ME 328: Medical Robotics
Winter 2019

# Lecture 8: Registration

Allison Okamura
Stanford University

# Updates

**Assignment 4**

Sign up for teams/ultrasound *by noon today* at:

**https://tinyurl.com/ME328-USlab**

Main parts:
Readings/questions
Image acquisition and analysis (ultrasound lab)
Image registration
Needle insertion (robot control)

Remainder of assignment will be posted this morning

https://tinyurl.com/ME328-USlab

# registration

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

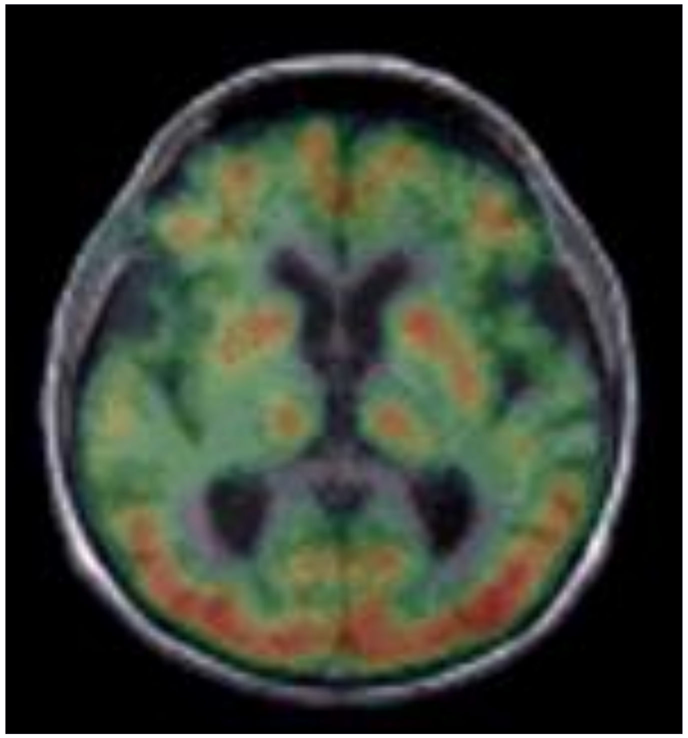
i.e., establish a quantitative relationship between different reference frames

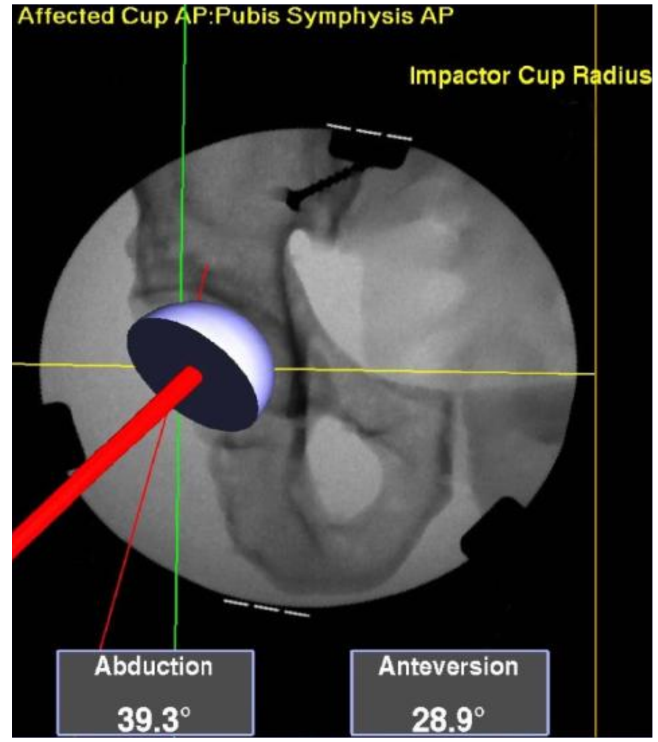
essential for the quantitative integration of various sources of information:

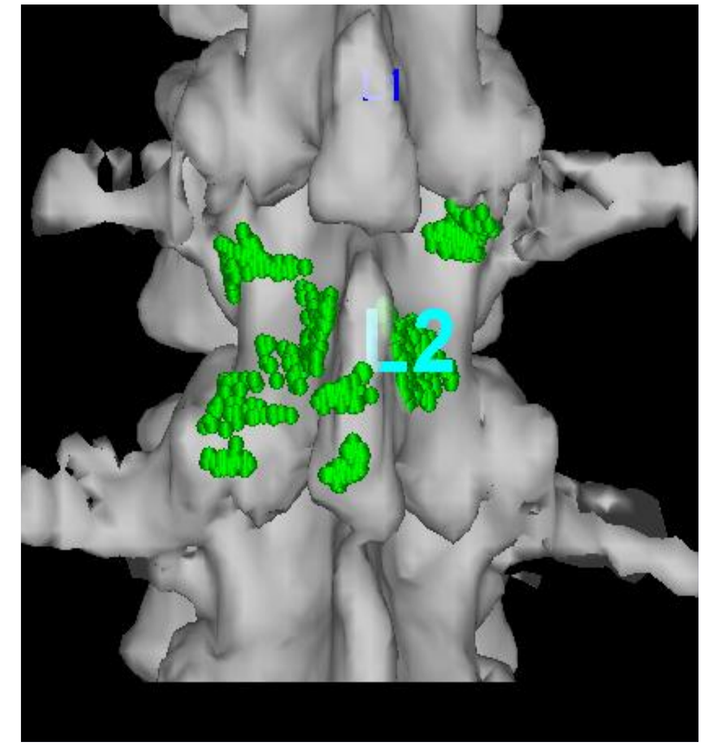
1. pre-, intra-, and post-operative images
2. trackers (tracking data not useful unless registered)
3. robots (cannot use images unless registered)

