Essential Math for CIS
Part 2: Transformations and Frames of Reference

Gabor Fichtinger, PhD

Associate Research Professor of Computer Science, Mechanical Engineering and Radiology

NSF-Funded Engineering Research Center for Computer-Integrated Surgical Systems and Technology,
Johns Hopkins University

Copyright © CISST ERC, 2001

Matrices

An ordered table of numbers (or sub-tables)

columns

$$\underbrace{\mathbf{A}}_{3\times 3} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix}$$

Rows

$$\text{Product} \quad : \mathbf{C} = \mathbf{BA} \neq \mathbf{AB} \, ,$$

$$\mathrm{C} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} a_{00} * b_{00} + a_{01} * b_{10} & a_{00} * b_{01} + a_{01} * b_{11} \\ a_{10} * b_{00} + a_{11} * b_{10} & a_{10} * b_{01} + a_{11} * b_{11} \end{bmatrix}$$

$$\text{Identity} \quad : \mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Copyright © CISST ERC, 2001

Matrix Addition (+)

Definition:

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix} + \begin{bmatrix} x & u \\ y & v \end{bmatrix} = \begin{bmatrix} a+x & c+u \\ b+y & d+v \end{bmatrix}$$

Example:

$$\begin{bmatrix} 1 & 5 \\ 2 & 6 \end{bmatrix} + \begin{bmatrix} 0 & -2 \\ 1 & -7 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 3 & -1 \end{bmatrix}$$

Properties:

1. Closed result is always a matrix
2. Commutative $A+B = B+A$
3. Associative $(A+B)+C = A+(B+C)$
4. Identity $A+0 = A$ (null matrix)
5. Inverse $A+A^{-}=0$ (negated matrix)

Copyright © CISST ERC, 2001

Matrix Multiplication (*)

Definition:

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix} * \begin{bmatrix} x & u \\ y & v \end{bmatrix} = \begin{bmatrix} ax+cy & au+cv \\ bx+dy & bu+dv \end{bmatrix}$$

Example:

$$\begin{bmatrix} 1 & 5 \\ 2 & 6 \end{bmatrix} + \begin{bmatrix} 0 & -2 \\ 1 & -7 \end{bmatrix} = \begin{bmatrix} 5 & -37 \\ 6 & -46 \end{bmatrix}$$

Properties:

1.	Closed	result is always a matrix
2.	NOT Commutative	A*B != B*A
3.	Associative	(A*B)*C = A*(B*C)
4.	Identity	A*I = A (identity matrix)
5.	Inverse	$A*A^{-}=I$ (difficult to obtain!)

Distributive from either side

(A+B)C = AC + BC

C(A+B) = CA + CB

Copyright © CISST ERC, 2001

Vector is a special matrix

Examples:

v=[x,y,z] row vector, 1x3 matrix

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$ column vector, 2x1 matrix

Copyright © CISST ERC, 2001

Vector addition: just like adding matrices

Column vector

$$\begin{bmatrix} a & \cancel{c} \\ b & \cancel{d} \end{bmatrix} + \begin{bmatrix} x & \cancel{u} \\ y & \cancel{v} \end{bmatrix} = \begin{bmatrix} a+x & \cancel{c+u} \\ b+y & \cancel{d+v} \end{bmatrix}$$

Row vector

$$\begin{bmatrix} a & c \\ \cancel{b} & \cancel{d} \end{bmatrix} + \begin{bmatrix} x & u \\ \cancel{y} & \cancel{v} \end{bmatrix} = \begin{bmatrix} a+x & c+u \\ \cancel{b+y} & \cancel{d+v} \end{bmatrix}$$

Copyright © CISST ERC, 2001

Vector* Matrix Multiplication:
just like multiplying matrices

Row vector on the left

$\mathbf{v_1} = [\mathrm{x}_1, \mathrm{y}_1]$

$$\mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$\mathbf{v_2} = \mathbf{v_1 M}$

$$\mathbf{v_2} = [\mathrm{x}_1, \mathrm{y}_1] \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [ax_1 + by_1 \quad , \quad cx_1 + dy_1]$$

Result is a new row vector

Copyright © CISST ERC, 2001

Matrix*Vector Multiplication:
just like multiplying matrices

$$\mathbf{v}_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Column vector on the right

$$\mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\mathbf{v}_2 = \mathbf{M}\mathbf{v}_1$$

$$= \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} ax_1 + by_1 \\ cx_1 + dy_1 \end{bmatrix}$$

Result is a new column vector

Copyright © CISST ERC, 2001

Matrix*Vector Multiplication(example)

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} 5 & 1 \\ 2 & 3 \end{bmatrix}$$

$$\begin{aligned} \mathbf{v}_2 &= \mathbf{M}\mathbf{v}_1 \\ &= \begin{bmatrix} 5 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 5*1+1*2 \\ 2*1+3*2 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \end{bmatrix} \end{aligned}$$

Copyright © CISST ERC, 2001

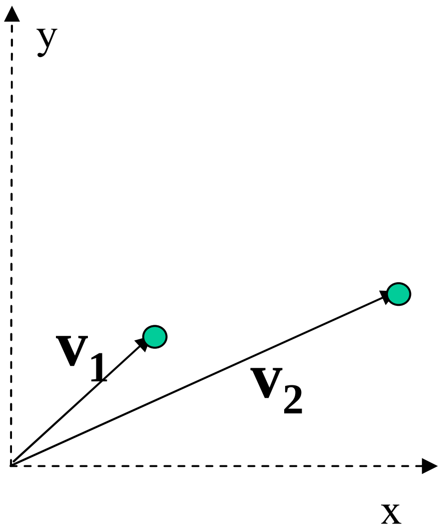
Matrices as transformations

Matrix-vector multiplication creates a new vector.
New vector means new location in space.
If all points of an object is multiplied by a matrix,
then the whole object assumes a new position (and may be new shape and size, too). This is called transformation.

$$\mathbf{v}_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\mathbf{v}_2 = \mathbf{M}\mathbf{v}_1$$



Copyright © CISST ERC, 2001

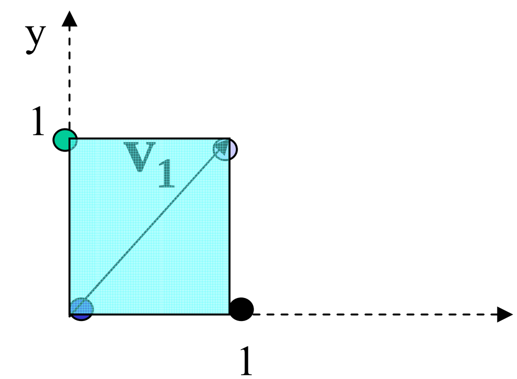
Matrices as transformations (examples)

