



NSF Engineering Research Center for  
Computer Integrated Surgical Systems and  
Technology



LABORATORY FOR  
**Computational  
Sensing + Robotics**  
THE JOHNS HOPKINS UNIVERSITY

## Registration – Part 1

**600.455/655 Computer Integrated Surgery**



100 YEARS  
JOHNS HOPKINS ENGINEERING

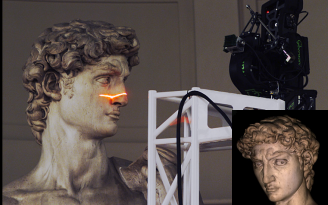
**WHITING  
SCHOOL OF  
ENGINEERING**  
THE JOHNS HOPKINS UNIVERSITY

**Russell H. Taylor**  
John C. Malone Professor of Computer Science,  
with joint appointments in Mechanical Engineering, Radiology & Surgery  
Director, Laboratory for Computational Sensing and Robotics  
The Johns Hopkins University  
rht@jhu.edu




1

## Why is Registration Important?



Digitize important cultural artifacts




Medical interventions



Archeology

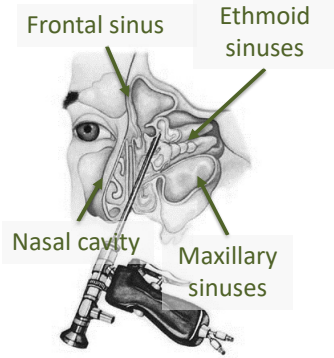
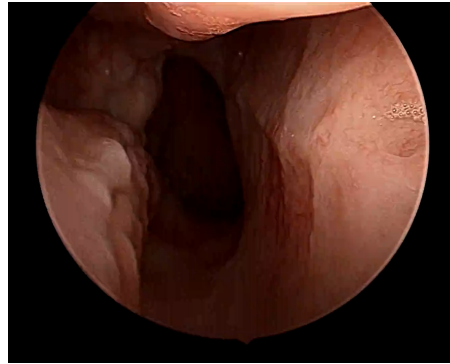
And many more applications...

Copyright 2021 R. H. Taylor      Slide Credit: **Ayushi Sinha**  
Engineering Research Center for Computer Integrated Surgical Systems and Technology



2

### Why is Registration Important?



Typical Example: Sinus Endoscopy. The surgeon can only see video from the endoscope. But crucial data is in the CT about structures that cannot be seen.

Slide Credit: Ayushi Sinha

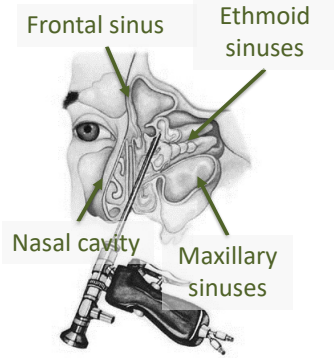
Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



3

### Why is Registration Important?



Typical Example: Sinus Endoscopy. After registration, the computer can create video overlays, help guide a robot, or provide other assistance.

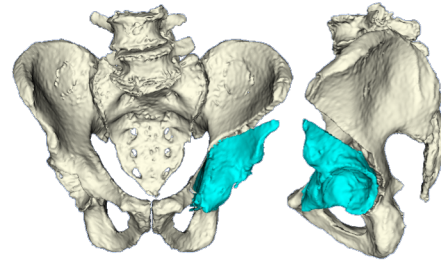
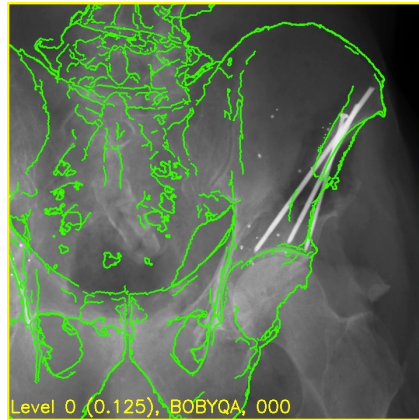
Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



4

## Why is Registration Important?



Typical Example: Osteotomies. Surgeon needs to know the position and orientation of bone fragment relative to pelvis, based on x-ray images.

