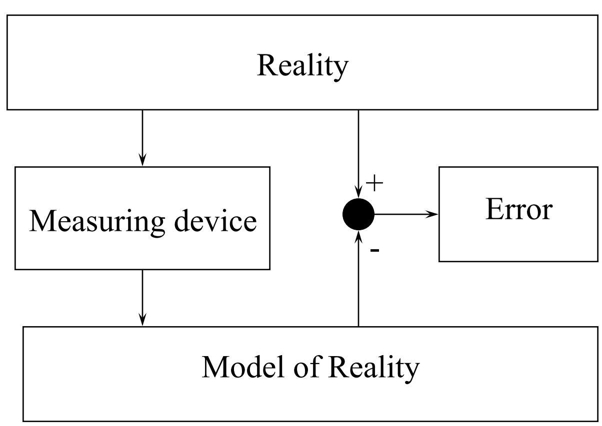
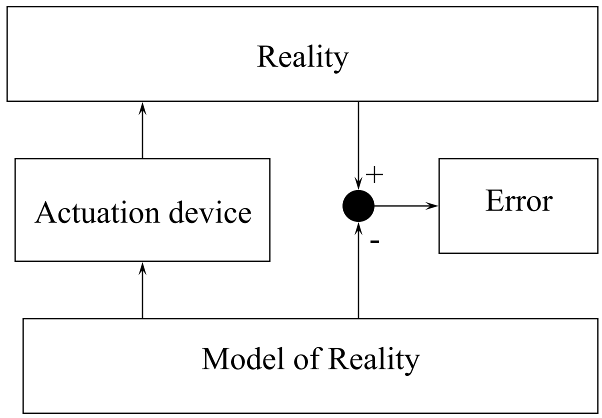
Calibration

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



Calibration

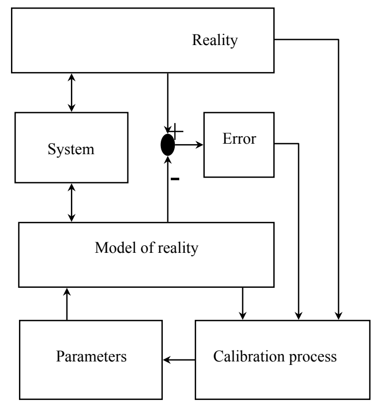


Basic Techniques

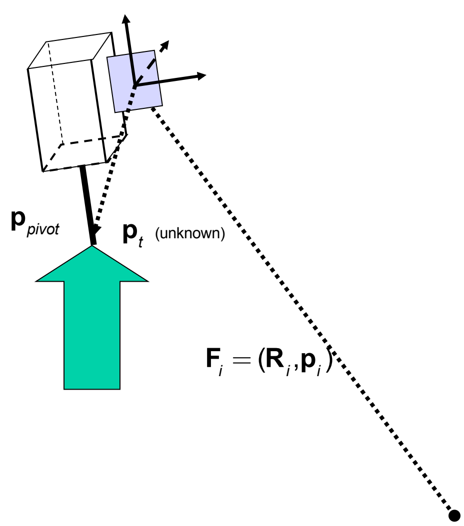
- **Parameter Estimation**
- **Mapping the space**

Parameter Estimation

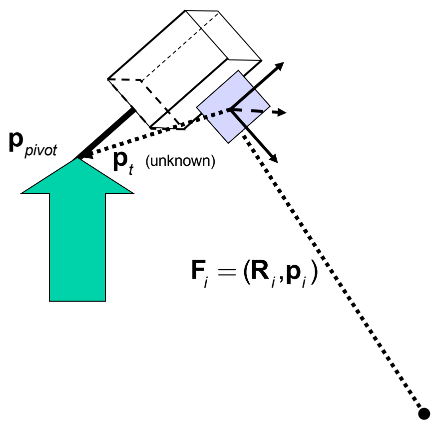
- Compare observed system performance to reference standard ("ground truth")
- Compute parameters of mathematical model that minimizes residual error.



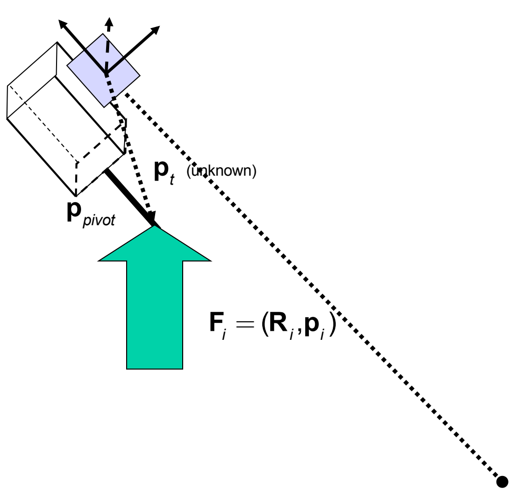
Pointing device calibration



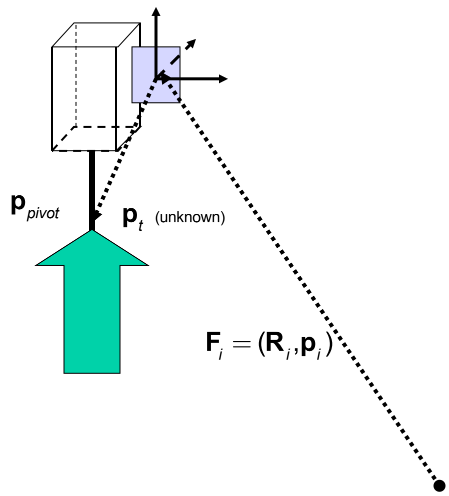
Pointing device calibration



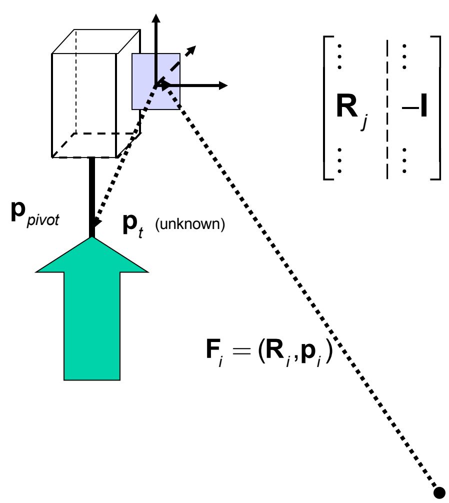
Pointing device calibration



Pointing device calibration



Pointing device calibration



$$\begin{bmatrix} \vdots & \vdots \\ \mathbf{R}_j & -\mathbf{I} \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{p}_t \\ \mathbf{p}_{pivot} \end{bmatrix} \cong \begin{bmatrix} \vdots \\ -\mathbf{p}_j \\ \vdots \end{bmatrix}$$

Parameter estimation

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

$$\vec{\mathbf{q}}^* = \arg\min_{\vec{\mathbf{q}}} E\left(\left\{\vec{\mathbf{f}}(\vec{\mathbf{x}}_k;\vec{\mathbf{q}}),\vec{\mathbf{p}}_k\right\}\right)$$

for a function $\vec{\mathbf{f}}(\vec{\mathbf{x}};\vec{\mathbf{q}})$ and observations $\vec{\mathbf{p}}_k = \vec{\mathbf{f}}(\vec{\mathbf{x}}_k;?)$

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 $E\left(\left\{\vec{\mathbf{f}}(\vec{\mathbf{x}}_k;\vec{\mathbf{q}}),\vec{\mathbf{p}}_k\right\}\right)$ is a sum of squares

$$E\left(\left\{\vec{\mathbf{f}}(\vec{\mathbf{x}}_k;\vec{\mathbf{q}}),\vec{\mathbf{p}}_k\right\}\right) = \sum_k \left\|\vec{\mathbf{f}}(\vec{\mathbf{x}}_k;\vec{\mathbf{q}}) - \vec{\mathbf{p}}_k\right\|^2$$

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

$$E\left(\left\{\vec{\mathbf{f}}(\vec{\mathbf{x}}_k;\vec{\mathbf{q}}),\vec{\mathbf{p}}_k\right\}\right) = \sum_k \left\|\vec{\mathbf{f}}(\vec{\mathbf{x}}_k;\vec{\mathbf{q}}) - \vec{\mathbf{p}}_k\right\|_{L1}$$

Linear Parameter Estimation

$$\mathbf{p}_{nom} = \mathbf{f}(\mathbf{q}) \quad \text{where } \mathbf{q} = [q_1,...q_n]^{\mathrm{T}} \text{ are parameters}$$

$$\begin{aligned}
\mathbf{p}^* &= \mathbf{f}(\mathbf{q} + \Delta\mathbf{q}) \\
&\approx \mathbf{f}(\mathbf{q}) + \begin{bmatrix} \ddots & \vdots & \\ \cdots & \dfrac{\partial f_i}{\partial q_j}(\mathbf{q}) & \cdots \\ & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \Delta q_1 \\ \vdots \\ \Delta q_n \end{bmatrix} \\
&\equiv \mathbf{f}(\mathbf{q}) + J_{\mathbf{f}}(\mathbf{q})\Delta\mathbf{q}
\end{aligned}$$

Parameter estimation: least squares adjustment

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

$$\vec{\mathbf{q}}^* = \arg\min \sum_k \left(\vec{\mathbf{f}}(\vec{\mathbf{x}}_k;\vec{\mathbf{q}}) - \vec{\mathbf{p}}_k\right)^2$$

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

Step 1 Solve the least squares problem

$$\begin{bmatrix} \vdots \\ J_{\mathbf{f}}(\vec{\mathbf{q}}^{(i)}) \\ \vdots \end{bmatrix} \Delta\vec{\mathbf{q}}^{(i+1)} \cong \begin{bmatrix} \vdots \\ \vec{\mathbf{p}}_k^* - \vec{\mathbf{f}}(\vec{\mathbf{q}}^{(i)}) \\ \vdots \end{bmatrix} \text{ to find } \Delta\vec{\mathbf{q}}^{(i+1)}$$

Step 2 $\vec{\mathbf{q}}^{(i+1)} \leftarrow \vec{\mathbf{q}}^{(i)} + \Delta\vec{\mathbf{q}}^{(i+1)}$; evaluate $\left\{\vec{\mathbf{e}}_k \leftarrow \vec{\mathbf{p}}_k^* - \vec{\mathbf{f}}(\vec{\mathbf{q}}^{(i+1)})\right\}$; $\zeta^{(i+1)} \leftarrow \sum_k \vec{\mathbf{e}}_k \bullet \vec{\mathbf{e}}_k$

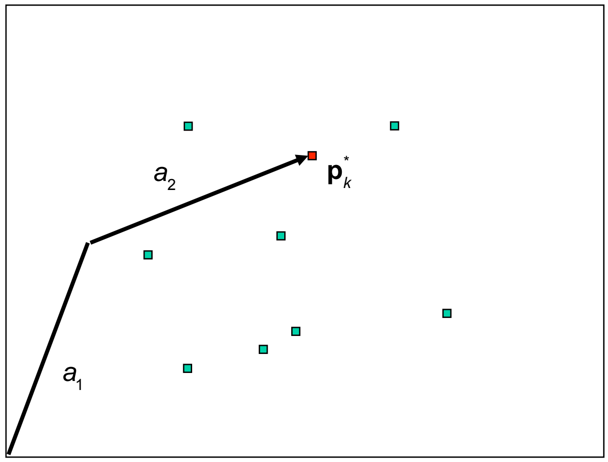
Step 3 If $\zeta^{(i+1)}$ is small enough, or otherwise converged, then stop.
Else set $i \leftarrow i+1$ and go back to Step 1.