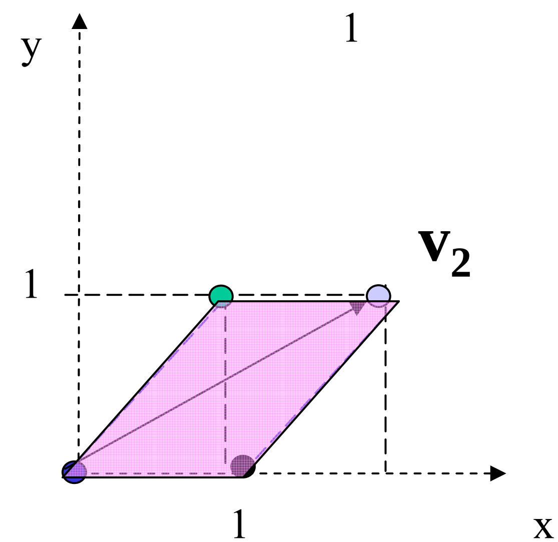
Transform 4 points [0,0], [1,0],[1,1],[0,1] using the matrix M.

$$\mathbf{M} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{v}_2 = \mathbf{M}\mathbf{v}_1$$

$$\mathbf{v}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \qquad \mathbf{v}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$





Copyright © CISST ERC, 2001

Scaling

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\mathbf{v}_2 = \mathbf{S}\mathbf{v}_1$$

$$= \begin{bmatrix} 5 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 5*1+0*2 \\ 0*1+3*2 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \qquad \mathbf{S}^{-1} = \begin{bmatrix} 1/s_x & 0 \\ 0 & 1/s_y \end{bmatrix} \qquad \mathbf{S} * \mathbf{S}^{-1} = \mathbf{I}$$

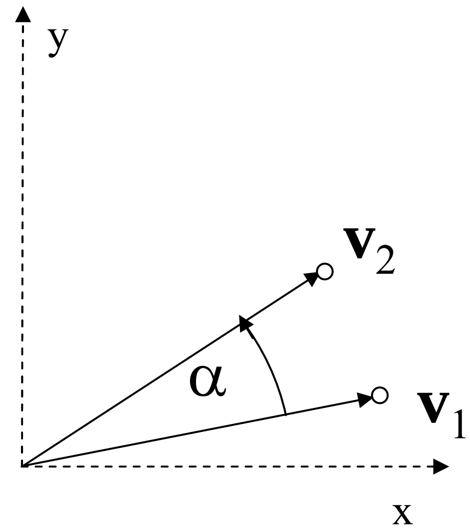
$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$$ 3D scaling matrix

Copyright © CISST ERC, 2001

Rotation

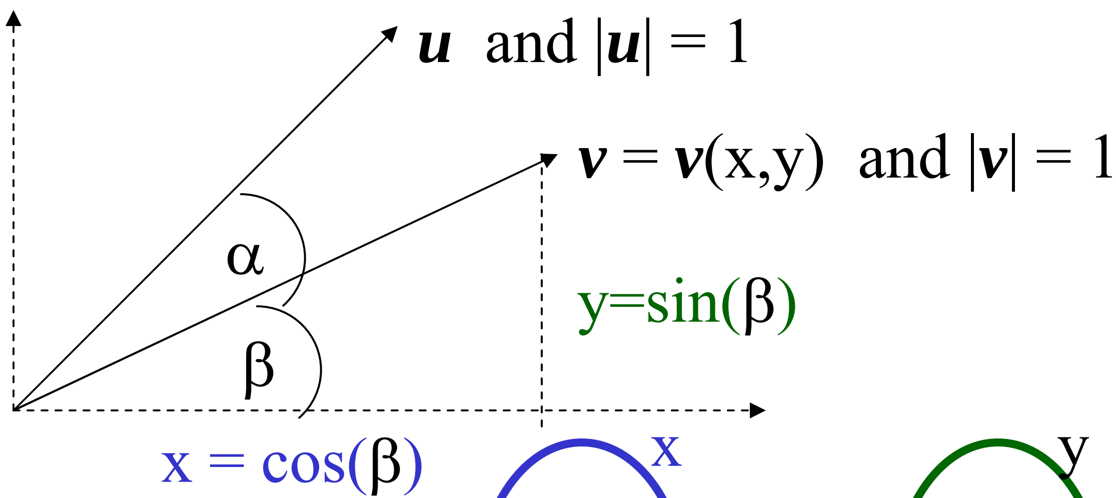
$$\mathbf{v}_2 = \mathbf{R}_\alpha \mathbf{v}_1$$

$$\mathbf{v}_1 = \mathbf{R}_{-\alpha} \mathbf{v}_2$$



Copyright © CISST ERC, 2001

Rotation Matrix



$$\bar{\boldsymbol{u}} = \begin{bmatrix} \cos(\alpha+\beta) \\ \sin(\alpha+\beta) \end{bmatrix} = \begin{bmatrix} \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) \\ \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta) \end{bmatrix} =$$

x y

$$\begin{bmatrix} \cos(\alpha)\,x - \sin(\alpha)\,y \\ \sin(\alpha)\,x + \cos(\alpha)\,y \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad \bar{\boldsymbol{v}}$$

$$R_\alpha = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

Copyright © CISST ERC, 2001

Inverse of Rotation Matrix

$$R_{\alpha} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \rightarrow R_{-\alpha} = \begin{bmatrix} \cos(-\alpha) & -\sin(-\alpha) \\ \sin(-\alpha) & \cos(-\alpha) \end{bmatrix} =$$

$$R_{-\alpha} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$R_{\alpha}^{-1} = R_{-\alpha}$ The proof: $R_{\alpha} * R_{-\alpha} = I$

Series of Rotations

$$R_{(\alpha+\beta+\gamma)} = R_{\alpha} R_{\beta} R_{\chi}$$

Copyright © CISST ERC, 2001

3D Rotation Matrices

Rotation by α around z axis

$$R_\alpha = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{v}_2 = R_\alpha \boldsymbol{v}_1$$

When applied on vector $\boldsymbol{v}_1$, it does not change z coordinate. Try it!

Rotation around all axes

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \qquad R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \qquad R_z(\gamma) = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