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



5

## What needs registering?

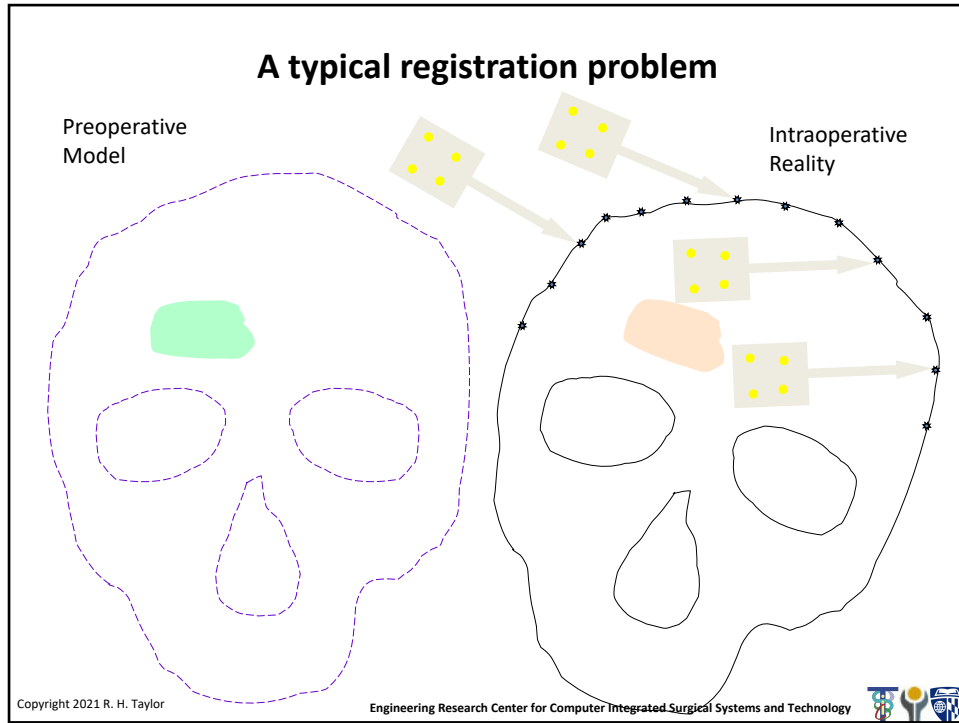
- **Preoperative Data**
  - 2D & 3D medical images
  - Models
  - Preoperative positions
- **Intraoperative Data**
  - 2D & 3D medical images
  - Models
  - Intraoperative positioning information
- **The Patient**

Copyright 2021 R. H. Taylor

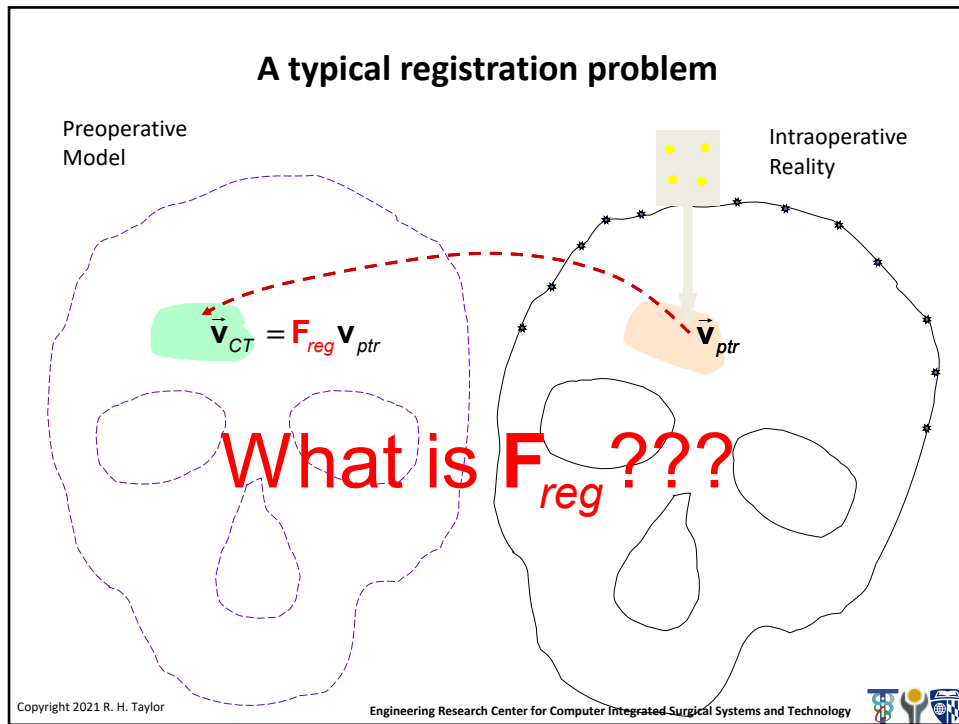
Engineering Research Center for Computer Integrated Surgical Systems and Technology



6



7



8

## Taxonomy of methods

- Feature-based
- Intensity-based

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



9

## Framework for feature-based methods

- Definition of coordinate system relations
- Segmentation of reference features
- Definition of disparity function between features
- Optimization of disparity function

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



10

## Definitions

**Overall Goal:** Given two coordinate systems,

$\text{Ref}_A$  &  $\text{Ref}_B$

and coordinates

$x_A$  &  $x_B$

associated with corresponding features in the two coordinate systems, the general goal is to determine a transformation function  $T$  that transforms one set of coordinates into the other:

$$x_A = T(x_B)$$

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



11

## Definitions

- **Rigid Transformation:** Essentially, our old friends 2D & 3D coordinate transformations:

$$T(x) = R \cdot x + p$$

The key assumption is that deformations may be neglected.