Linear Least Squares

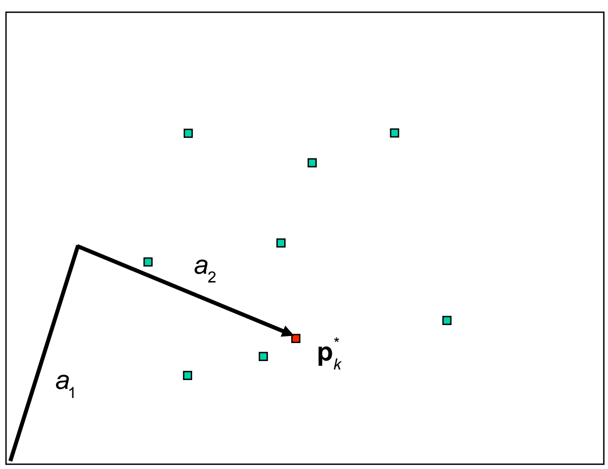
- Most commonly used method for parameter estimation
- Many numerical libraries
- See the web site
- Here is a quick review

Click Here

Example: 2 link robot calibration



Example: 2 link robot calibration



Example: 2 link robot calibration

$$\mathbf{p} = \begin{bmatrix} a_1 \sin\theta_1 + a_2 \sin(\theta_{12}) \\ 0 \\ a_1 \cos\theta_1 + a_2 \cos(\theta_{12}) \end{bmatrix} \qquad \text{where } \theta_{12} = \theta_1 + \theta_2$$

$$\mathbf{p}^* = \begin{bmatrix} (a_1 + \Delta a_1)\sin(\theta_1 + \Delta\theta_1) + (a_2 + \Delta a_2)\sin(\theta_{12} + \Delta\theta_1 + \Delta\theta_2) \\ 0 \\ (a_1 + \Delta a_1 \cos(\theta_1 + \Delta\theta_1) + (a_2 + \Delta a_2)\cos(\theta_{12} + \Delta\theta_1 + \Delta\theta_2) \end{bmatrix}$$

Example: 2 link robot calibration

$$\mathbf{p}_k = \mathbf{f}(\mathbf{q}_k) = \mathbf{f}(a_1, a_2, \theta_1, \theta_2) = \begin{bmatrix} a_1 \sin\theta_{1,k} + a_2 \sin(\theta_{12,k}) \\ 0 \\ a_1 \cos\theta_{1,k} + a_2 \cos(\theta_{12,k}) \end{bmatrix} \qquad \text{where } \theta_{12} = \theta_1 + \theta_2$$

so we solve the least squares problem

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \mathbf{f}}{\partial a_1}(\mathbf{q}_k) & \frac{\partial \mathbf{f}}{\partial a_2}(\mathbf{q}_k) & \frac{\partial \mathbf{f}}{\partial \theta_1}(\mathbf{q}_k) & \frac{\partial \mathbf{f}}{\partial \theta_2}(\mathbf{q}_k) \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \Delta a_1 \\ \Delta a_2 \\ \Delta\theta_1 \\ \Delta\theta_2 \end{bmatrix} \approx \begin{bmatrix} \vdots \\ \mathbf{p}_k^* - \begin{bmatrix} a_1 Rot(\mathbf{y}, \theta_{1,k}) \\ + a_2 Rot(\mathbf{y}, \theta_{1,k}) \end{bmatrix} \\ \vdots \end{bmatrix}$$

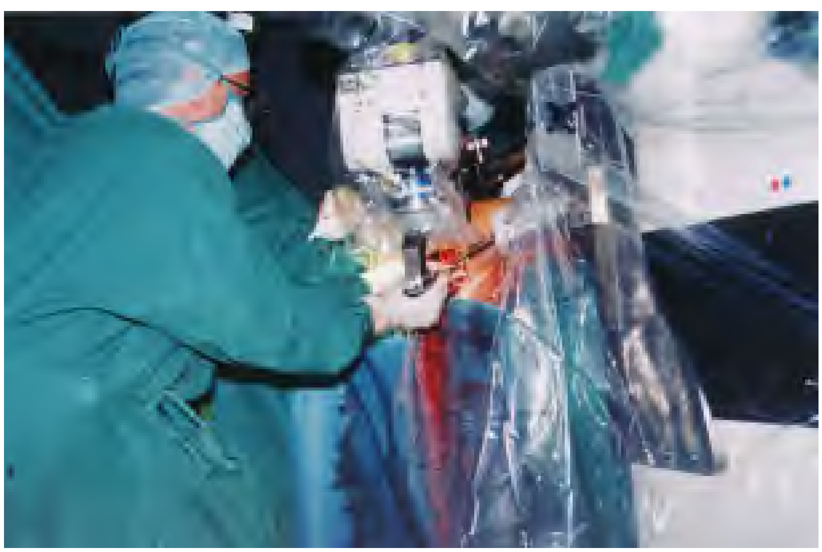
Example: 2 link robot calibration

Here

$$J_{\mathbf{f}}(\mathbf{q}_k) = \begin{bmatrix} \sin\theta_{1,k} & \sin\theta_{12,k} & a_1(\cos\theta_{1,k} + \cos\theta_{12,k}) & a_2\cos\theta_{12,k} \\ 0 & 0 & 0 & 0 \\ \cos\theta_{1,k} & \cos\theta_{12,k} & -a_1(\sin\theta_{1,k} + \sin\theta_{12,k}) & -a_2\cos\theta_{12,k} \end{bmatrix}$$

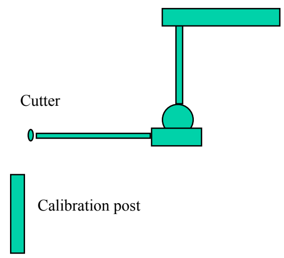
so

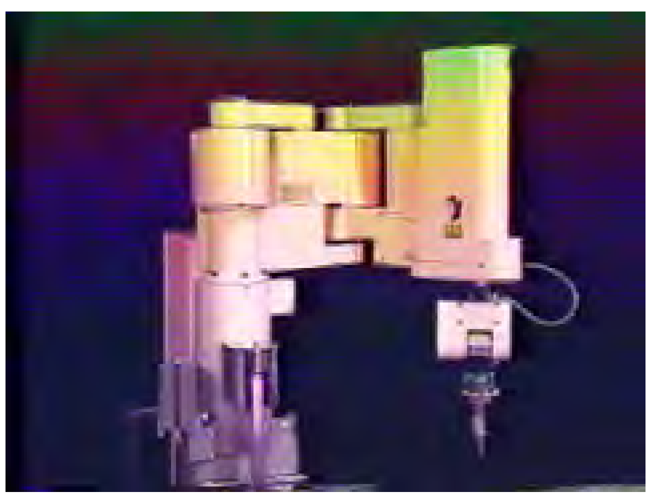
$$\begin{bmatrix} & & \vdots & \\ \sin\theta_{1,k} & \sin\theta_{12,k} & a_1(\cos\theta_{1,k} + \cos\theta_{12,k}) & a_2\cos\theta_{12,k} \\ \cos\theta_{1,k} & \cos\theta_{12,k} & -a_1(\sin\theta_{1,k} + \sin\theta_{12,k}) & -a_2\cos\theta_{12,k} \\ & & \vdots & \end{bmatrix} \begin{bmatrix} \Delta a_1 \\ \Delta a_2 \\ \Delta\theta_1 \\ \Delta\theta_2 \end{bmatrix} \approx \begin{bmatrix} \vdots \\ x_k^* - a_1\sin\theta_{1,k} + a_2\sin(\theta_{12,k}) \\ z_k^* - a_1\cos\theta_{1,k} + a_2\cos(\theta_{12,k}) \\ \vdots \end{bmatrix}$$



Example: Robodoc Wrist Calibration

- Basic robot had very accurate calibration
- Custom wrist was less accurate
- Crucial goal was to determine position of cutter tip





Kinematic Model

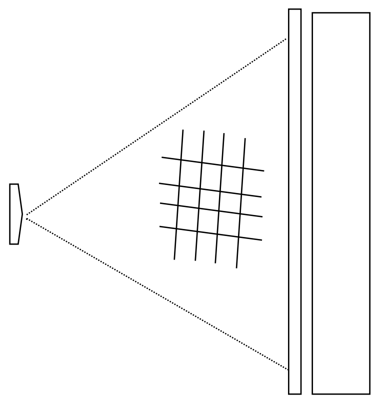
$$\mathbf{p}_{tool} = \mathbf{p}_{wrist} + \mathbf{R}(\mathbf{z}, \theta_4 + \Delta\theta_4) \bullet (\alpha\, \mathbf{x} + \mathbf{v}_{distal})$$

$$\mathbf{v}_{distal} = \mathbf{R}(\mathbf{x}, \beta) \bullet [\mathbf{R}(\mathbf{y}, \theta_5 + \Delta\theta_5)(\mathbf{v}_c + \Delta\mathbf{v}_c)]$$

Linearization

$$\mathbf{p}_{post} \approx \mathbf{p}_{wrist} + [\mathbf{R}_4 \mathbf{R}_5 (\mathbf{v}_c + \Delta\mathbf{v}_c)] + \ldots..$$