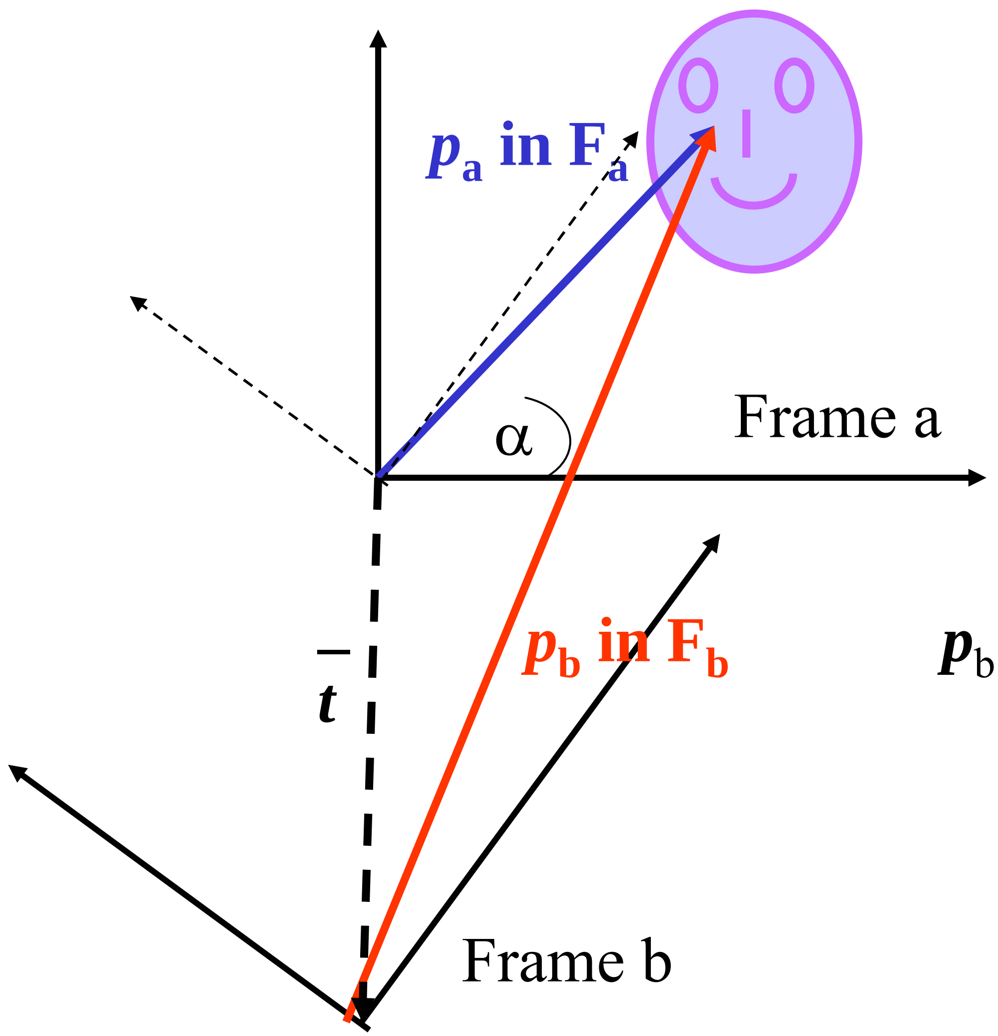
Copyright © CISST ERC, 2001

Rotation Matrix: example

$$\mathbf{R}_{60} = \begin{bmatrix} 1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{bmatrix}$$

Copyright © CISST ERC, 2001

Passage from reference system to another



$$p_b = RSp_a + t$$

(1) scale, (2) rotate, and (3) translate to go from F_a to F_b

Copyright © CISST ERC, 2001

Change of reference systems backwards (inverse transform)

$\boldsymbol{p}_b = RS\boldsymbol{p}_a + \boldsymbol{t}$ //subtract $\boldsymbol{t}$

$\boldsymbol{p}_b - \boldsymbol{t} = Rs\boldsymbol{p}_a$ //left multiply by R^{-1}

$R^{-1}(\boldsymbol{p}_b - \boldsymbol{t}) = S\boldsymbol{p}_a$ //left multiply by S^{-1}

$S^{-1}R^{-1}(\boldsymbol{p}_b - \boldsymbol{t}) = \boldsymbol{p}_a$

Slight problem: mixes translation vector with matrices…
Solution: make $\boldsymbol{t}$ translation vector appear as matrix T in multiplications. Then the equations will read as:

$$\boldsymbol{p}_b = (T(R(S\boldsymbol{p}_a)))$$

Copyright © CISST ERC, 2001

Introduce 4x4 Translation Matrix

$$T = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad v = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Padding with 1 (If I padded with 0 then inverse would not exist!)

$$Tv = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x+d_x \\ y+d_y \\ z+d_z \\ 1 \end{bmatrix}$$

Problem: rotation and scaling matrices must also be padded, so that we can multiply all 4x4 matrices.

Copyright © CISST ERC, 2001

Homogeneous Matrices and Translation Vector

Rotation matrix by α around z axis

$$R_\alpha = \begin{bmatrix} \cos(a) & -\sin(a) & 0 & 0 \\ \sin(a) & \cos(a) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translation matrix

$$T = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling matrix

$$S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{v} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Padding all vectors with 1

$$\boldsymbol{p}_b = (T(R(S\boldsymbol{p}_a)))$$

Copyright © CISST ERC, 2001

Series of transformations

Inverse transformation:

$Ap=q$ //apply A on p

$A^{-1}Ap = A^{-1}q$ //mult both sides from left by A^{-1}

$p=A^{-1}q$ //because $A^{-1}A=I$

Series of transformations

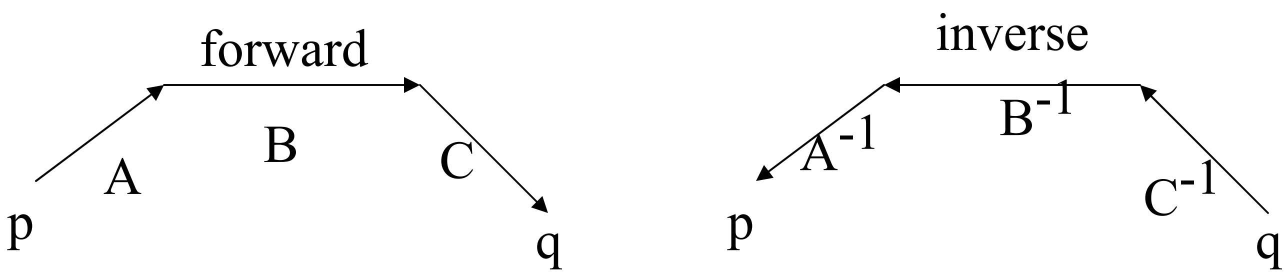
$CBAp=q$ //Apply A,B, then C on p

$(C\ (B\ (Ap))) = q$ //use associative property

Inverse of series of transformations

$CBAp=q$ //Apply A,B, then C on p

$A^{-1}B^{-1}C^{-1}q=p$ //multiply by inverse from left



Copyright © CISST ERC, 2001

Inverse of coordinate transformations

$$\boldsymbol{p}_b = TRS\boldsymbol{p}_a$$

Multiply from the left by T^{-1}

$$T^{-1}\,\boldsymbol{p}_b = RS\boldsymbol{p}_a$$

Multiply from the left by R^{-1}

$$R^{-1}T^{-1}\,\boldsymbol{p}_b = S\boldsymbol{p}_a$$

Multiply from the left by S^{-1}

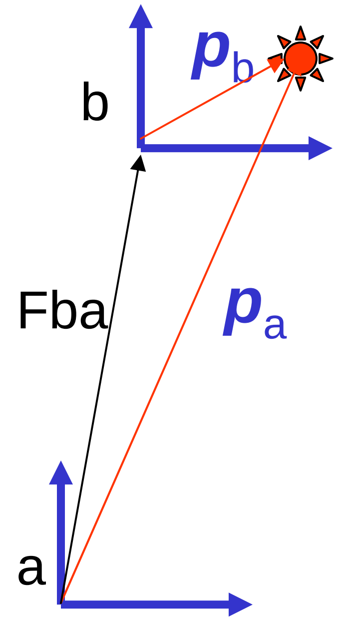
$$S^{-1}R^{-1}T^{-1}\boldsymbol{p}_b = \boldsymbol{p}_a$$

