
\cong
\begin{bmatrix}
a_0 	imes d_0 \\
\vdots \\
a_{N-1} 	imes d_{N-1}
\end{bmatrix}
\]

Typical fiducial objects

Points

Curves

FTRAC
(Jain et al.)
What if pose of calibration object is imprecisely known?

- This is a hairier problem, but solvable
- In fact, it makes a great homework assignment ….

Calibrating trackers to robots and similar “AX = XB” problems
**Problem:** How do you determine $F_{rob}$ and $F_{wc}$?

To simplify this situation, define

$$F_{hw,k} = F_{h,k} F_{wns,t,k}$$

This gives

$$F_{hc,k} = F_{hw,k} F_{wc} = F_{rob} F_{rc,k}$$
Move the robot to a sequence of poses \( F_{rc,k} \) for \( k = 0, \ldots, N \) and observe the corresponding values of \( F_{hw,k} \). Define

\[
A_k = F_{hw,k} F_{hw,0}^{-1} \quad \text{for} \quad k = 1, \ldots, N
\]

\[
B_k = F_{rc,k} F_{rc,0}^{-1} \quad \text{for} \quad k = 1, \ldots, N
\]

\( A_0 \) represents motion from an initial pose \( A_{F_{hw,0}} = F_{hw,0} \)

so

\[
A_k = F_{hw,k} F_{hw,0}^{-1}
\]

and similarly for \( B_k \).
Problems of this form are often referred to as “AX=XB” problems.
Solving “AX = XB” problems where X is a rigid transformation

Given known frame transformations \( \{F_{A,k}, F_{B,k}\} \) we want to find a best estimate \( F_x = [R_x, \tilde{p}_x] \) such that \( F_{A,k} \cdot F_x \approx F_x \cdot F_{B,k} \). This is equivalent to

\[
R_{A,k} R_x = R_x R_{B,k} \\
R_{A,k} \tilde{p}_x + \tilde{p}_{A,k} \approx R_x \tilde{p}_{B,k} + \tilde{p}_x
\]

We will solve first for the rotation part and then for the translation part.

Rotation Part (less good way)

Note: The quaternion method (discussed next) is better

We want to solve

\[ R_{A,k} R_x = R_x R_{B,k} \]

Using the notation

\[ R_A = \text{Rot}(\tilde{\alpha}) = \text{Rot}(\frac{\tilde{\alpha}}{\|\tilde{\alpha}\|}) = \text{Rot}(\tilde{n}_A, \theta_A) \]

etc., we recall that

\[ R_A R_x = \text{Rot}(\tilde{n}_A, \theta_A) R_x = R_x \text{Rot}(R_x^{-1} \tilde{n}_A, \theta_A) \]

So

\[ R_x \text{ Rot}(R_x^{-1} \tilde{n}_A, \theta_A) = R_x \text{ Rot}(\tilde{n}_B, \theta_B) \]
Rotation Part (less good way), continued

From previous slide
\[ R_x \text{Rot}(R_x^{-1} \hat{n}_A, \theta_A) = R_x \text{Rot}(\hat{n}_B, \theta_B) \]

Multiplying both sides by by \( R_x^{-1} \) gives
\[ \text{Rot}(R_x^{-1} \hat{n}_A, \theta_A) = \text{Rot}(\hat{n}_B, \theta_B) \]

This can be expressed as
\[ R_x^{-1} \hat{\alpha} = \hat{\beta} \]

where \( \hat{\alpha} = \theta_A \hat{n}_A \) and \( \hat{\beta} = \theta_B \hat{n}_B \). Rearranging and inserting subscripts gives a system
\[ R_x \hat{\beta}_k = \hat{\alpha}_k \]

which can be solved for \( R_x \) by standard rigid rotation estimation methods.