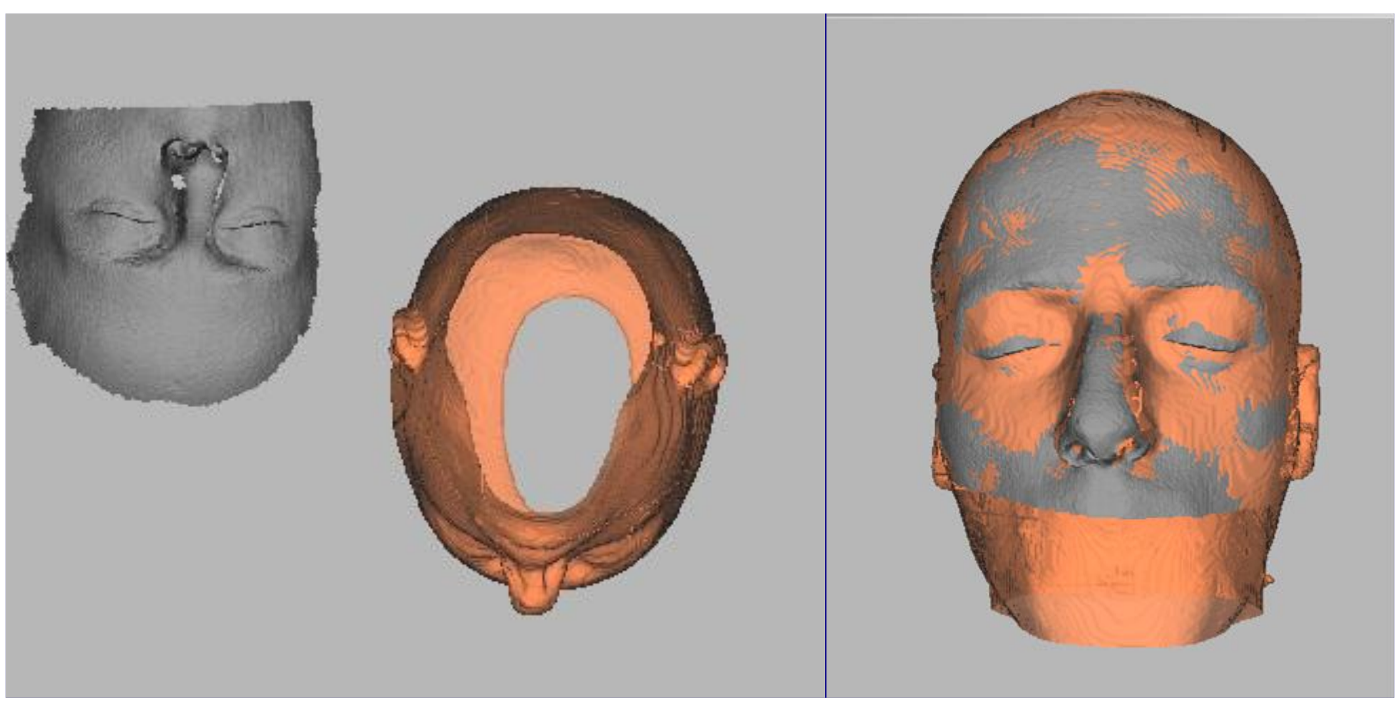
Image-Guided Procedures: A Review, by Ziv Yaniv and Kevin Cleary (2006)

# examples




Affected Cup AP:Pubis Symphysis AP
Impactor Cup Radius
Abduction
39.3°
Anteversion
28.9°


L2



# 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
  ~1 mm typically satisfactory
- 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