Copyright © CISST ERC, 2001

Reference frame transformations

$$\boldsymbol{p}_b = T*R*S*\ \boldsymbol{p}_a$$

$$\boldsymbol{p}_b = F_{ba}\boldsymbol{p}_a$$



F_{ba} is often called frame transformation or shortly frame

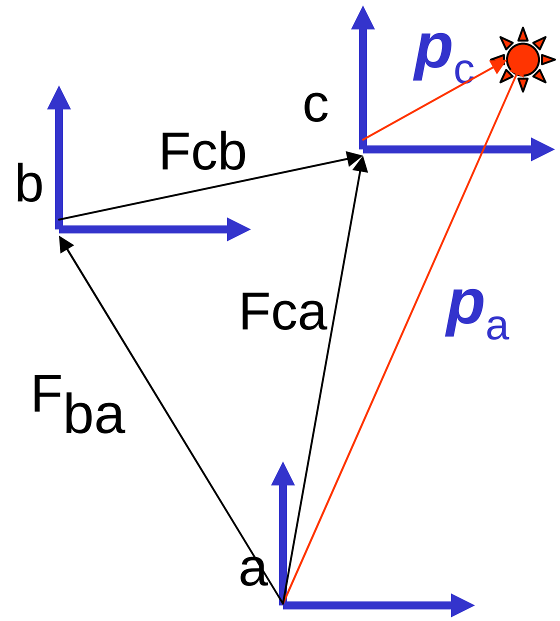
Copyright © CISST ERC, 2001

Series of reference frame transformation

Most real systems involve complicated series of frames, where we traverse from frame to frame…

$$\overbrace{F_{cb}F_{ba}}^{F_{ca}}$$

$$\boldsymbol{p}_c = F_{cb}F_{ba}\,\boldsymbol{p}_a$$

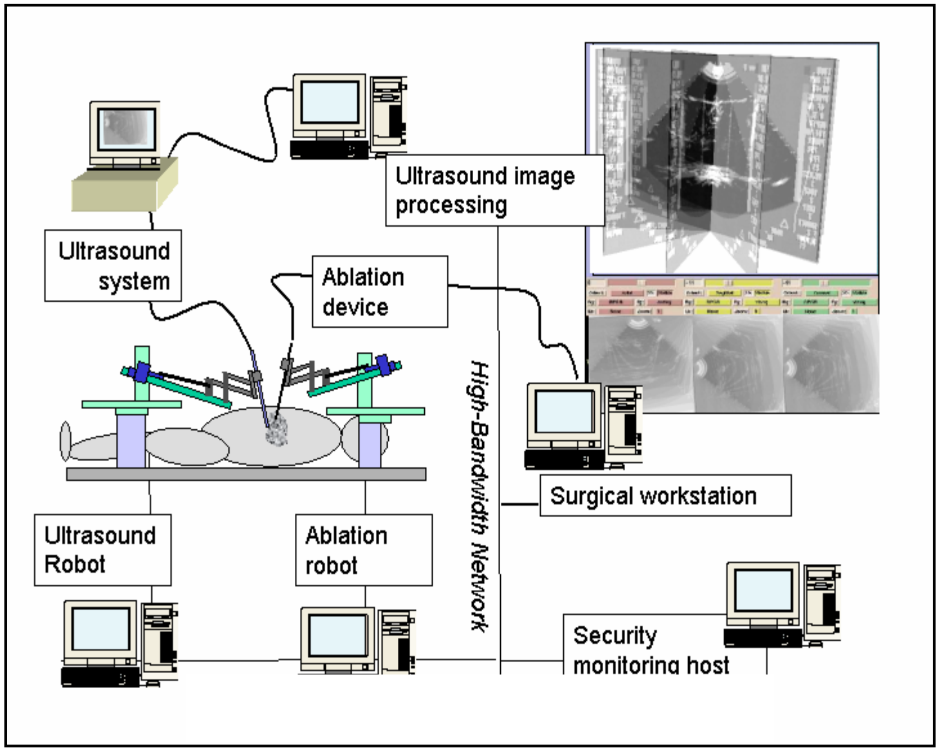


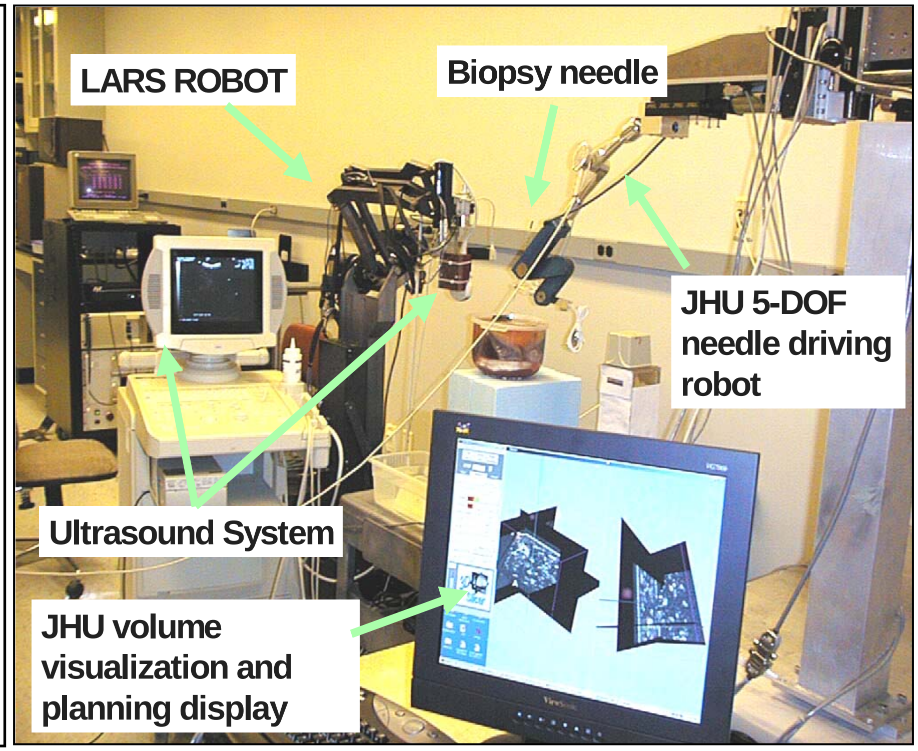
All frames must be known: some of them are measured/calculated during the procedure, some of them are pre-operatively known through “calibration”.