---

Rotation Part (with quaternions)

Let \( q_x = s_x + \vec{v}_x \) be the unit quaternion corresponding to \( R_x \), with similar definitions for \( q_A \) and \( q_B \). Then we have for \( R_A R_x = R_x R_B \)

\[ q_x q_X = q_X q_B \]

Expanding the scalar and vector parts gives
\[ s_A s_X - \vec{v}_A \cdot \vec{v}_X = s_x s_B - \vec{v}_X \cdot \vec{v}_B \]
\[ s_A \vec{v}_X + s_x \vec{v}_A + \vec{v}_A \times \vec{v}_X = s_x \vec{v}_B + s_B \vec{v}_X + \vec{v}_X \times \vec{v}_B \]

Rearranging ... 
\[ (s_A - s_B) s_X - (\vec{v}_A - \vec{v}_B) \cdot \vec{v}_X = 0 \]
\[ (\vec{v}_A - \vec{v}_B) s_X + (s_A - s_B) \vec{v}_X + (\vec{v}_A + \vec{v}_B) \times \vec{v}_X = 0 \]
Rotation Part (with quaternions, con’d)

Expressing this as a matrix equation

$$\begin{bmatrix}
(s_A - s_B) & (\mathbf{v}_A - \mathbf{v}_B)^T \\
(\mathbf{v}_A - \mathbf{v}_B) & (s_A - s_B) I_3 + sk((\mathbf{v}_A + \mathbf{v}_B))
\end{bmatrix}
\begin{bmatrix}
\mathbf{s}_X \\
\mathbf{v}_x
\end{bmatrix}
= \begin{bmatrix}
0 \\
0_3
\end{bmatrix}$$

If we now express the quaternion $\mathbf{q}_X$ as a 4-vector $\bar{\mathbf{q}}_X = [s_X, \mathbf{n}_X]^T$, we can express the AX=AB rotation problem as the system

$$M(q_A, q_B) \bar{\mathbf{q}}_X = \mathbf{0}_4$$

$$\|\bar{\mathbf{q}}_X\| = 1$$

Rotation Part (with quaternions, con’d)

In general, we have many observations, and we want to solve the problem in a least squares sense:

$$\text{min } \|M\bar{\mathbf{q}}_X\| \text{ subject to } \|\bar{\mathbf{q}}_X\| = 1$$

where

$$M = \begin{bmatrix}
M(q_{A,1}, q_{B,1}) \\
\vdots \\
M(q_{A,n}, q_{B,n})
\end{bmatrix} \text{ and } n \text{ is the number of observations}$$

Taking the singular value decomposition of $M=USV^T$ reduces this to the easier problem

$$\text{min } \|USV^T\bar{\mathbf{q}}_X\| = \|U\mathbf{y}\| = \|\mathbf{y}\| \text{ subject to } \|\mathbf{y}\| = \|V^T\bar{\mathbf{q}}_X\| = \|\bar{\mathbf{q}}_X\| = 1$$
Rotation Part (with quaternions, con’d)

This problem is just

$$\min \|\Sigma \tilde{y}\| = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & 0 & 0 & \sigma_4 \end{bmatrix} \begin{bmatrix} \tilde{y} \\ 1 \end{bmatrix}$$
subject to \(\|\tilde{y}\| = 1\)

where \(\sigma_i\) are the singular values. Recall that SVD routines return the \(\sigma_i \geq 0\) and sorted in decreasing magnitude. So \(\sigma_4\) is the smallest singular value and the value of \(\tilde{y}\) with \(\|\tilde{y}\| = 1\) that minimizes \(\|\Sigma \tilde{y}\|\) is \(\tilde{y} = \begin{bmatrix} 0, 0, 0, 1 \end{bmatrix}^T\). The corresponding value of \(\bar{q}_x\) is given by \(\bar{q}_x = V\tilde{y} = V_4\). Where \(V_4\) is the 4th column of \(V\).

Displacement part

The displacement part is given by

$$R_{Ak} \tilde{p}_x + \bar{p}_{Ak} \approx R_x \tilde{p}_{Bk} + \tilde{p}_x$$

Once we have solved for \(R_x\), we can rearrange the system above as

$$\begin{bmatrix} R_{Ak} - I \end{bmatrix} \tilde{p}_x \approx R_x \tilde{p}_{Bk} - \bar{p}_{Ak}$$

which we can solve by least squares.
Exercise: Perform a similar analysis to estimate $F_{wc}$.