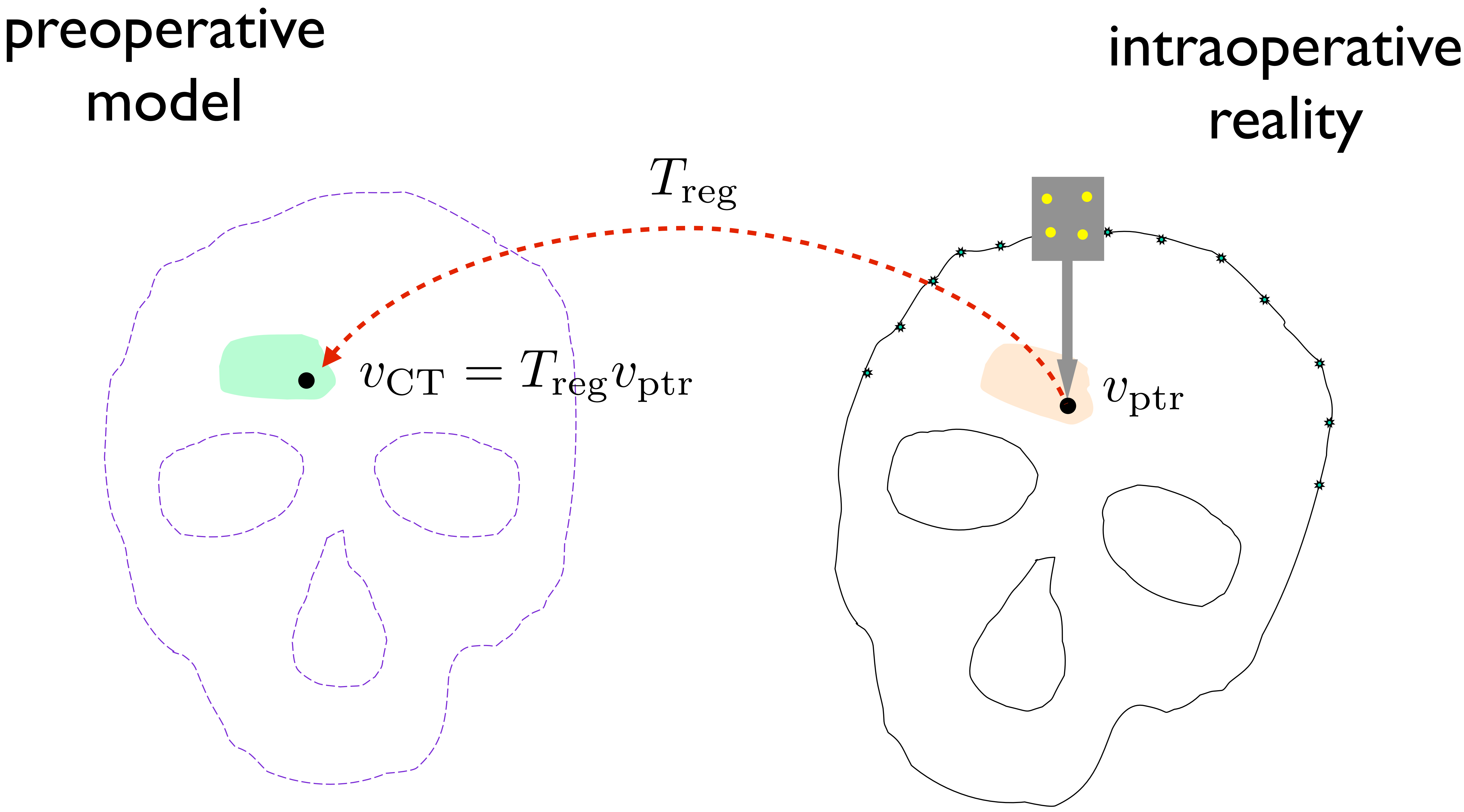
# a typical robot registration problem

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 pre-operative image

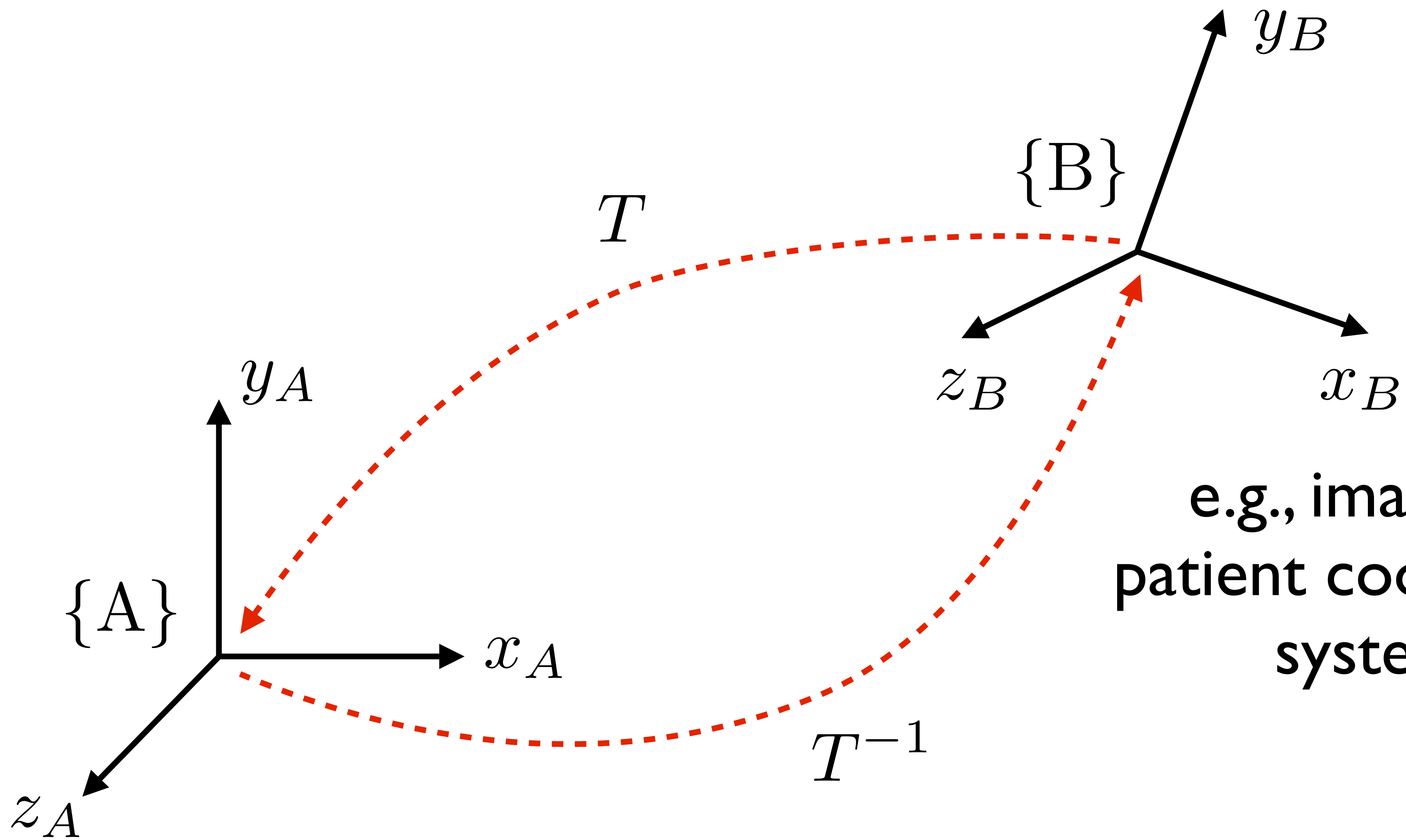
registration example and graphics provided by Russell H. Taylor, JHU

# a typical robot registration problem




What is $T_{\text{reg}}$?