Copyright © CISST ERC, 2001

Example: Robot-Assisted Ultrasound-Guided Liver Surgery

R. Taylor, G. Fichtinger, C. Burdette, M. Choti



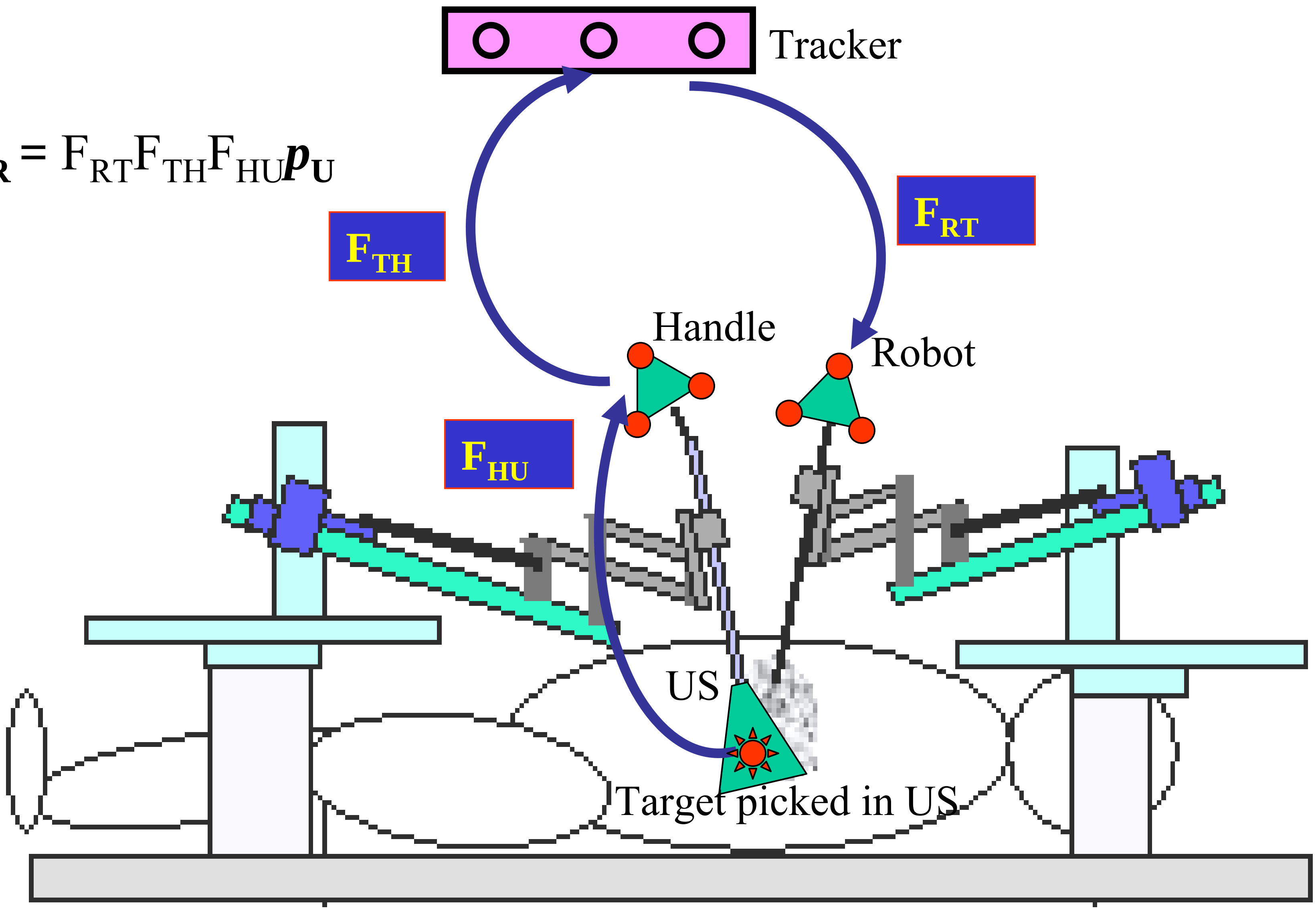


Credit: Emad Boctor, doctoral student,

Copyright © CISST ERC, 2001

Example cont'd: Some frames in the tracked dual-arm robot system

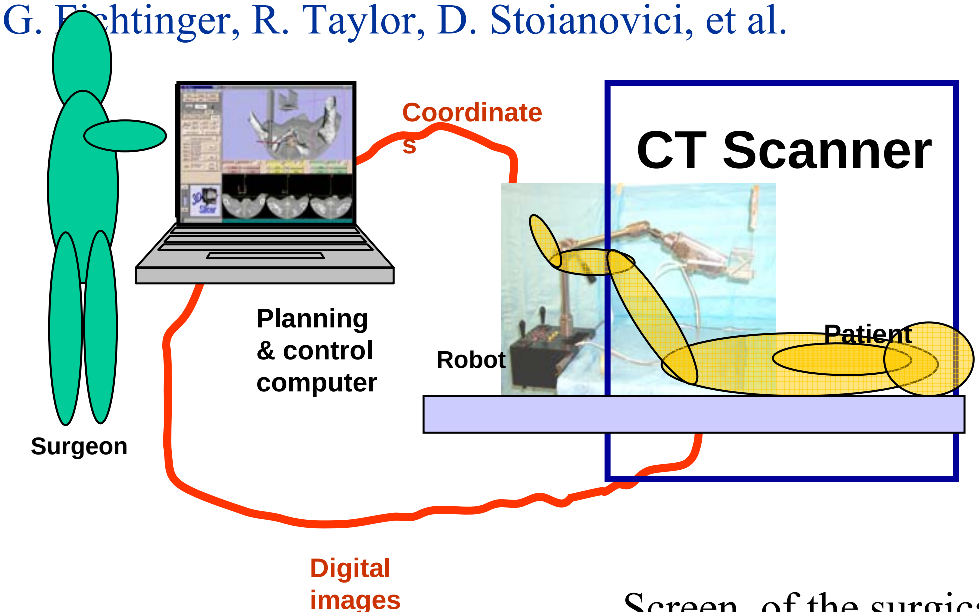
$\boldsymbol{p}_{\mathbf{R}} = F_{RT}F_{TH}F_{HU}\boldsymbol{p}_{\mathbf{U}}$