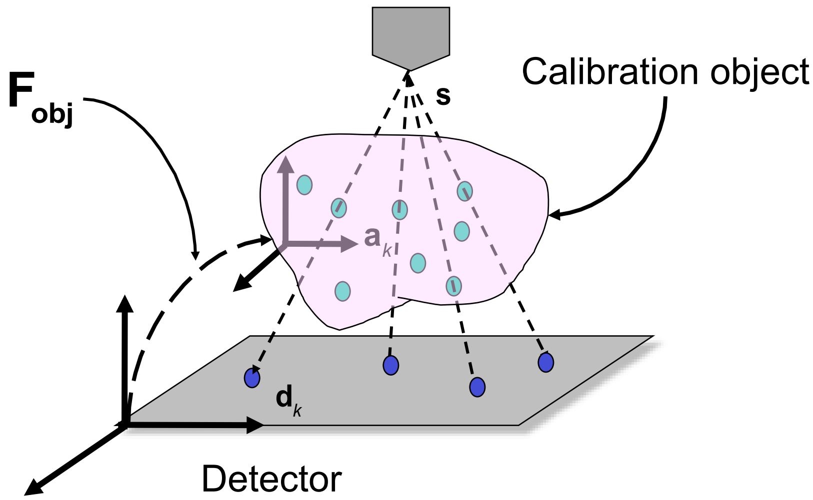
Example: Undistorted fluoroscope calibration



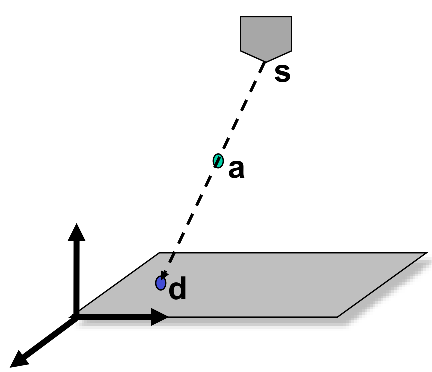
Calibration if no distortion (version 1)

Assume no distortion. For the moment also assume that you have N point calibration features (e.g., small steel balls) at known positions $\{\mathbf{a}_0, \cdots, \mathbf{a}_{N-1}\}$ relative to the detector. Assume further that the points create images at corresponding points $\{\mathbf{d}_0, \cdots, \mathbf{d}_{N-1}\}$ on the detector. Estimate the position $\mathbf{s}$ of the x-ray source relative to the detector

Approach



Projection of a point feature



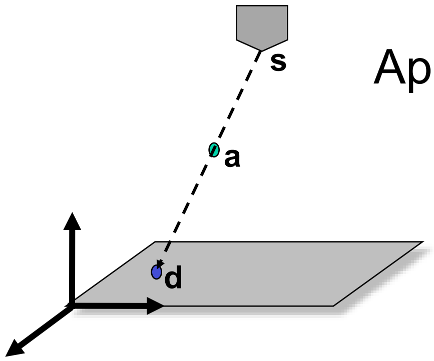
$$\mathbf{s} = \lambda(\mathbf{a} - \mathbf{d}) + \mathbf{d}$$

$$\lambda = \frac{(\mathbf{a} - \mathbf{d}) \bullet (\mathbf{s} - \mathbf{d})}{(\mathbf{a} - \mathbf{d}) \bullet (\mathbf{a} - \mathbf{d})}$$

$$\mathbf{d} = \mu(\mathbf{a} - \mathbf{s}) + \mathbf{s}$$

$$\mu = \frac{(\mathbf{a} - s) \bullet (\mathbf{d} - \mathbf{s})}{(\mathbf{a} - s) \bullet (\mathbf{a} - \mathbf{s})}$$

Approach



$$(\mathbf{a}-\mathbf{d})\times(\mathbf{s}-\mathbf{d})=\mathbf{0}$$

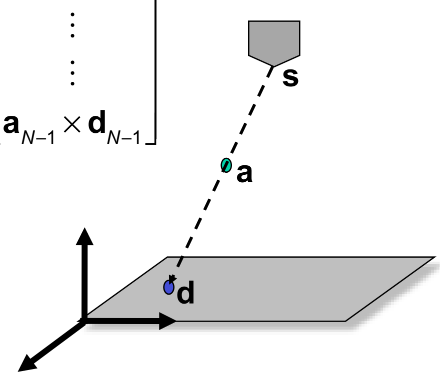
$$\begin{aligned} skew(\mathbf{a}-\mathbf{d})\bullet s &= (\mathbf{a}-\mathbf{d})\times\mathbf{d} \\ &= \mathbf{a}\times\mathbf{d}-\mathbf{d}\times\mathbf{d} \\ &= \mathbf{a}\times\mathbf{d} \end{aligned}$$

$$\begin{bmatrix} 0 & \mathbf{d}_z-\mathbf{a}_z & \mathbf{a}_y-\mathbf{d}_y \\ \mathbf{a}_z-\mathbf{d}_z & 0 & \mathbf{d}_x-\mathbf{a}_x \\ \mathbf{d}_y-\mathbf{a}_y & \mathbf{a}_x-\mathbf{d}_x & 0 \end{bmatrix}\mathbf{s}=\mathbf{a}\times\mathbf{d}$$

Approach

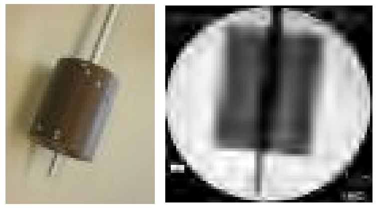
Solve least squares problem

$$\begin{bmatrix} skew(\mathbf{a}_0-\mathbf{d}_0) \\ \vdots \\ \vdots \\ skew(\mathbf{a}_{N-1}-\mathbf{d}_{N-1}) \end{bmatrix}\begin{bmatrix} \mathbf{s}_x \\ \mathbf{s}_y \\ \mathbf{s}_z \end{bmatrix} \cong \begin{bmatrix} \mathbf{a}_0\times\mathbf{d}_0 \\ \vdots \\ \vdots \\ \mathbf{a}_{N-1}\times\mathbf{d}_{N-1} \end{bmatrix}$$

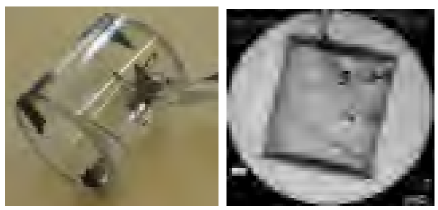


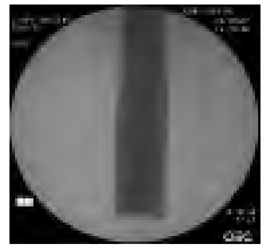
Typical fiducial objects

Points



Curves

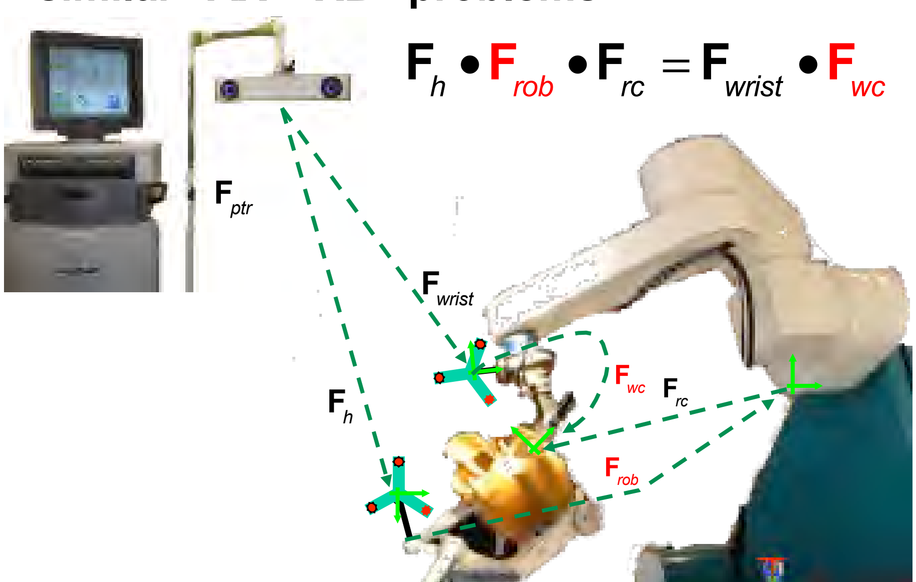




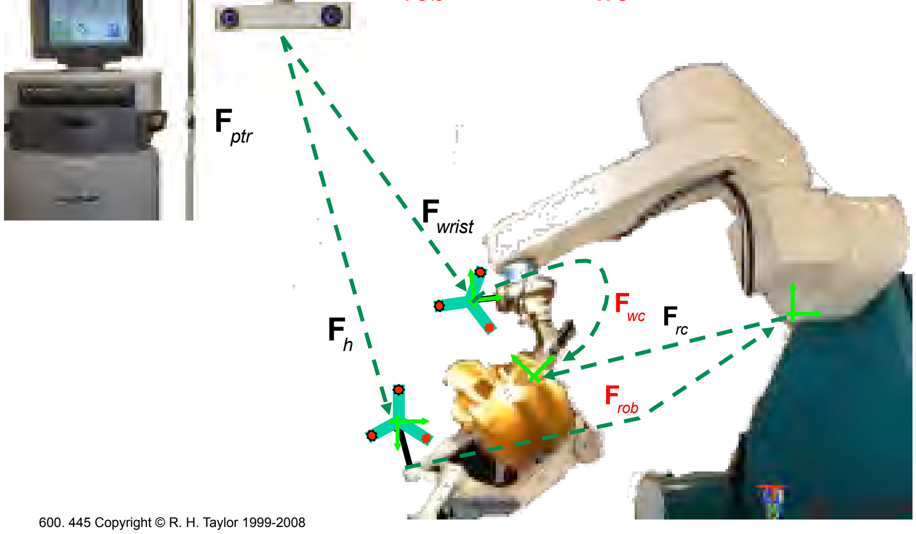
FTRAC
(Jain *et al.*)

Calibrating trackers to robots and similar “AX = XB” problems

$$\mathbf{F}_h \bullet \mathbf{F}_{rob} \bullet \mathbf{F}_{rc} = \mathbf{F}_{wrist} \bullet \mathbf{F}_{wc}$$



Problem: How do you determine $\mathbf{F}_{rob}$ and $\mathbf{F}_{wc}$?