# coordinate systems and transformations




e.g., image or patient coordinate system

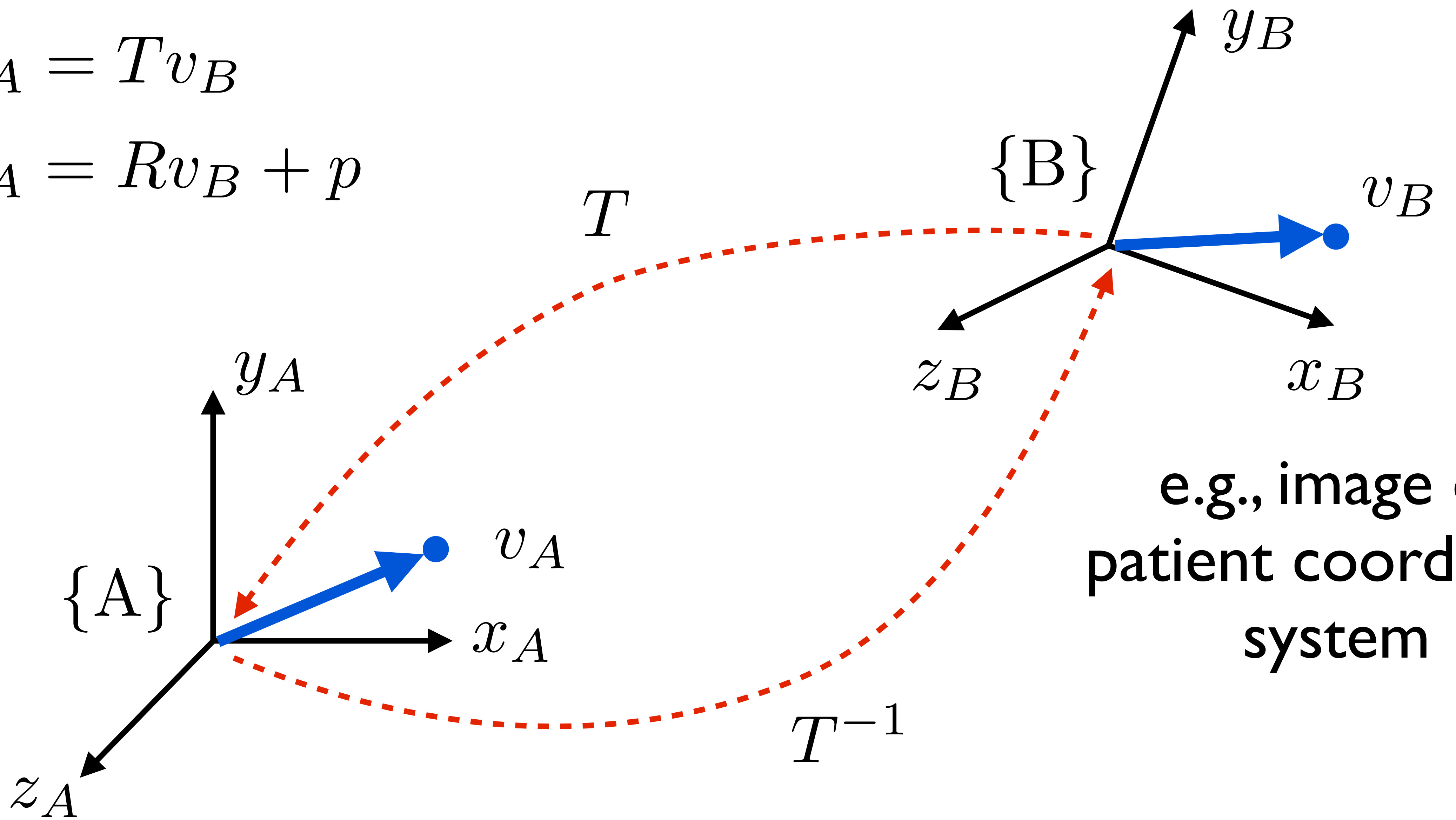
e.g., robot base coordinate system

# coordinate systems and transformations

$$v_{\mathrm{A}} = F(v_{\mathrm{B}})$$

$$v_A = T v_B$$

$$v_A = R v_B + p$$




e.g., image or patient coordinate system

e.g., robot base coordinate system

$$T(R,p) = \left[ \begin{array}{cccc} & R_{3\times 3} & & p_{3\times 1} \\ 0 & 0 & 0 & 1 \end{array} \right]$$

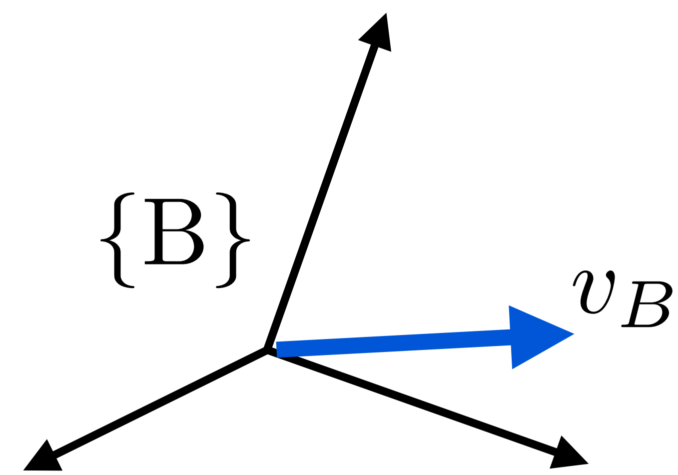
# coordinate systems and transformations

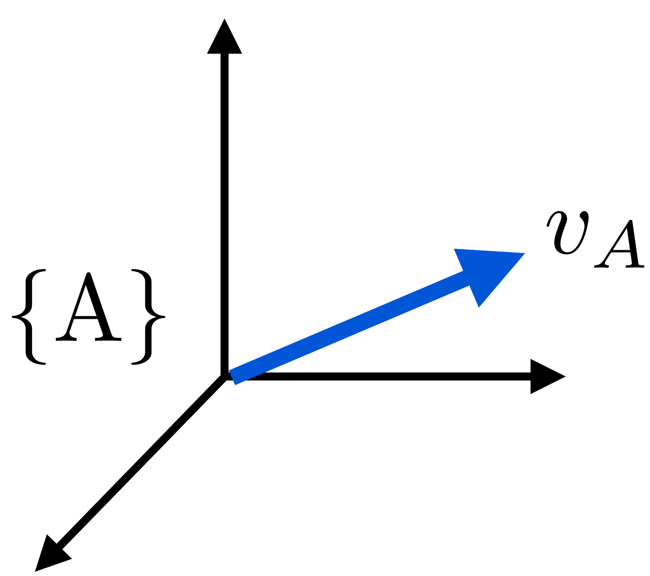
$$v_{\mathrm{A}} = F(v_{\mathrm{B}})$$

$$v_A = T v_B$$

$$v_A = R v_B + p$$

first rotate



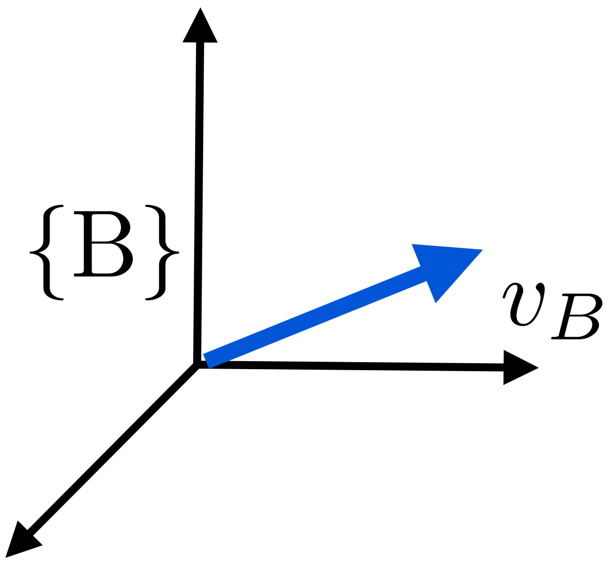

$$T(R,p) = \begin{bmatrix} & R_{3\times 3} & & p_{3\times 1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# coordinate systems and transformations