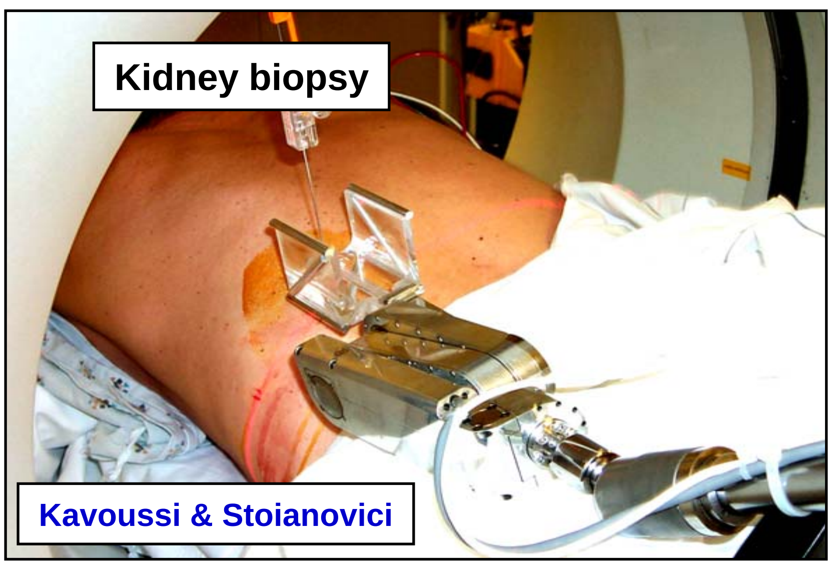
Copyright © CISST ERC, 2001

Example: CT-guided Needle Placement with Robot

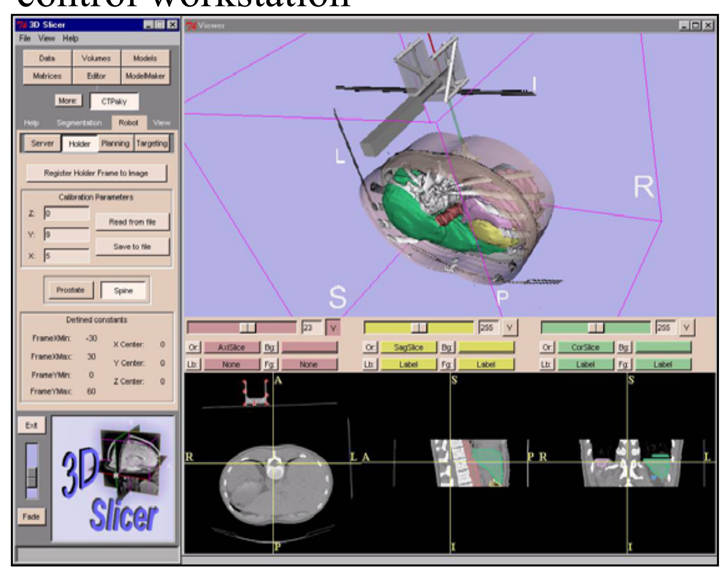
G. Fichtinger, R. Taylor, D. Stoianovici, et al.