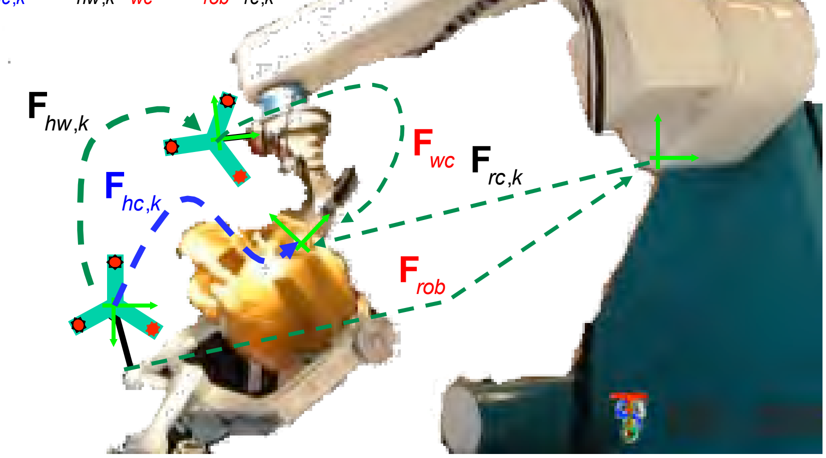


To simplify this situation, define

$$\mathbf{F}_{hw,k} = \mathbf{F}_{h,k}^{-1}\mathbf{F}_{wrist,k}$$

This gives

$$\mathbf{F}_{hc,k} = \mathbf{F}_{hw,k}\mathbf{F}_{wc} = \mathbf{F}_{rob}\mathbf{F}_{rc,k}$$



Move the robot to a sequence of poses $\mathbf{F}_{rc,k}$ for $k = 0,\ldots,N$ and observe the corresponding values of $\mathbf{F}_{hw,k}$. Define

$$\mathbf{A}_k = \mathbf{F}_{hw,k}\mathbf{F}_{hw,0}^{-1} \text{ for } k = 1,\ldots,N$$

$$\mathbf{B}_k = \mathbf{F}_{rc,k}\mathbf{F}_{rc,0}^{-1} \text{ for } k = 1,\ldots,N$$

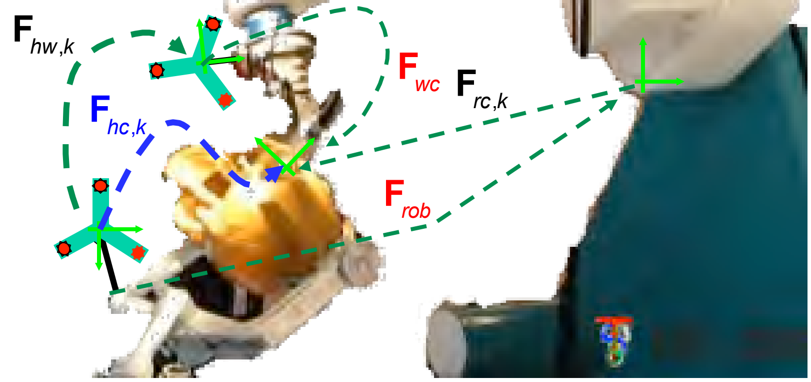
$\mathbf{A}_k$ represents motion from an initial pose

$$\mathbf{A}_k\mathbf{F}_{hw,0} = \mathbf{F}_{hw,k}$$

so

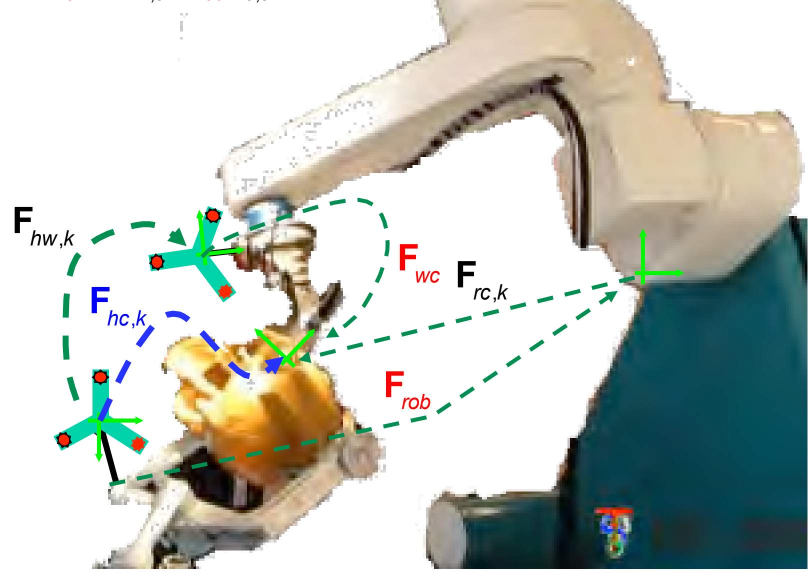
$$\mathbf{A}_k = \mathbf{F}_{hw,k}\mathbf{F}_{hw,0}^{-1}$$

and similarly for $\mathbf{B}_k$



$$\mathbf{F}_{hw,0}\mathbf{F}_{wc} = \mathbf{F}_{rob}\mathbf{F}_{rc,0}$$

$$\mathbf{F}_{wc} = \mathbf{F}_{hw,0}^{-1}\mathbf{F}_{rob}\mathbf{F}_{rc,0}$$

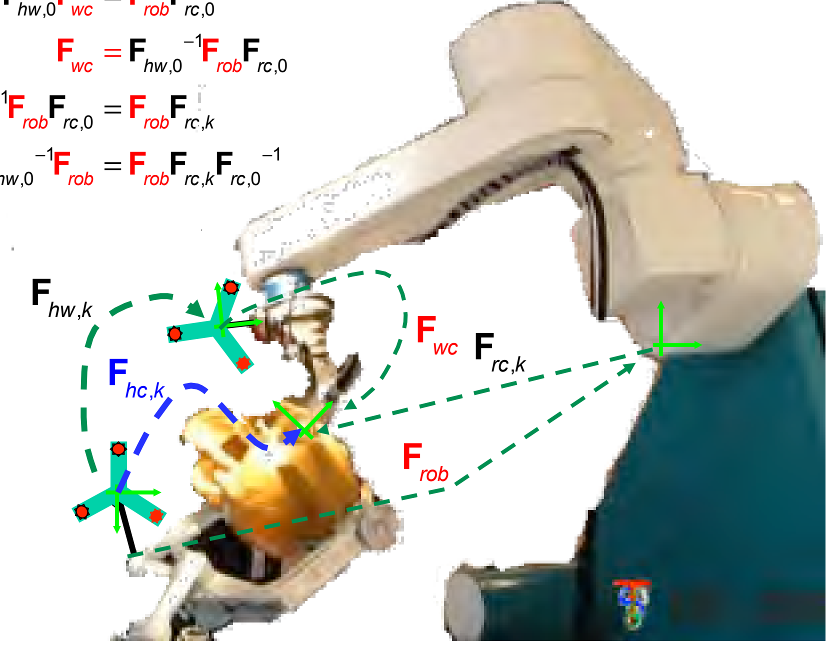


$$\mathbf{F}_{hw,0}\mathbf{F}_{wc} = \mathbf{F}_{rob}\mathbf{F}_{rc,0}$$

$$\mathbf{F}_{wc} = \mathbf{F}_{hw,0}^{-1}\mathbf{F}_{rob}\mathbf{F}_{rc,0}$$

$$\mathbf{F}_{hw,k}\mathbf{F}_{hw,0}^{-1}\mathbf{F}_{rob}\mathbf{F}_{rc,0} = \mathbf{F}_{rob}\mathbf{F}_{rc,k}$$

$$\mathbf{F}_{hw,k}\mathbf{F}_{hw,0}^{-1}\mathbf{F}_{rob} = \mathbf{F}_{rob}\mathbf{F}_{rc,k}\mathbf{F}_{rc,0}^{-1}$$