Calibrating an Ultrasound Probe

Mapping the space

- Compare observed system performance to reference standard (“ground truth”)
- Interpolate residual errors

Example: Fluoroscope calibration
Projection of a point feature with distortion

\[ s = \lambda (a - d) + d \]

\[ \lambda = \frac{(a - d) \cdot (s - d)}{(a - d) \cdot (a - d)} \]

\[ u = f(d, \nu) \]

C-arm Calibration: Motivation

- Rectify (dewarp) geometric distortions in acquired images.
- Determine the geometry of cone-beam projection
- Compute calibration for each acquired x-ray

Prior studies: Boone et al., 1991; Fahrig et al., 1997; Yaniv et al., 1998; Yao, 2002; Daly et al., 2008; ...
C-arm Calibration: Instruments, Methods and Results

Initial x-ray of phantom
Vertical grid lines detected (horizontal follows)
Rectified phantom image
Diamond patterns detected → cone-beam parameters

Interpolation

• Ubiquitous throughout CIS research and applications
• Many techniques and methods
• Here are a few more notes

Click here for Interpolation.ppt
Experimental Setup

- C-Arm Detector and dewarp plate
- Robot Arm
- Corkscrew
- Phantom
- Surgical Cutter
Dewarping Method

Intrinsic Image Calibration

- Intrinsic imaging parameters (Schreiner et. al.)
- Image Warping (Checkerboard Based Method)
Step 0: Acquire Image

Step 1: Find groove points

- Find image points corresponding to the centerline of each vertical and horizontal groove
Step 2: Fit 5’th order Bernstein Polynomial Curves

- Fit least square smooth curve to each vertical and horizontal groove
- 5’th order Bernstein Polynomial

\[ B(a_0, \ldots, a_5, v) = \sum_{k=0}^{5} \binom{5}{k} (1-v)^{5-k} v^k \]

Step 3: Dewarp

- Employ a two pass scan line algorithm to dewarp the image
Advantages

• Fast
  – < 2 seconds on Pentium II 400
• Robust
  – works well even with overlaid objects
• Sub-pixel Accuracy
  – mean error 0.12 mm on the central area
• Does not completely obscure the image
  – trades off image contrast depth for image area
Spheres $i,j$:
- physical location in plate = $\mathbf{b}_i$
- Image location = $\mathbf{\tilde{u}}_i$

What are the physical coordinates in the plate associated with image coordinates $\mathbf{\tilde{u}}_i$?

Photos: Sofamor Danek

---

**Two Plane Method**

- Plane 1 pattern
- Plane 2 pattern

- E.g., Lavallee
- E.g., Helm
Two Plane Method

Given $q = \text{a point in image coordinates}$, determine the points

$$f_1^* = \text{the point on grid 1 corresponding to } q$$
$$f_2^* = \text{the point on grid 2 corresponding to } q$$

The desired ray in space passes through $f_1^*$ and $f_2^*$.

Two plane calibration

- Again, the essential problem is to determine the coordinates in the two planes at which the source-to-detector ray passes through the plane.
- Many methods for this. E.g.,
  - Find the four surrounding bead locations on each plane and use bilinear interpolation
  - Fit a general spline model for the distortion on each plane and then directly interpolate
Spheres $i,j$:
physical location in plate $= \mathbf{b}_j$
Image location $= \mathbf{u}_j$

What are the physical coordinates in the plate associated with image coordinates $\mathbf{u}_i$?

$[\lambda, \mu] \leftarrow \text{solve}(\mathbf{u}_i = \text{bilinear}(\lambda, \mu, \{\mathbf{u}_j\}))$
$\mathbf{b}_i \leftarrow \text{bilinear}(\lambda, \mu, \{\mathbf{b}_j\})$

Spheres $i,j$:
physical location in other plate $= \mathbf{c}_j$
Image location $= \tilde{\mathbf{u}}_j$

What are the physical coordinates $\mathbf{c}_i$ in the other plate associated with image coordinates $\mathbf{u}_i$?

$[\lambda, \mu] \leftarrow \text{solve}(\mathbf{u}_i = \text{bilinear}(\lambda, \mu, \{\tilde{\mathbf{u}}_j\}))$
$\mathbf{c}_i \leftarrow \text{bilinear}(\lambda, \mu, \{\mathbf{c}_j\})$
So the points in space on the line from the x-ray source to detector corresponding to the image coordinates \( \mathbf{u}_i \) will be given by

\[
\mathbf{b}_i + \gamma (\mathbf{c}_i - \mathbf{b}_i)
\]