Patient in the scanner with robot

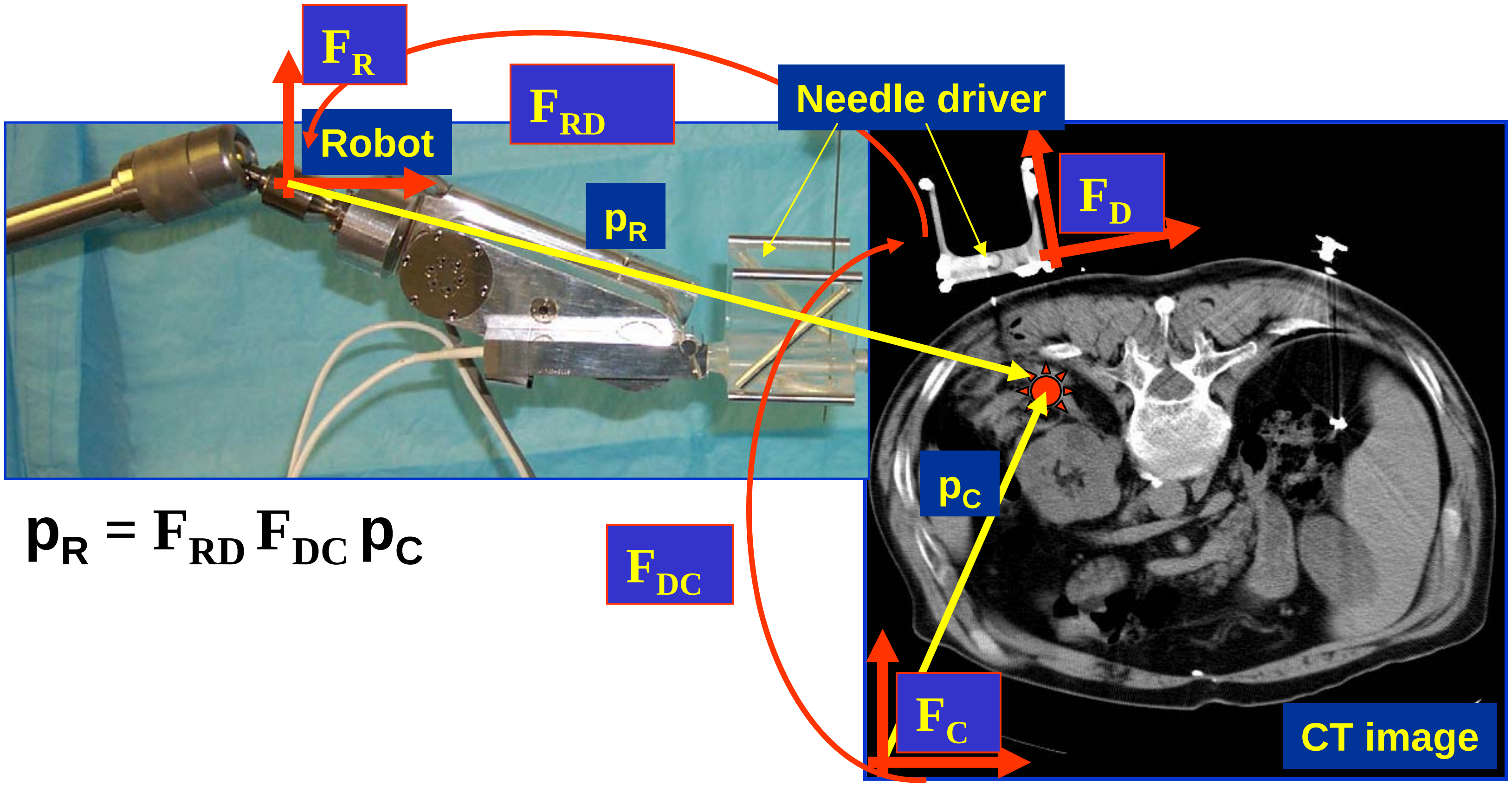


Screen of the surgical planning and control workstation



Copyright © CISST ERC, 2001

Example cont'd: Frames

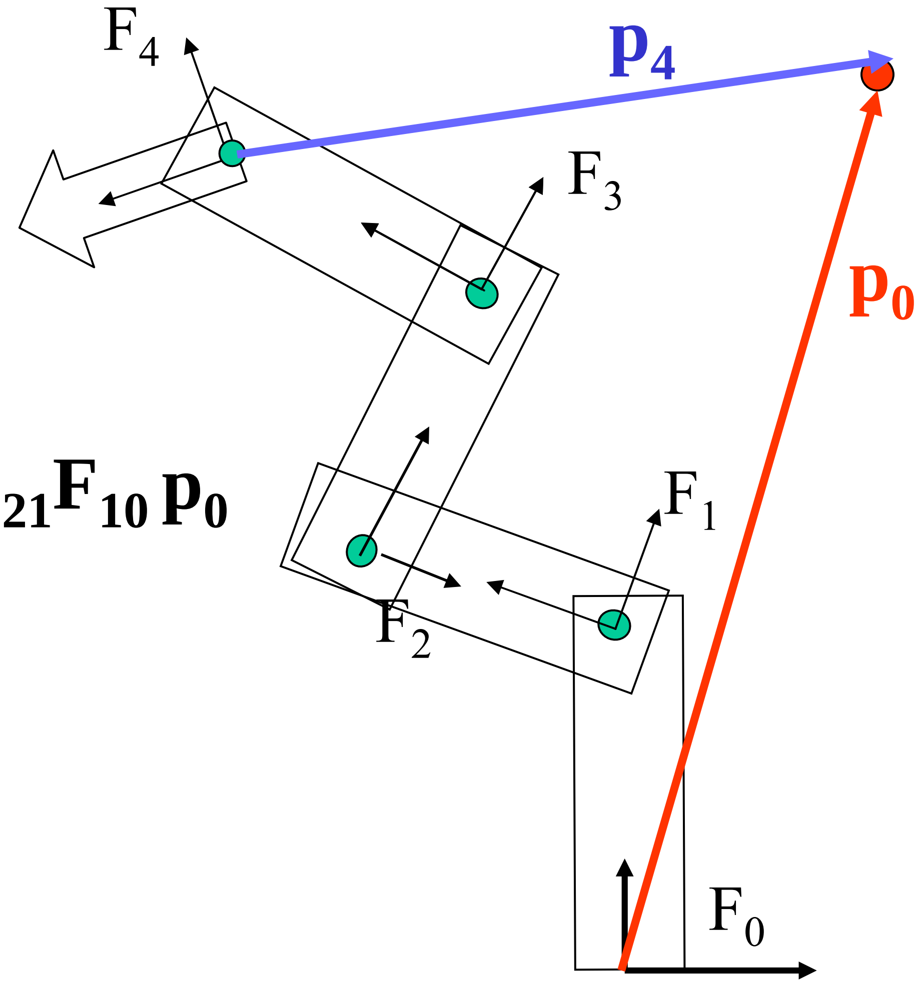


$$p_R = F_{RD}\,F_{DC}\,p_C$$

Copyright © CISST ERC, 2001

Example: Multi-Joint Serial Robots

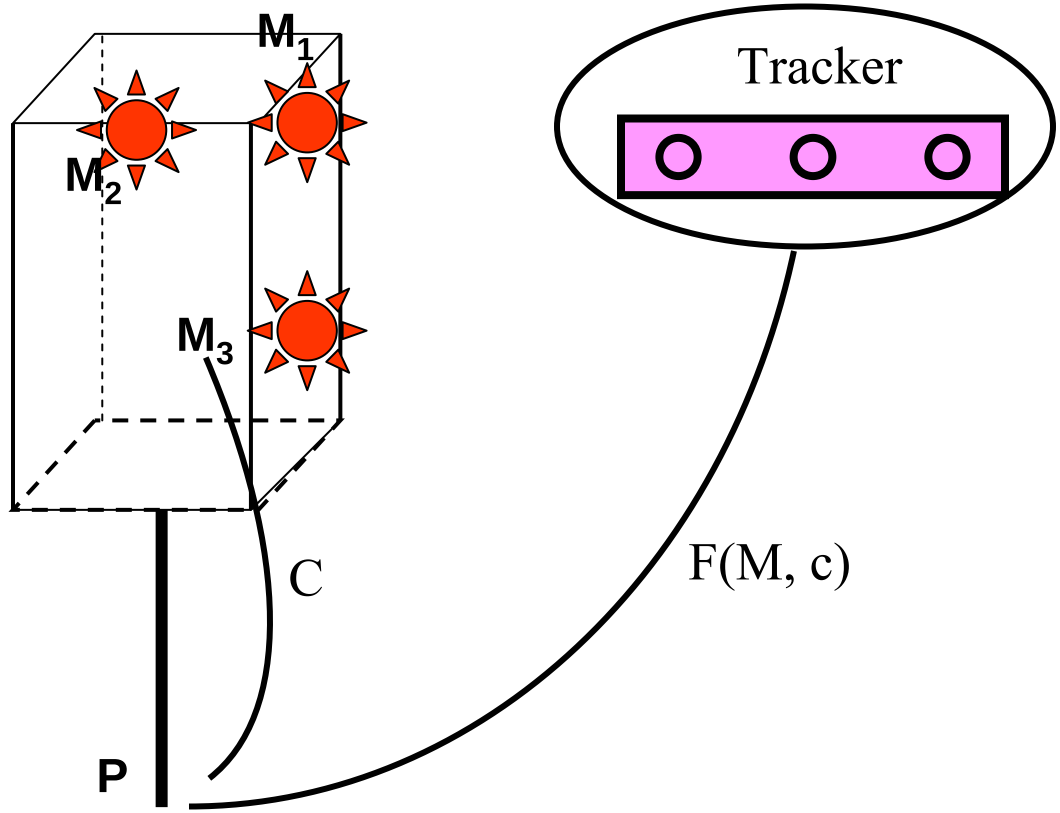
4-joint serial robot



$$p_4 = F_{43} F_{32} F_{21} F_{10} \, p_0$$

Copyright © CISST ERC, 2001

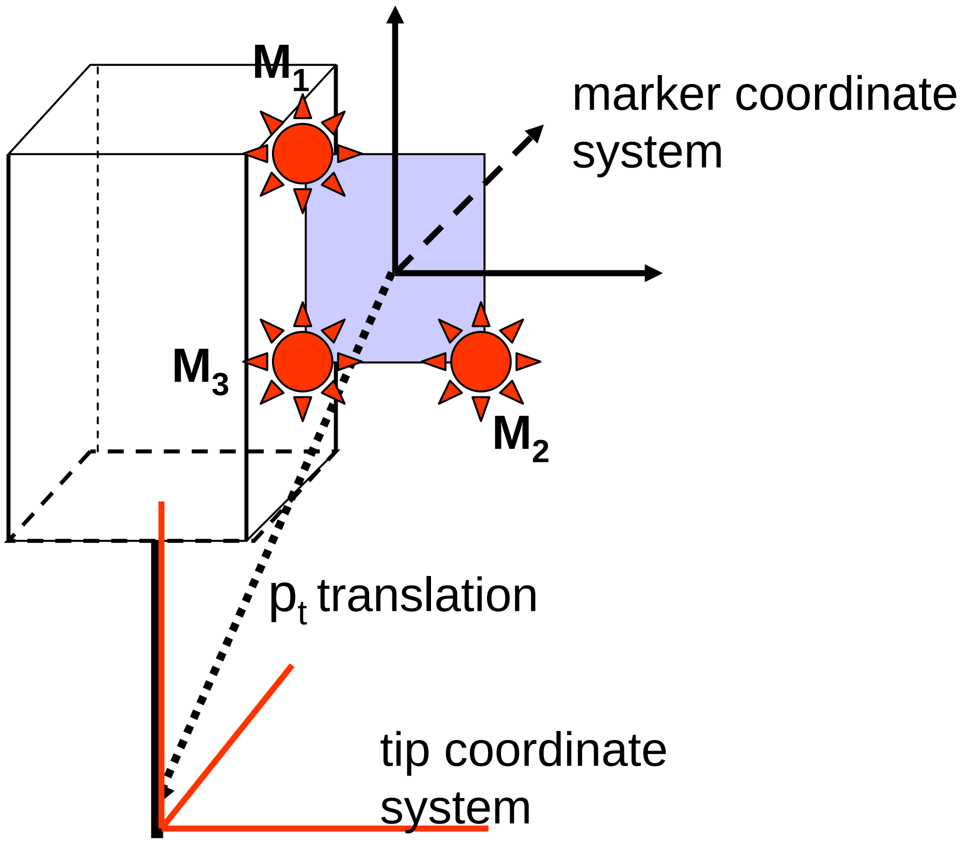
Example: Pointer Calibration



- M = (M1,M2,…Mi) tracker readings
- P = F(M, C) function to get tool tip position from tracker readings, where C = (c1,c2,c3…) is set of constants
- How to get these constants? -- CALIBRATION