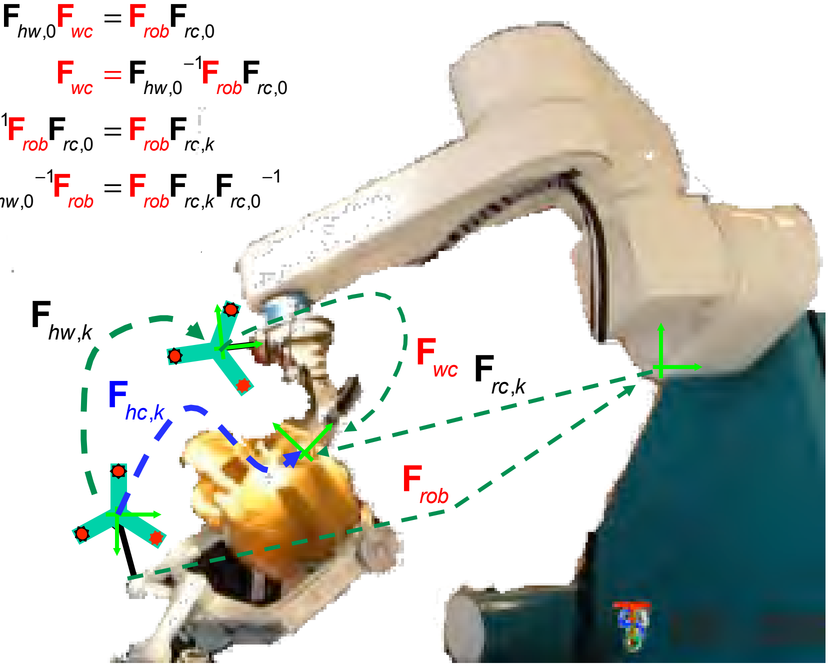


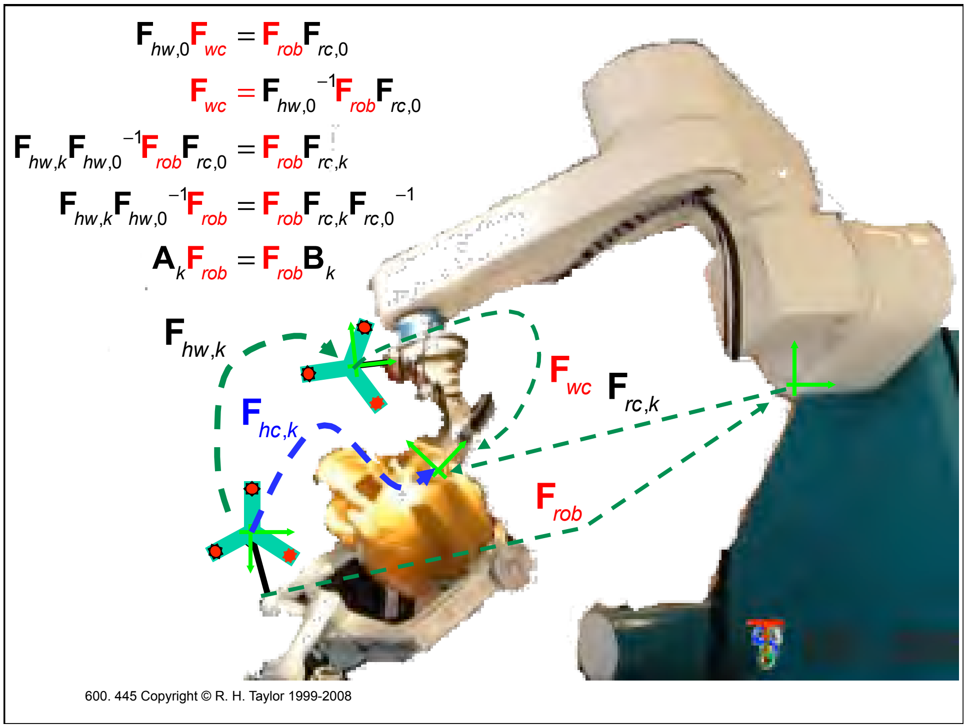
$$\mathbf{F}_{hw,0}\mathbf{F}_{wc} = \mathbf{F}_{rob}\mathbf{F}_{rc,0}$$

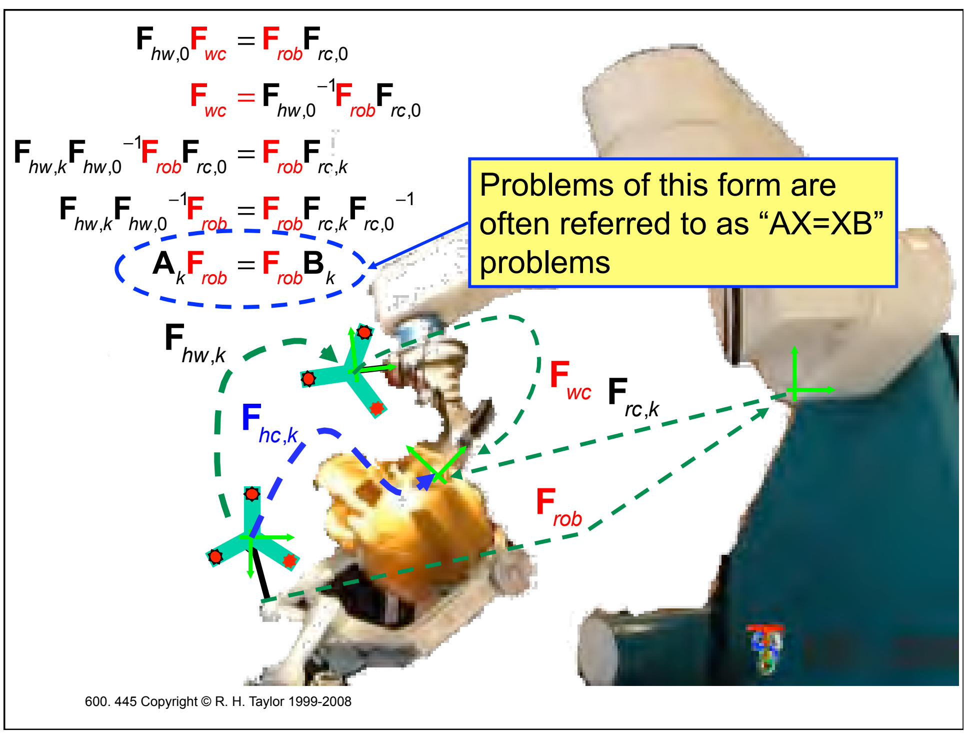
$$\mathbf{F}_{wc} = \mathbf{F}_{hw,0}^{-1}\mathbf{F}_{rob}\mathbf{F}_{rc,0}$$

$$\mathbf{F}_{hw,k}\mathbf{F}_{hw,0}^{-1}\mathbf{F}_{rob}\mathbf{F}_{rc,0} = \mathbf{F}_{rob}\mathbf{F}_{rc,k}$$

$$\mathbf{F}_{hw,k}\mathbf{F}_{hw,0}^{-1}\mathbf{F}_{rob} = \mathbf{F}_{rob}\mathbf{F}_{rc,k}\mathbf{F}_{rc,0}^{-1}$$




$F_{hw,0} F_{wc} = F_{rob} F_{rc,0}$
$F_{wc} = F_{hw,0}^{-1} F_{rob} F_{rc,0}$
$F_{hw,k} F_{hw,0}^{-1} F_{rob} F_{rc,0} = F_{rob} F_{rc,k}$
$F_{hw,k} F_{hw,0}^{-1} F_{rob} = F_{rob} F_{rc,k} F_{rc,0}^{-1}$
$A_k F_{rob} = F_{rob} B_k$
$F_{hw,k}$
$F_{hc,k}$
F_{wc}
$F_{rc,k}$
F_{rob}
600. 445 Copyright © R. H. Taylor 1999-2008


$F_{hw,0} F_{wc} = F_{rob} F_{rc,0}$
$F_{wc} = F_{hw,0}^{-1} F_{rob} F_{rc,0}$
$F_{hw,k} F_{hw,0}^{-1} F_{rob} F_{rc,0} = F_{rob} F_{rc,k}$
$F_{hw,k} F_{hw,0}^{-1} F_{rob} = F_{rob} F_{rc,k} F_{rc,0}^{-1}$
$A_k F_{rob} = F_{rob} B_k$
Problems of this form are often referred to as “AX=XB” problems
$F_{hw,k}$
$F_{hc,k}$
F_{wc}
$F_{rc,k}$
F_{rob}
600. 445 Copyright © R. H. Taylor 1999-2008

Solving "AX = XB" problems where X is a rigid transformation

Given known frame transformations $\{\mathbf{F}_{A,k},\mathbf{F}_{B,k}\}$ we want to find a best estimate $\mathbf{F}_x = [\mathbf{R}_X,\vec{\mathbf{p}}_X]$ such that $\mathbf{F}_{A,k} \bullet \mathbf{F}_X \approx \mathbf{F}_X \bullet \mathbf{F}_{B,k}$.

This is equivalent to

$$\mathbf{R}_{A,k}\mathbf{R}_X \approx \mathbf{R}_X\mathbf{R}_{B,k}$$

$$\mathbf{R}_{A,k}\vec{\mathbf{p}}_X + \vec{\mathbf{p}}_{A,k} \approx \mathbf{R}_X\vec{\mathbf{p}}_{B,k} + \vec{\mathbf{p}}_X$$

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

$$\mathbf{R}_{A,k}\mathbf{R}_X \approx \mathbf{R}_X\mathbf{R}_{B,k}$$

Using the notation

$$\mathbf{R}_A = Rot(\vec{\alpha}) = Rot(\frac{\vec{\alpha}}{\|\vec{\alpha}\|},\|\vec{\alpha}\|) = Rot(\vec{\mathbf{n}}_{\mathbf{A}},\theta_A)$$

etc., we recall that

$$\mathbf{R}_A\mathbf{R}_X = Rot(\vec{\mathbf{n}}_A,\theta_A)\mathbf{R}_X = \mathbf{R}_X Rot(\mathbf{R}_X^{-1}\vec{\mathbf{n}}_A,\theta_A)$$

So

$$\mathbf{R}_X Rot(\mathbf{R}_X^{-1}\vec{\mathbf{n}}_A,\theta_A) = \mathbf{R}_X Rot(\vec{\mathbf{n}}_B,\theta_B)$$

Rotation Part (less good way), continued

From previous slide

$$\mathbf{R}_X Rot(\mathbf{R}_X^{-1}\vec{\mathbf{n}}_A, \theta_A) = \mathbf{R}_X Rot(\vec{\mathbf{n}}_B, \theta_B)$$

Multiplying both sides by by $\mathbf{R}_X^{-1}$ gives

$$Rot(\mathbf{R}_X^{-1}\vec{\mathbf{n}}_A, \theta_A) = Rot(\vec{\mathbf{n}}_B, \theta_B)$$

This can be expressed as

$$\mathbf{R}_X^{-1}\vec{\alpha} = \vec{\beta}$$

where $\vec{\alpha} = \theta_A \vec{\mathbf{n}}_A$ and $\vec{\beta} = \theta_B \vec{\mathbf{n}}_B$. Rearranging and inserting subscripts gives a system

$$\mathbf{R}_X \vec{\beta}_k = \vec{\alpha}_k$$

which can be solved for $\mathbf{R}_X$ by standard rigid rotation estimation methods .