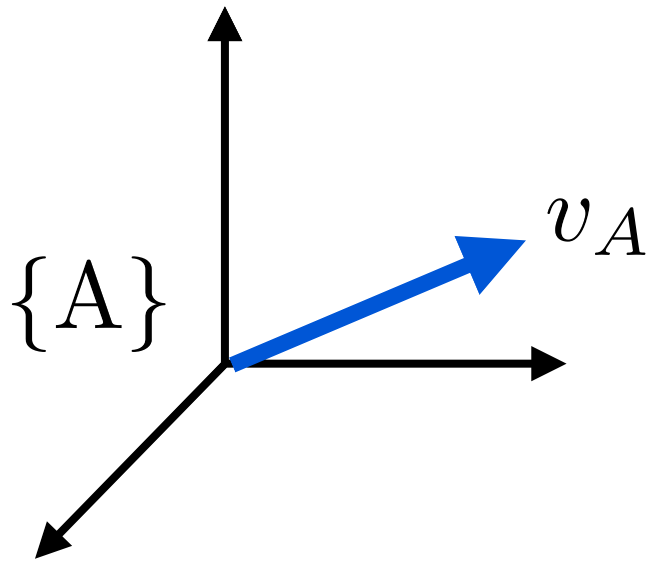
$$v_{\mathrm{A}} = F(v_{\mathrm{B}})$$

$$v_A = T v_B$$

$$v_A = R v_B + p$$




then translate




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

# 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

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

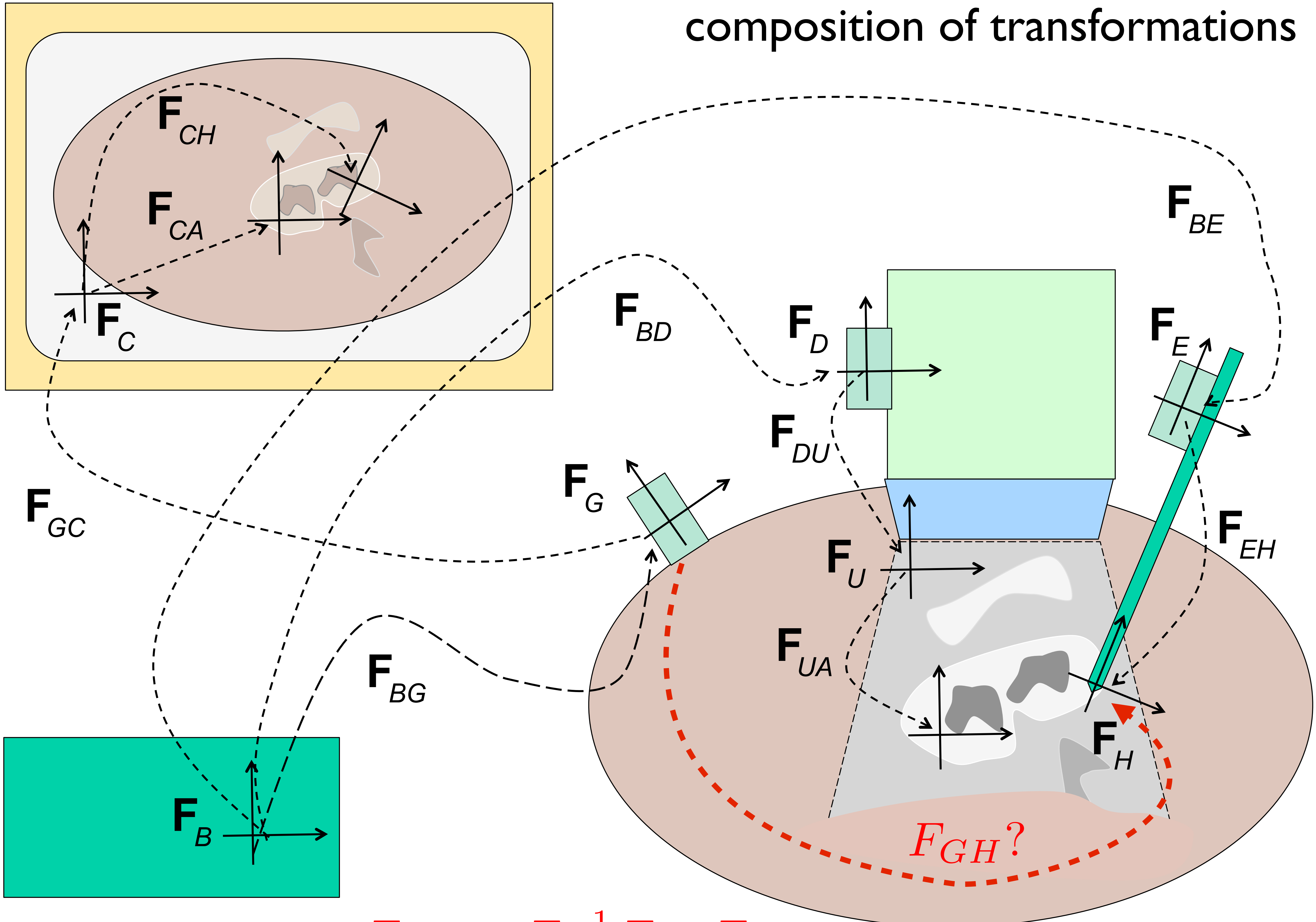
## rotation matrices are orthogonal matrices

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

and members of the group SO(3)