Copyright © CISST ERC, 2001

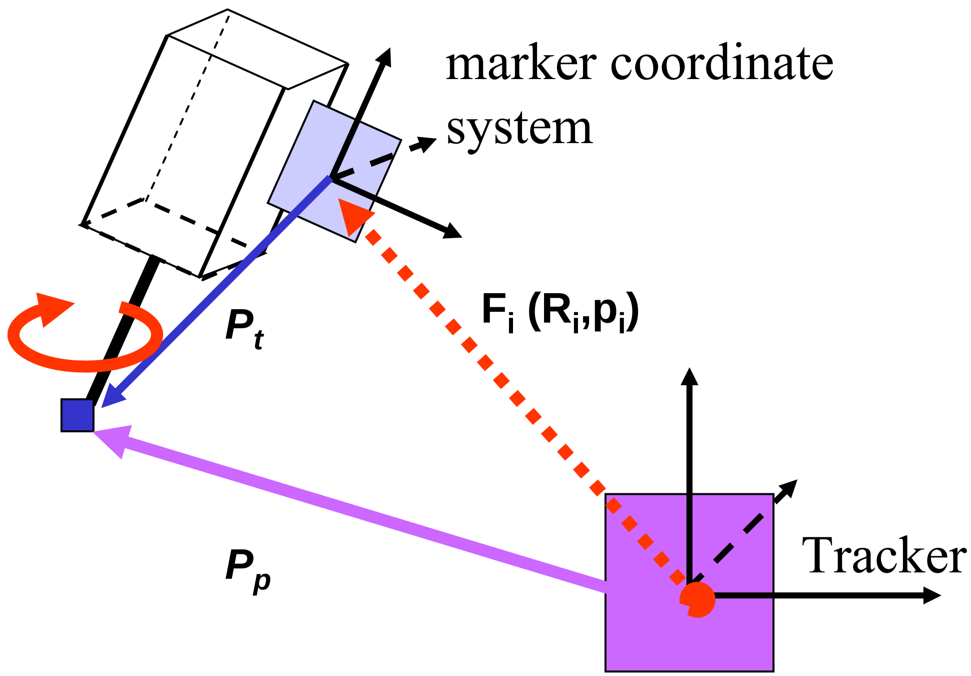
Generic Pivot Calibration



- Determine p_t translation between tip and marker coordinate system
- Pivot around a fixed point

Copyright © CISST ERC, 2001

Pivot and measure many times…



- p_t vector is constant if looking from marker coordinate system
- Pivot point is constant if looking from the tracker base
- $M_1,M_2,M_{3\ldots}\ M_n$ are reported by tracker
- $F_i\ (R_i,p_i)$ is easily calculated by software package
- $F_i\ (R_i,p_i)$ takes the p_t vector to the pivot point
- $F_i*p_t = p_p$
- First rotation by R_i, then translation by p_i
- $R_i*p_t + p_i = p_p$
- Unknowns: Pt and Pp
- Two poses are sufficient to calculate Pt
- Take many poses (i.e. redundancy) to reduce errors!!!

Copyright © CISST ERC, 2001

Solve the math…

(1) $R_i * p_t + p_i = p_p$

(2) $R_j * p_t + p_j = p_p$ subtract and 1 and 2

$R_i * p_t - R_j * p_t + p_i - p_j = 0$

$(R_i - R_j) p_t + p_i - p_j = 0$

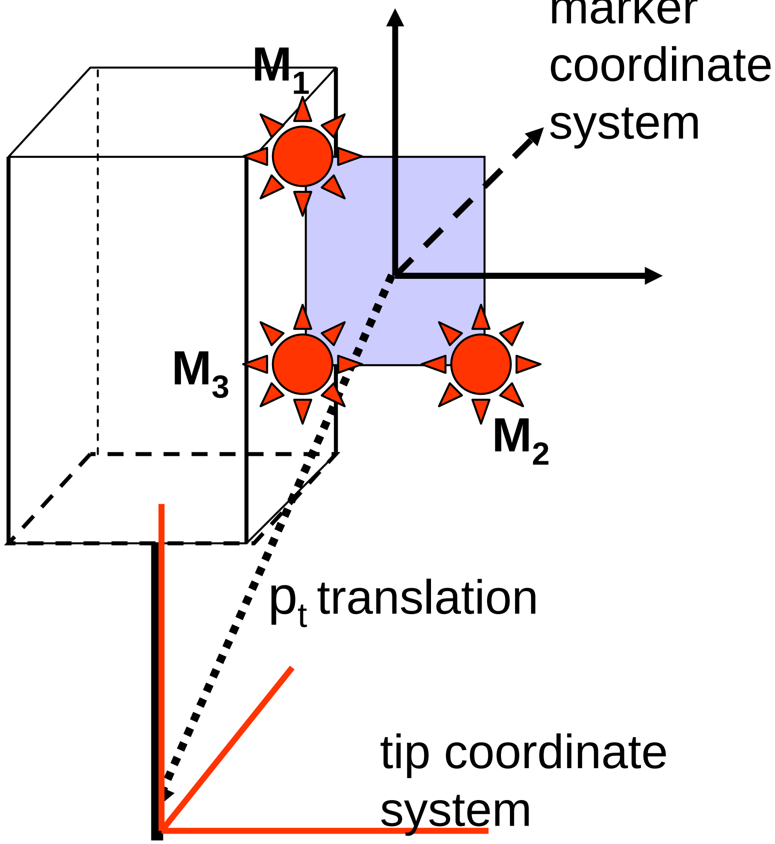
$(R_i - R_j) p_t = -(p_i - p_j)$

$p_t = -(R_i - R_j)^{-1} (p_i - p_j)$

Repeat the above on all pairs of measurements and then take the average p_t

Copyright © CISST ERC, 2001

Now how to use the pointer



- **Read M_1,M_2,$M_{3\ldots}$ M_n are reported by tracker**
- **F_i (R_i,p_i) is easily calculated by software package**
- **Plug p_t into R_i*p_t + p_i = p_p**
- **If possible, take multiple measurements, reject outliers, and average the rest**

Copyright © CISST ERC, 2001