Rotation Part (with quaternions)

Let $\mathbf{q}_X = s_X + \vec{\mathbf{v}}_X$ be the unit quaternion corresponding to $\mathbf{R}_X$, with similar definitions for $\mathbf{q}_A$ and $\mathbf{q}_B$. Then we have for $\mathbf{R}_A \mathbf{R}_x = \mathbf{R}_x \mathbf{R}_B$

$$\mathbf{q}_A \mathbf{q}_X = \mathbf{q}_X \mathbf{q}_B$$

Expanding the scalar and vector parts gives

$$s_A s_X - \vec{\mathbf{v}}_A \bullet \vec{\mathbf{v}}_X = s_X s_B - \vec{\mathbf{v}}_X \bullet \vec{\mathbf{v}}_B$$

$$s_A \vec{\mathbf{v}}_X + s_X \vec{\mathbf{v}}_A + \vec{\mathbf{v}}_A \times \vec{\mathbf{v}}_X = s_X \vec{\mathbf{v}}_B + s_B \vec{\mathbf{v}}_X + \vec{\mathbf{v}}_X \times \vec{\mathbf{v}}_B$$

Rearranging ...

$$\left(s_A - s_B\right) s_X - \left(\vec{\mathbf{v}}_A - \vec{\mathbf{v}}_B\right) \bullet \vec{\mathbf{v}}_x = 0$$

$$\left(\vec{\mathbf{v}}_A - \vec{\mathbf{v}}_B\right) s_X + (s_A - s_B)\vec{\mathbf{v}}_X + \left(\vec{\mathbf{v}}_A + \vec{\mathbf{v}}_B\right) \times \vec{\mathbf{v}}_X = \vec{\mathbf{0}}_3$$

Rotation Part (with quaternions, con'd)

Expressing this as a matrix equation

$$\left[\begin{array}{c|c} (s_A - s_B) & (\vec{\mathbf{v}}_A - \vec{\mathbf{v}}_B)^T \\ \hline (\vec{\mathbf{v}}_A - \vec{\mathbf{v}}_B) & (s_A - s_B)\mathbf{I}_3 + sk((\vec{\mathbf{v}}_A + \vec{\mathbf{v}}_B) \end{array}\right]\left[\begin{array}{c} s_X \\ \hline \vec{\mathbf{v}}_x \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \vec{\mathbf{0}}_3 \end{array}\right]$$

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

$$\mathbf{M}(\mathbf{q}_A, \mathbf{q}_B)\vec{\mathbf{q}}_X = \vec{\mathbf{0}}_4$$
$$\left\|\vec{\mathbf{q}}_X\right\| = 1$$

Rotation Part (with quaternions, con'd)

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

$$\min \left\|\mathbf{M}\vec{\mathbf{q}}_X\right\| \text{ subject to } \left\|\vec{\mathbf{q}}_X\right\| = 1$$

where

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

Taking the singular value decomposition of $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T$ reduces this to the easier problem

$$\min \left\|\mathbf{U}\Sigma\mathbf{V}^T\vec{\mathbf{q}}_X\right\| = \left\|\mathbf{U}(\Sigma\vec{\mathbf{y}})\right\| = \left\|\Sigma\vec{\mathbf{y}}\right\| \text{ subject to } \left\|\vec{\mathbf{y}}\right\| = \left\|\mathbf{V}^T\vec{\mathbf{q}}_X\right\| = \left\|\vec{\mathbf{q}}_X\right\| = 1$$

Rotation Part (with quaternions, con'd)

This problem is just

$$\min \left\|\Sigma\vec{\mathbf{y}}\right\| = \left\| \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & 0 & 0 & \sigma_4 \end{bmatrix} \vec{\mathbf{y}} \right\| \quad \text{subject to } \left\|\vec{\mathbf{y}}\right\| = 1$$

where σ_i are the singular values. Recall that SVD routines return the $\sigma_i \geq 0$ and sorted in decreasing magnitude. So σ_4 is the smallest singular value and the value of $\vec{\mathbf{y}}$ with $\left\|\vec{\mathbf{y}}\right\| = 1$ that minimizes $\left\|\Sigma\vec{\mathbf{y}}\right\|$ is $\vec{\mathbf{y}} = \left[0,0,0,1\right]^T$. The corresponding value of $\vec{\mathbf{q}}_X$ is given by $\vec{\mathbf{q}}_x = \mathbf{V}\vec{\mathbf{y}} = \mathbf{V}_4$. Where $\mathbf{V}_4$ is the 4th column of **V**.

Displacement part

The displacement part is given by

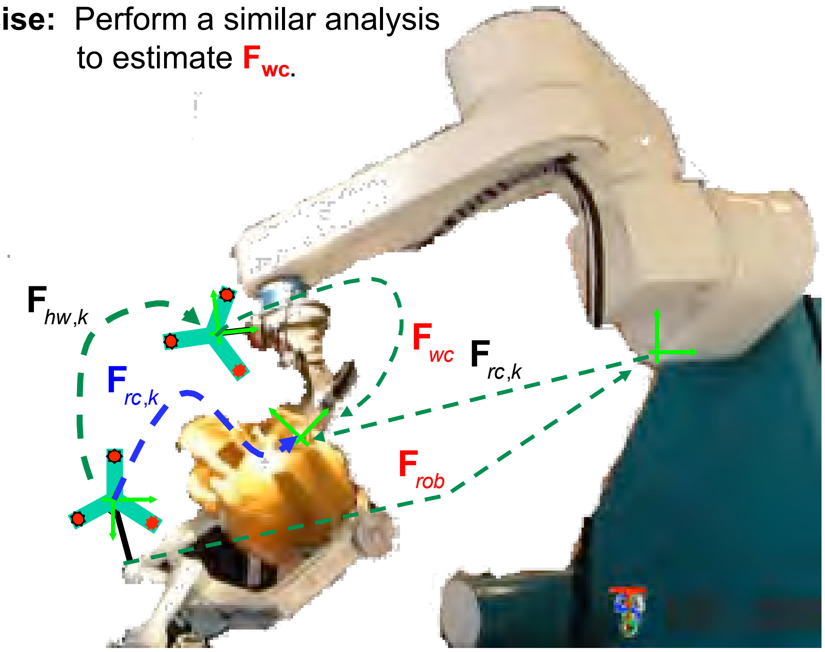
$$\mathbf{R}_{A,k}\vec{\mathbf{p}}_X + \vec{\mathbf{p}}_{A,k} \approx \mathbf{R}_X\vec{\mathbf{p}}_{B,k} + \vec{\mathbf{p}}_X$$

Once we have solved for $\mathbf{R}_X$, we can rearrange the system above as

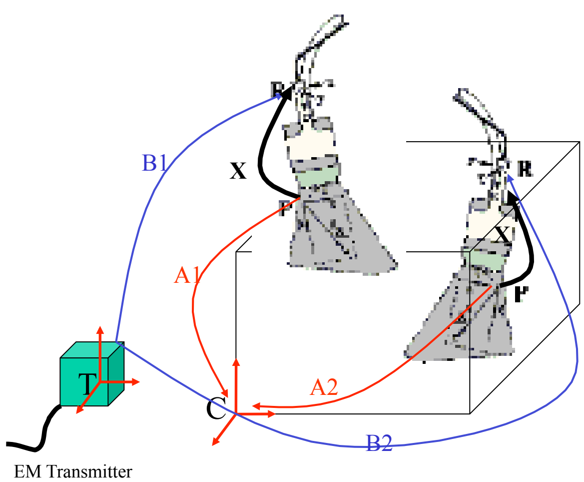
$$\left(\mathbf{R}_{A,k} - \mathbf{I}\right)\vec{\mathbf{p}}_X \approx \mathbf{R}_X\vec{\mathbf{p}}_{B,k} - \vec{\mathbf{p}}_{A,k}$$

which we can solve by least squares

Exercise: Perform a similar analysis to estimate $\mathbf{F}_{wc}$.



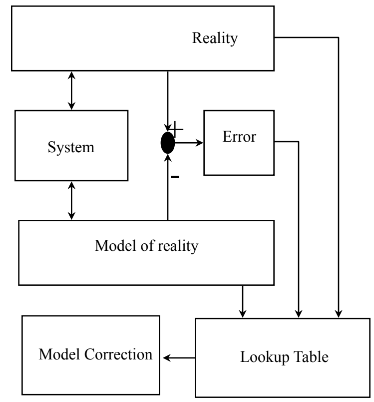
Calibrating an Ultrasound Probe



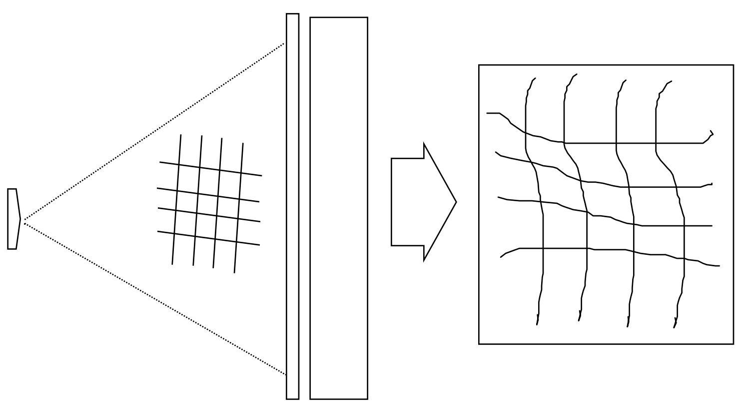
Boctor E, et. al., "A Novel Closed Form Solution For Ultrasound Calibration", ISBI 2004.

Mapping the space

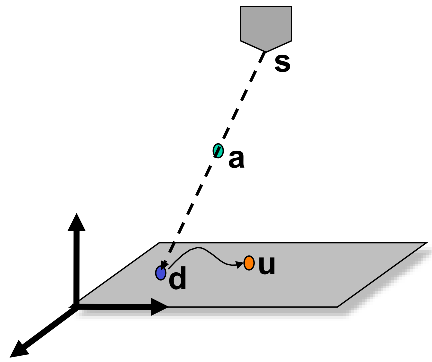
- Compare observed system performance to reference standard ("ground truth")
- Interpolate residual errors



Example: Fluoroscope calibration



Projection of a point feature with distortion



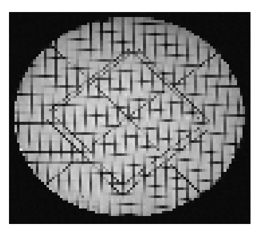
$$\mathbf{s} = \lambda(\mathbf{a}-\mathbf{d}) + \mathbf{d}$$

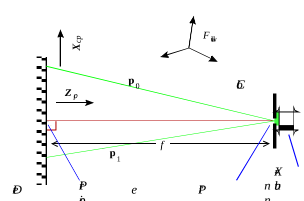
$$\lambda = \frac{(\mathbf{a}-\mathbf{d})\bullet(\mathbf{s}-\mathbf{d})}{(\mathbf{a}-\mathbf{d})\bullet(\mathbf{a}-\mathbf{d})}$$

$$\mathbf{u} = \mathbf{f}(\mathbf{d},\overline{v})$$

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

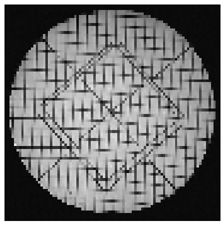




Prior studies: Boone et al., 1991; Fahrig et al., 1997; Yaniv et al., 1998; Yao, 2002; Daly et al., 2008; …