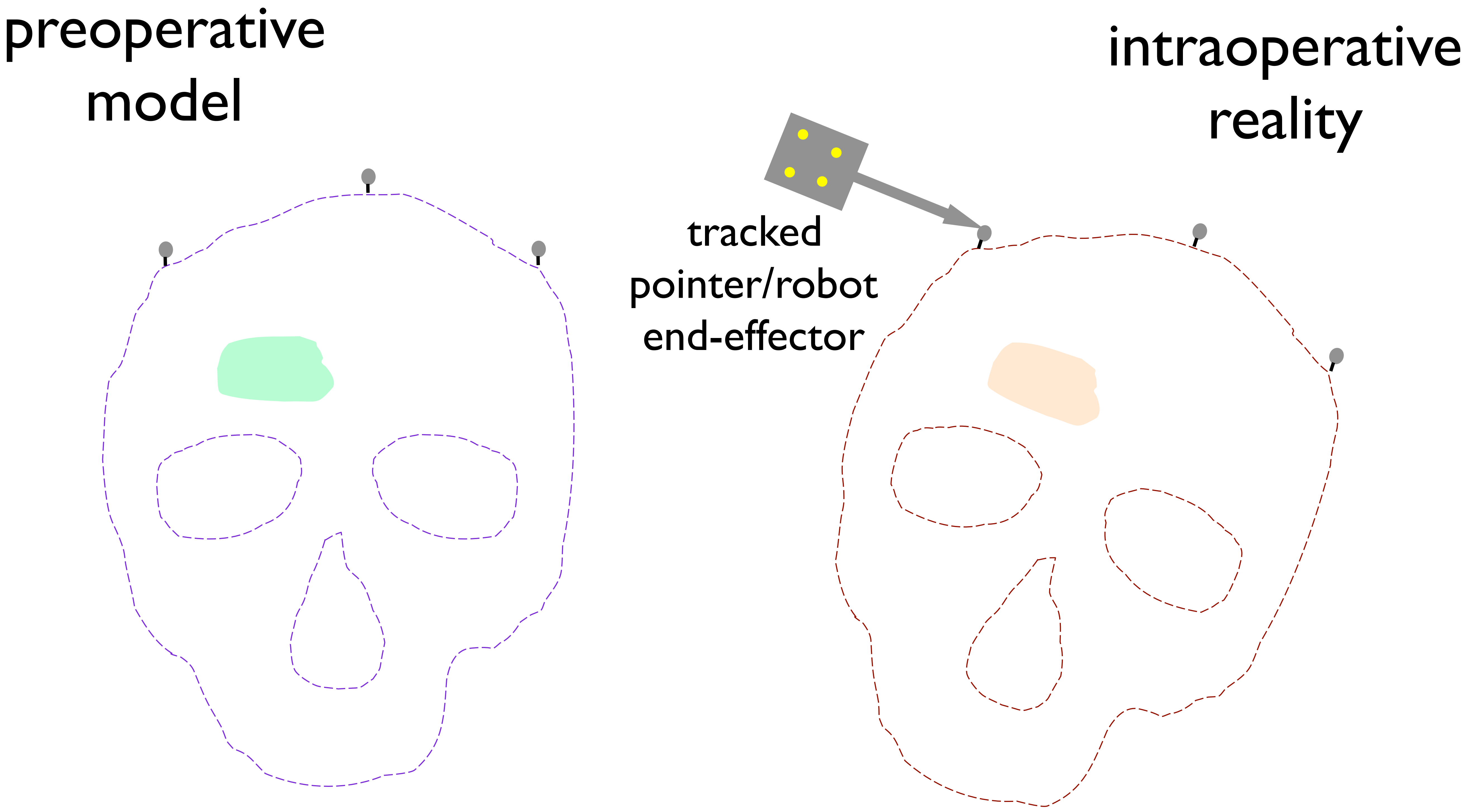
$$\det(R) = +1$$

# composition of transformations

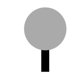
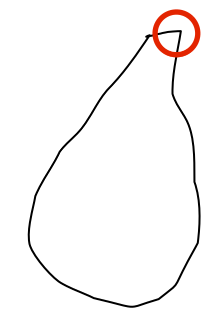
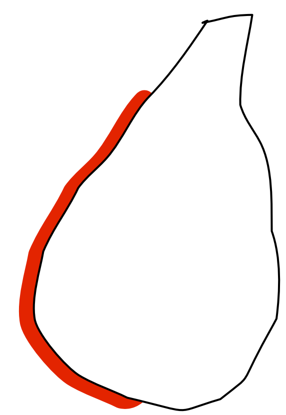



$$F_{GH} = F_{BG}^{-1} F_{BE} F_{EH}$$

# feature-based registration



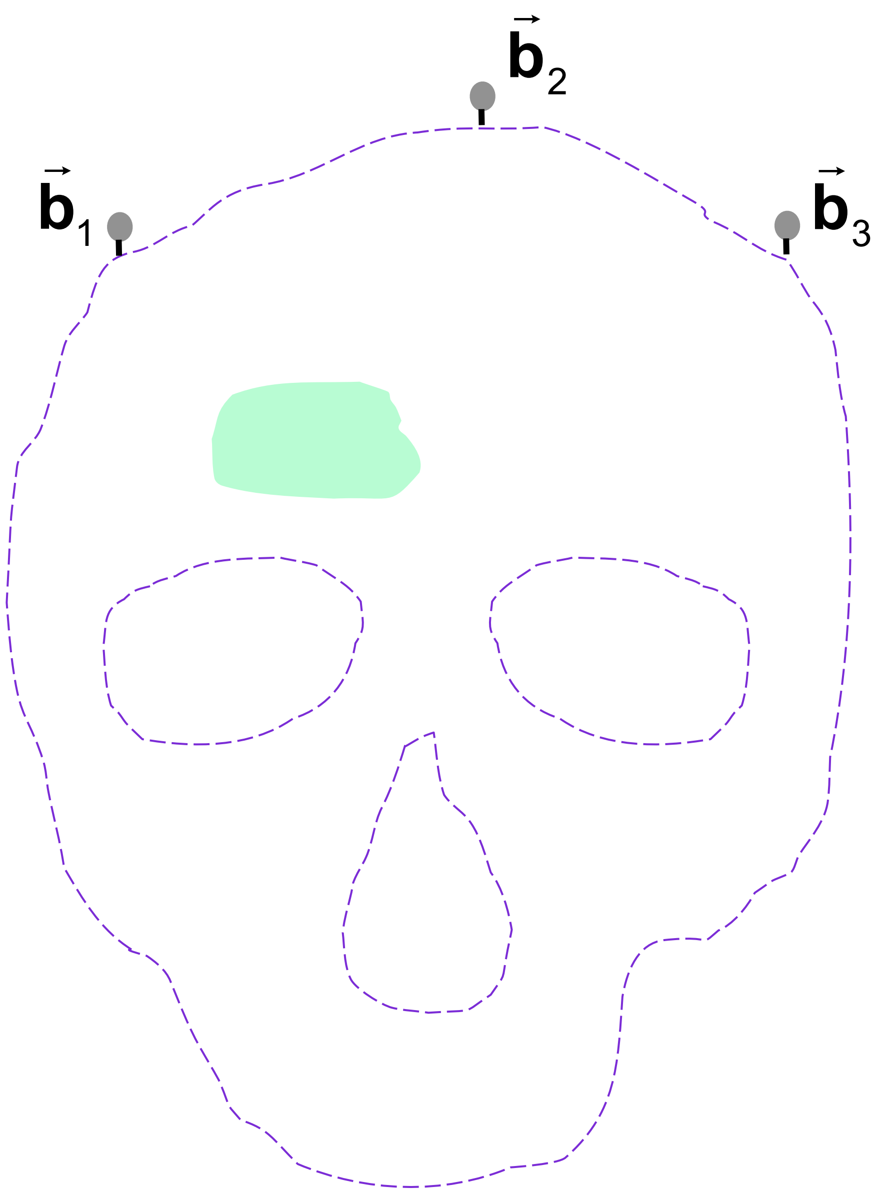
# features used for feature-based registration

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

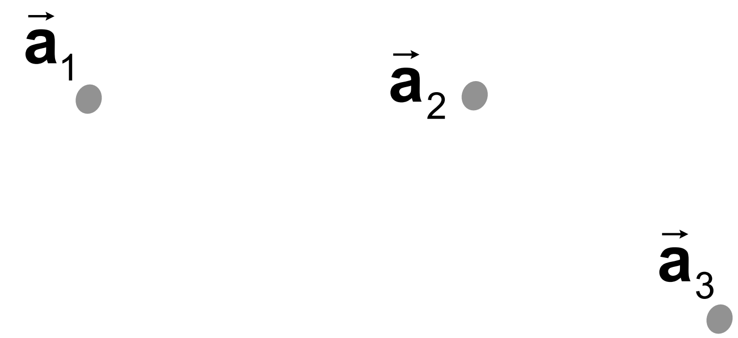
given two features (either the same features in two different reference frames, of two different features in the same reference frame), we need to know the "distance" between them