Sadowsky, 2008

C-arm Calibration: Instruments, Methods and Results

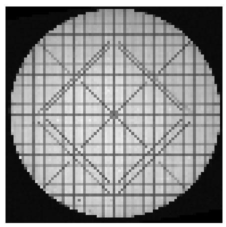


Initial x-ray of phantom

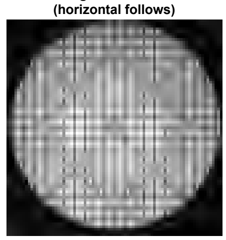
Vertical grid lines detected (horizontal follows)

Calibration phantom mounted on image intensifier

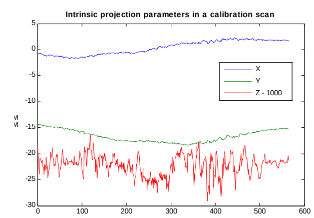
Phantom Design: Iulian Iordachita, Ofri Sadowsky Russ Taylor



Rectified phantom image



Diamond patterns detected → cone-beam parameters



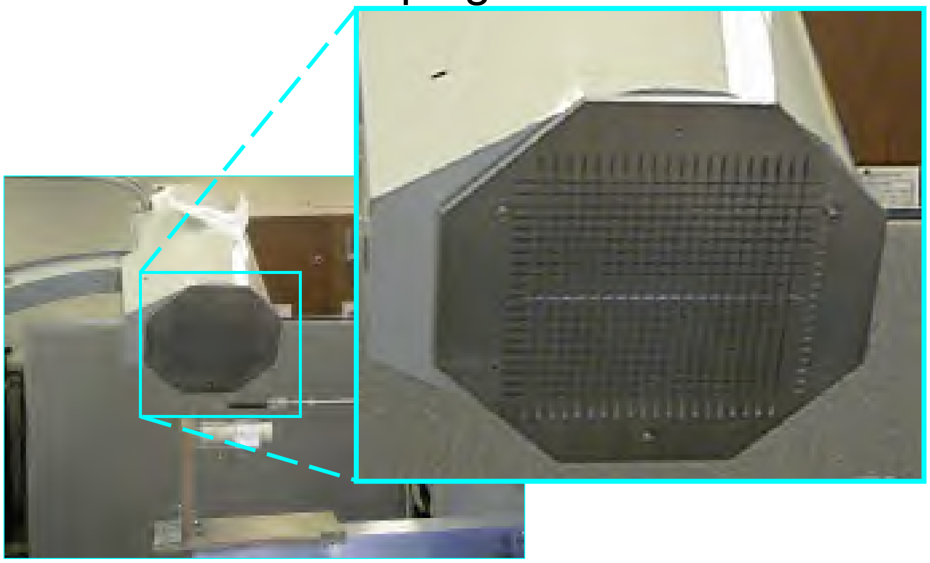
Sadowsky, 2008

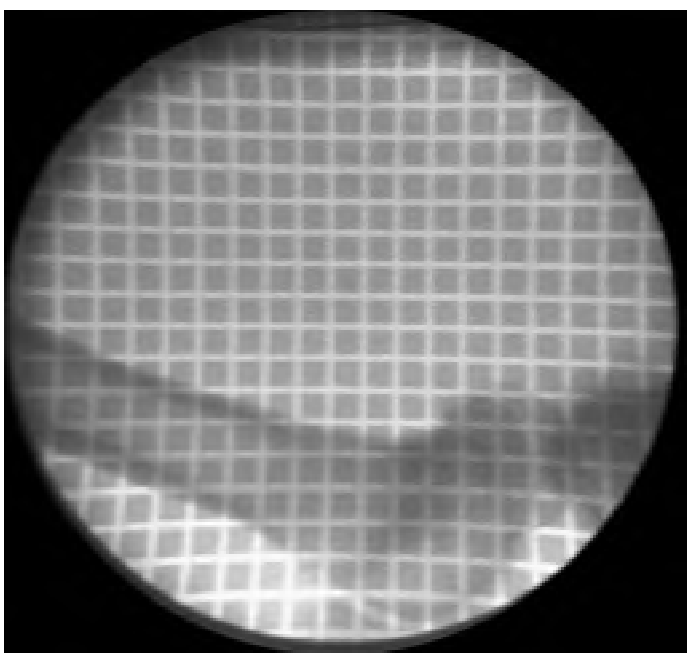
Interpolation

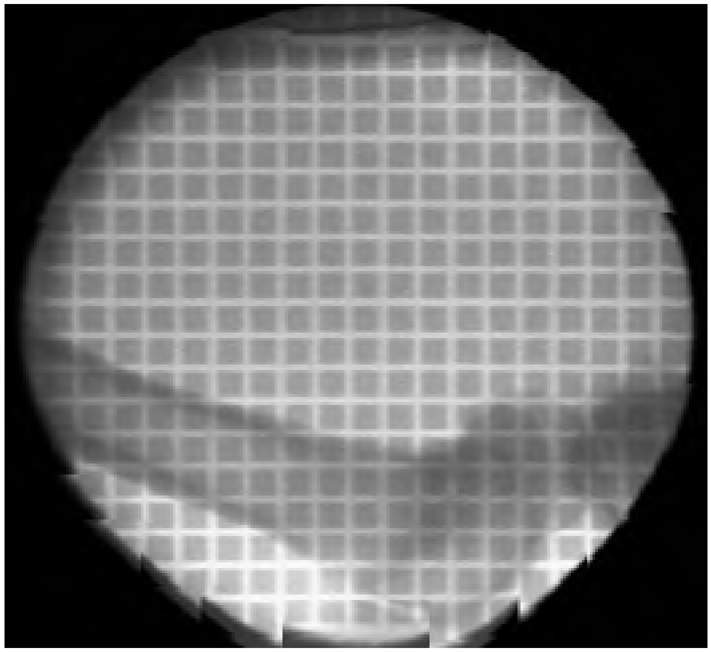
- Ubiquitous throughout CIS research and applications
- Many techniques and methods
- Here are a few more notes

Click here for Interpolation.ppt

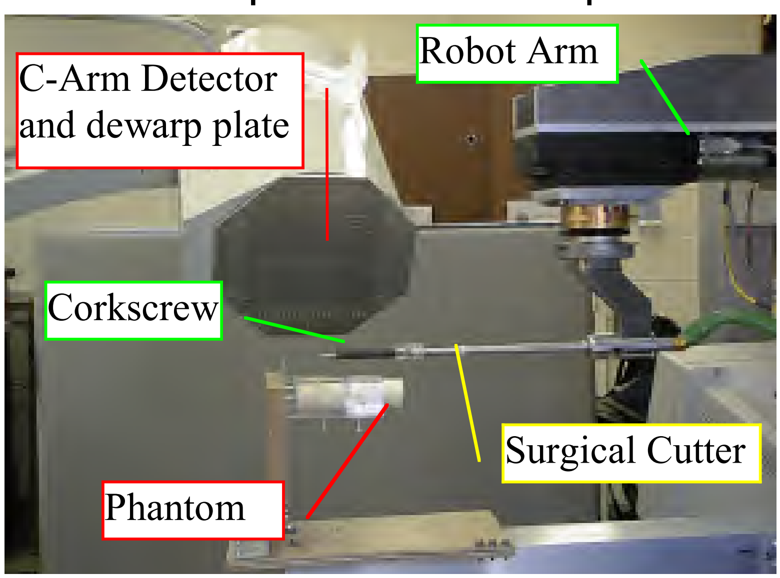
Dewarping Method



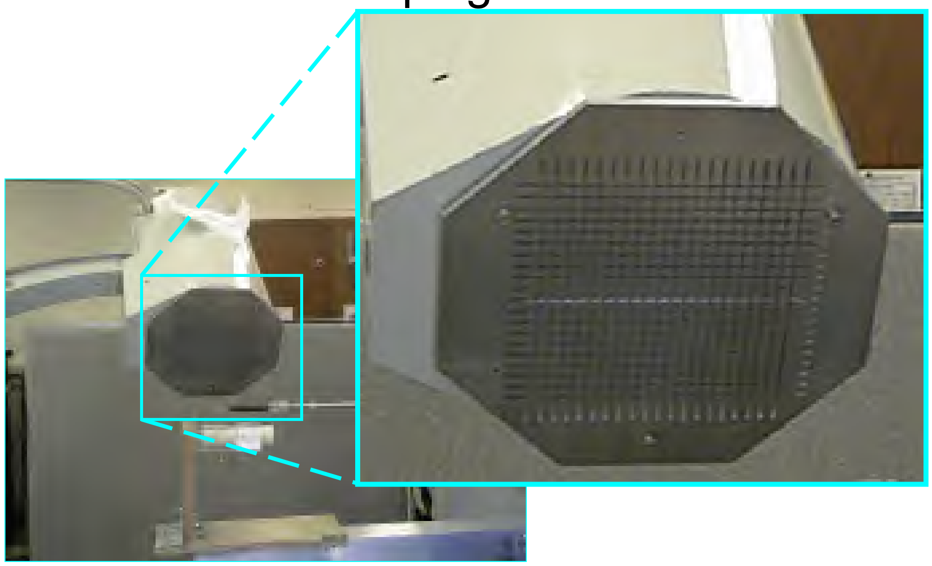




Experimental Setup

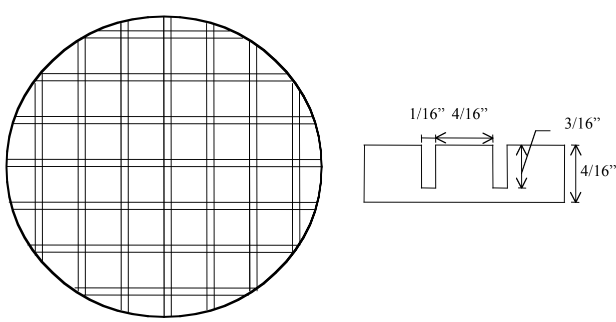


Dewarping Method



Intrinsic Image Calibration

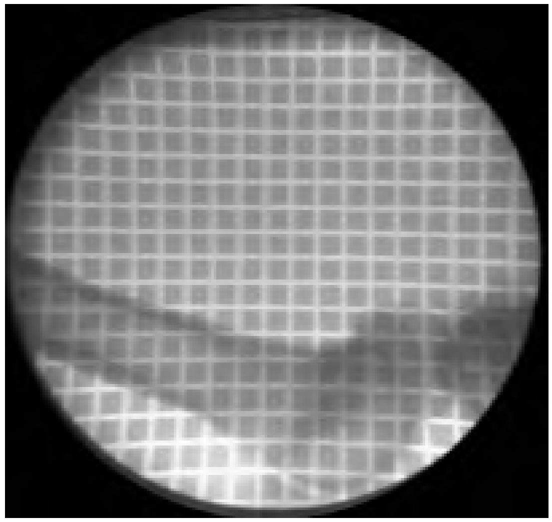
- Intrinsic imaging parameters (Schreiner et. al.)
- Image Warping (Checkerboard Based Method)



Top View

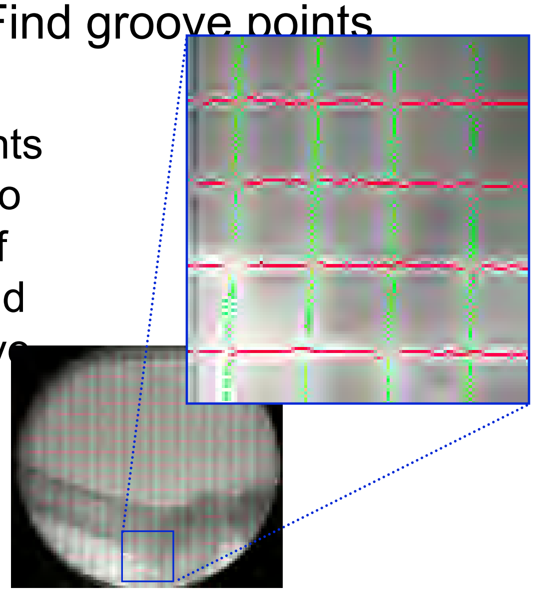
Side View

Step 0: Acquire Image



Step 1: Find groove points

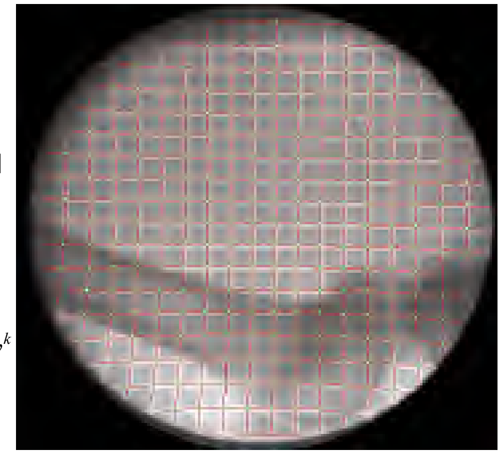
- Find image points corresponding to the centerline of each vertical and horizontal groove



Step 2: Fit 5'th order Bernstein Polynomial Curves

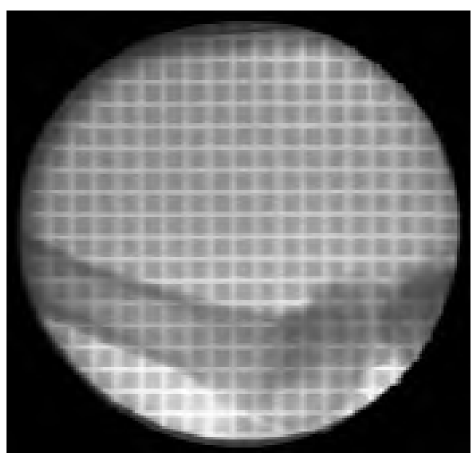
- Fit least square smooth curve to each vertical and horizontal groove
- 5'th order Bernstein Polynomial

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



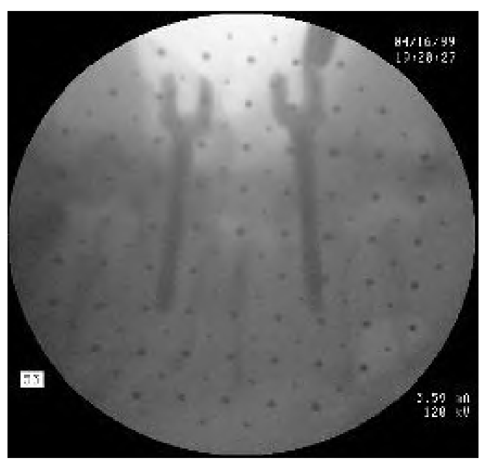
Step 3: Dewarp

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

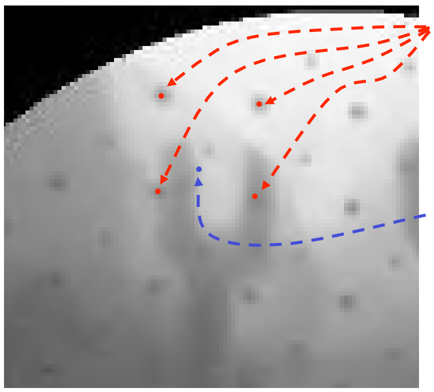


Advantages

- **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*



Photos: Sofamor Danek



Spheres i, j :

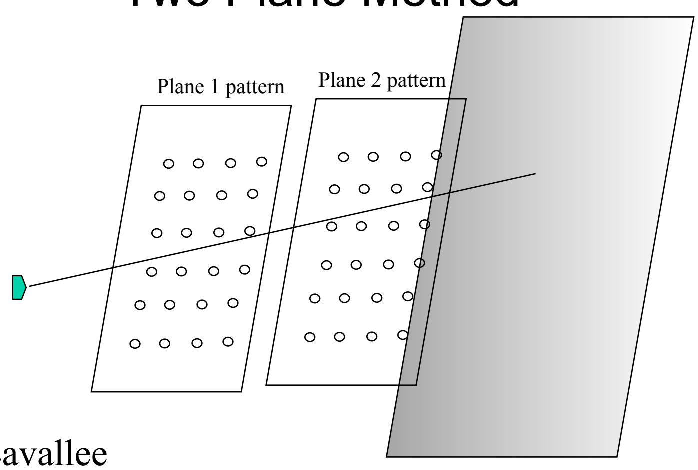
physical location in plate = $\vec{\mathbf{b}}_{ij}$

Image location = $\vec{\mathbf{u}}_{ij}$

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

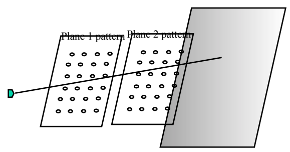
Photos: Sofamor Danek

Two Plane Method



- E.g., Lavallee
- E.g., Helm

Two Plane Method



Given **q** = a point in image coordinates, determine the points

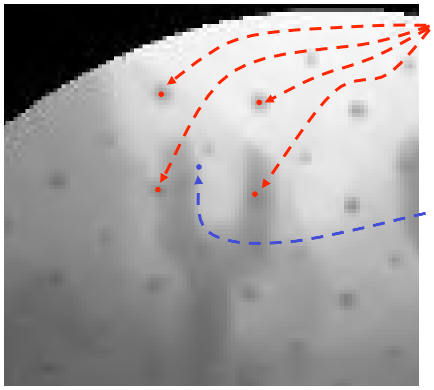
$\mathbf{f}_1^*$ = the point on grid 1 corresponding to **q**

$\mathbf{f}_2^*$ = the point on grid 2 corresponding to **q**

The desired ray in space passes through $\mathbf{f}_1^*$ and $\mathbf{f}_2^*$.

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 = $\vec{\mathbf{b}}_{ij}$

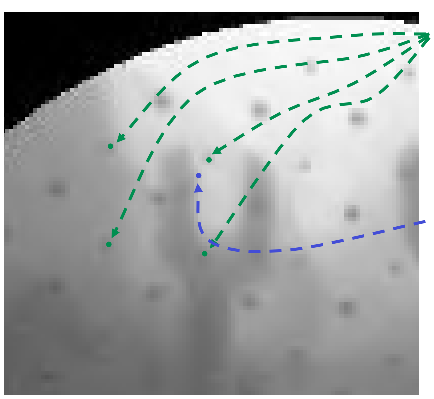
Image location = $\vec{\mathbf{u}}_{ij}$

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

Photos: Sofamor Danek

$$[\lambda,\mu] \leftarrow \text{solve}(\vec{\mathbf{u}}_t = \text{bilinear}(\lambda,\mu,\{\vec{\mathbf{u}}_{ij}\}))$$

$$\vec{\mathbf{b}}_t \leftarrow \text{bilinear}(\lambda,\mu,\{\vec{\mathbf{b}}_{ij}\})$$



Spheres i,j :

physical location in other plate = $\vec{\mathbf{c}}_{ij}$

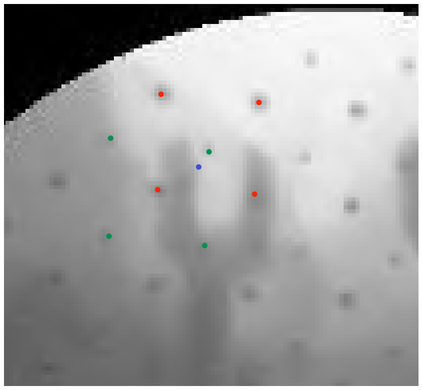
Image location = $\vec{\mathbf{u}}_{ij}^{(c)}$

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

$$[\lambda,\mu] \leftarrow \text{solve}(\vec{\mathbf{u}}_t = \text{bilinear}(\lambda,\mu,\{\vec{\mathbf{u}}_{ij}^{(c)}\}))$$

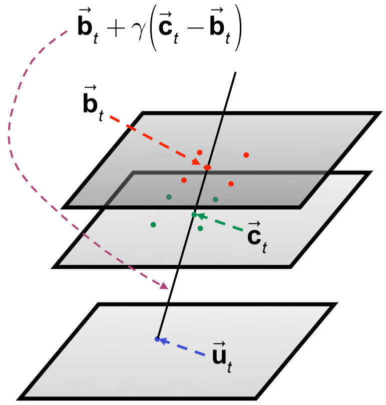
$$\vec{\mathbf{c}}_t \leftarrow \text{bilinear}(\lambda,\mu,\{\vec{\mathbf{c}}_{ij}\})$$

Photos: Sofamor Danek



So the points in space on the line from the x-ray source to detector corresponding to the image coordinates $\vec{\mathbf{u}}_t$ will be given by

$$\vec{\mathbf{b}}_t + \gamma\left(\vec{\mathbf{c}}_t - \vec{\mathbf{b}}_t\right)$$



Photos: Sofamor Danek