# what the computer knows

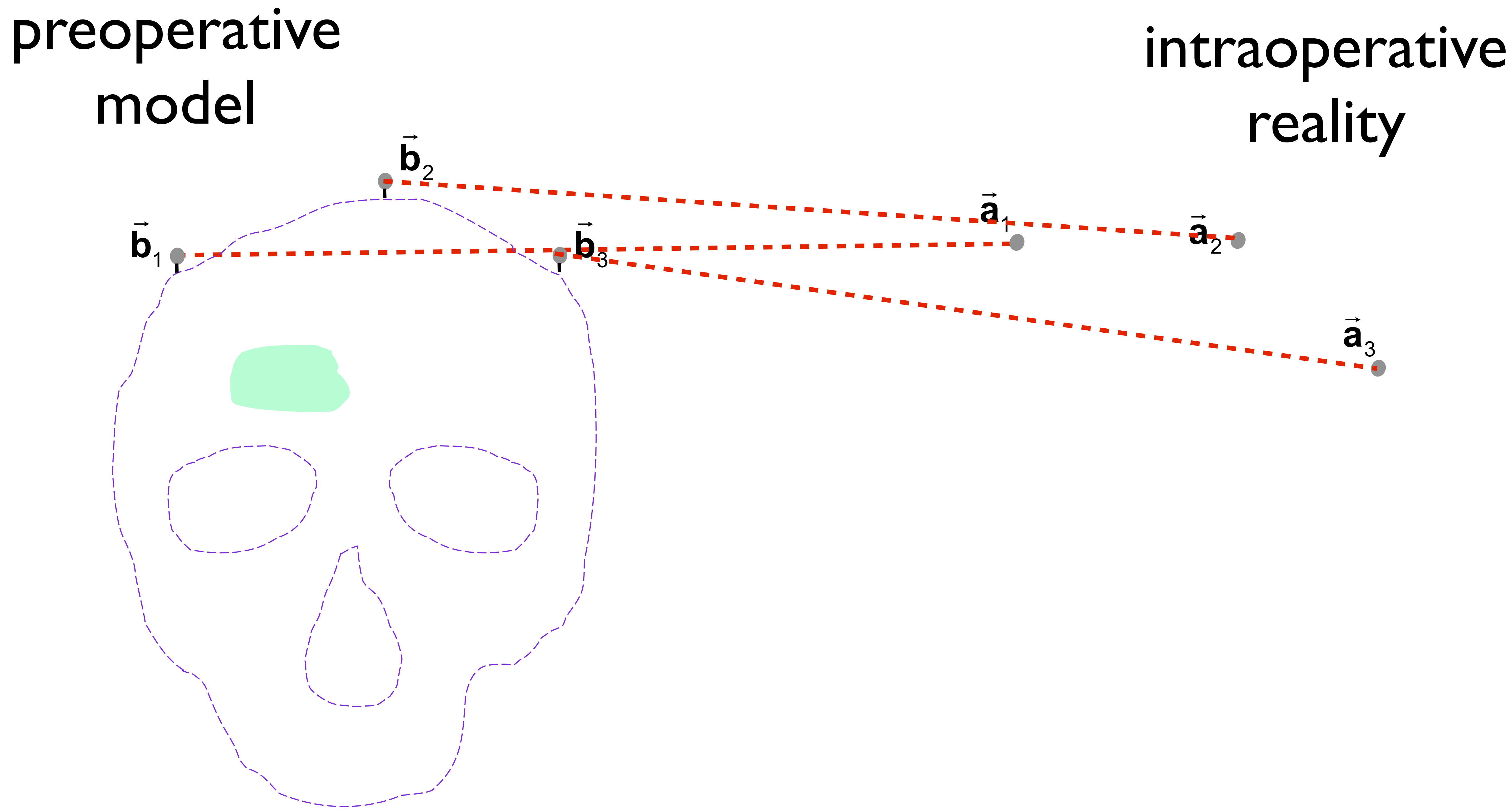
preoperative
model




intraoperative
reality



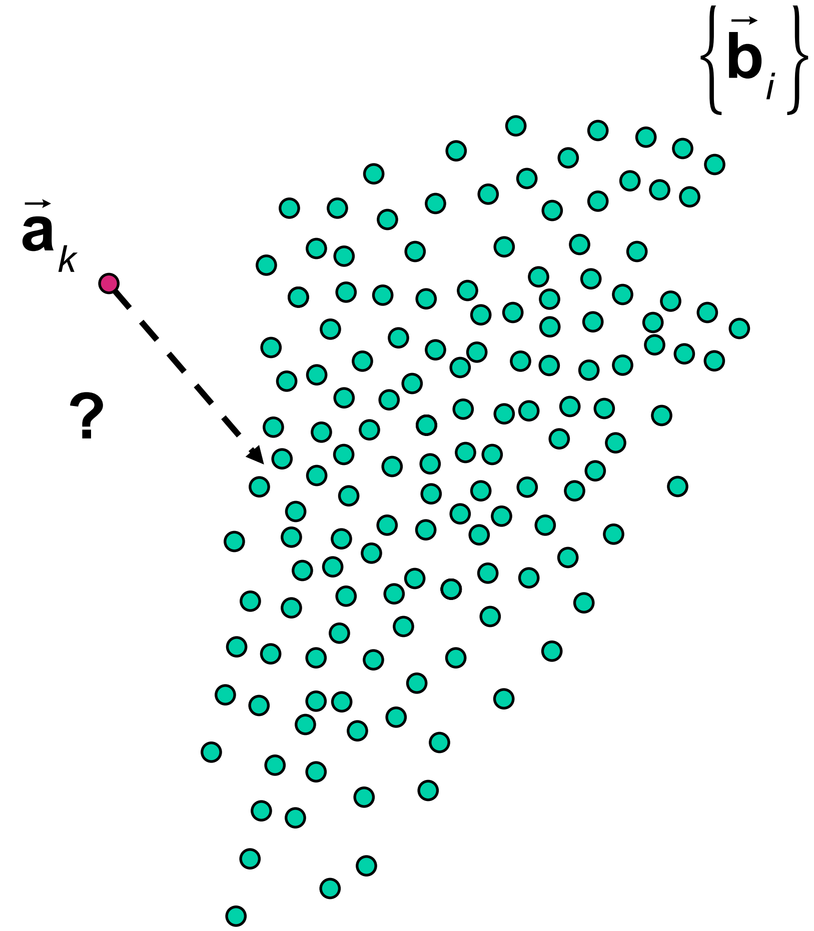
# identify corresponding points



# how to identify correspondences

manual
vs.
part of the optimization/
minimization process

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

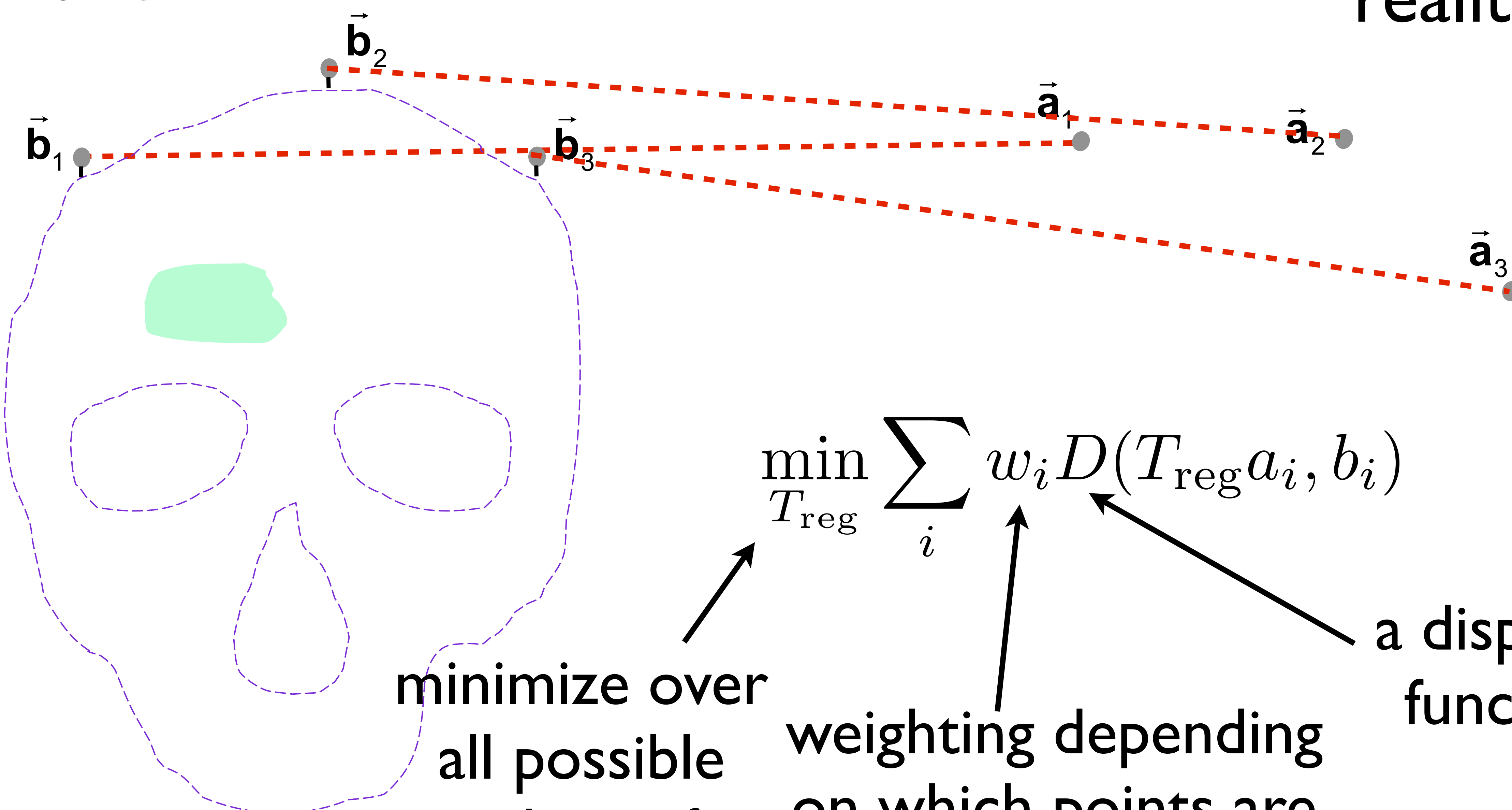



... but we will assume manual is possible

# find the best rigid transformation

preoperative model

intraoperative reality




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

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

weighting depending on which points are being considered

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

a disparity function

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

$$\bar{a} = \frac{1}{N}\sum_{i=1}^{N} a_i \qquad \bar{b} = \frac{1}{N}\sum_{i=1}^{N} b_i$$

$$\tilde{a}_i = a_i - \bar{a} \qquad \tilde{b}_i = b_i - \bar{b}$$

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

There are 4 substeps within this step

Step 3: Find $p$

$$p = \bar{b} - R\bar{a}$$

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

$$\mathbf{H} = \sum_i \begin{bmatrix} \tilde{a}_{i,x}\tilde{b}_{i,x} & \tilde{a}_{i,x}\tilde{b}_{i,y} & \tilde{a}_{i,x}\tilde{b}_{i,z} \\ \tilde{a}_{i,y}\tilde{b}_{i,x} & \tilde{a}_{i,y}\tilde{b}_{i,y} & \tilde{a}_{i,y}\tilde{b}_{i,z} \\ \tilde{a}_{i,z}\tilde{b}_{i,x} & \tilde{a}_{i,z}\tilde{b}_{i,y} & \tilde{a}_{i,z}\tilde{b}_{i,z} \end{bmatrix}$$

outer product between a and b

Step 2: Compute the SVD of $\mathbf{H} = \mathbf{USV}^{\mathbf{t}}$

Step 3: $\mathbf{R} = \mathbf{VU}^{\mathbf{t}}$

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

K. S. Arun, T. S. Huang, S. D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(5): 698-700, 1987.

# Extra Slides: An Iterative Method for Finding the Rotation Matrix

(I don't recommend using this for your assignment, 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

$$a = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T \qquad \text{skew}(a) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

$$a \times v = \text{skew}(a)v$$

## representations/constructions of rotation matrices

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

Rodrigues' formula: $c = \text{rot}(a, \alpha)b = b\cos(\alpha) + a \times b\sin(\alpha) + a(a^T b)(1 - \cos(\alpha))$

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

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

more notation: $\text{rot}(a, \alpha) = R_a(\alpha)$ $\qquad R(a) = \text{rot}(a, ||a||)$

# “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 $a$

$\mathrm{rot}(a, ||a||) b \approx a \times b + b$ 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 $\Delta R v \approx v + \bar{\alpha} \times v$

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