- **Similarity Transformation:** Essentially, rigid+scale change. Preserves angles and shape, but not size

$$T(x) = sR \cdot x + p$$

- **Elastic Transformation:** Cases where must take more general deformations into account. Many different flavors, depending on what is being deformed

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



12

## Uses of Rigid Transformations

- Register (approximately) multiple image data sets
- Transfer coordinates from preoperative data to reality (especially in orthopaedics & neurosurgery)
- Initialize non-rigid transformations

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



13

## Uses of Elastic Transformations

- Register different patients to common data base (e.g., for statistical analysis)
- Overlay atlas information onto patient data
- Study time-varying deformations
- Assist segmentation

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



14

## Typical Features

- Point fiducials
- Point anatomical landmarks
- Ridge curves
- Contours
- Surfaces
- Line fiducials

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



15

## Distance Functions

Given two (possibly distributed) features  $F_i$  and  $F_j$ , need to define a distance metric distance  $(F_i, F_j)$  between them. Some choices include:

- Minimum distance between points
- Maximum of minimum distances
- Area between line features
- Volume between surface features
- Area between point and line
- etc.

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



16



## Distance Functions Between Feature Sets

Let  $\mathcal{F}_A = \{\dots F_{Ai} \dots\}$  and  $\mathcal{F}_B = \{\dots F_{Bi} \dots\}$  be corresponding sets of features in  $\mathbf{Ref}_A$  and  $\mathbf{Ref}_B$ , respectively. We need to define an appropriate disparity function  $D(\mathcal{F}_A, \mathcal{F}_B)$  between feature sets. Some typical choices include:

$$D = \sum_i w_i [\text{distance}(F_{Ai}, \mathbf{T}(F_{Bi}))]^2$$

$$D = \max_i \text{distance}(F_{Ai}, \mathbf{T}(F_{Bi}))$$

$$D = \text{median}_i \text{distance}(F_{Ai}, \mathbf{T}(F_{Bi}))$$

$$D = \text{Cardinality}\{i | \text{distance}(F_{Ai}, \mathbf{T}(F_{Bi})) > \text{threshold}\}$$

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



17

## Optimization

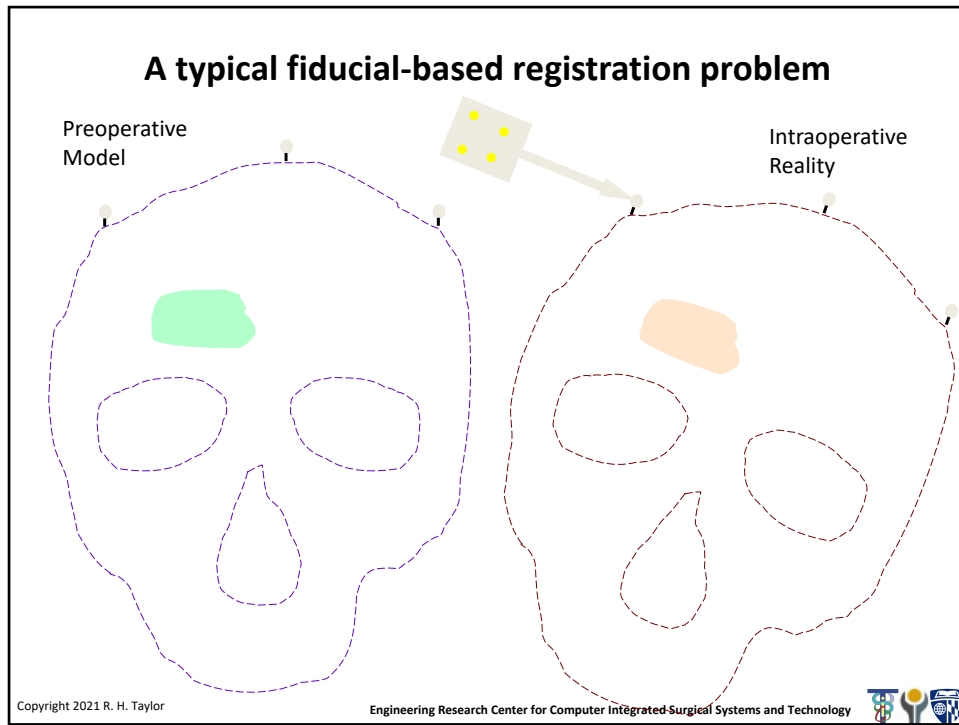
- Global vs Local
- Numerical vs Direct Solution
- Local Minima

Copyright 2021 R. H. Taylor

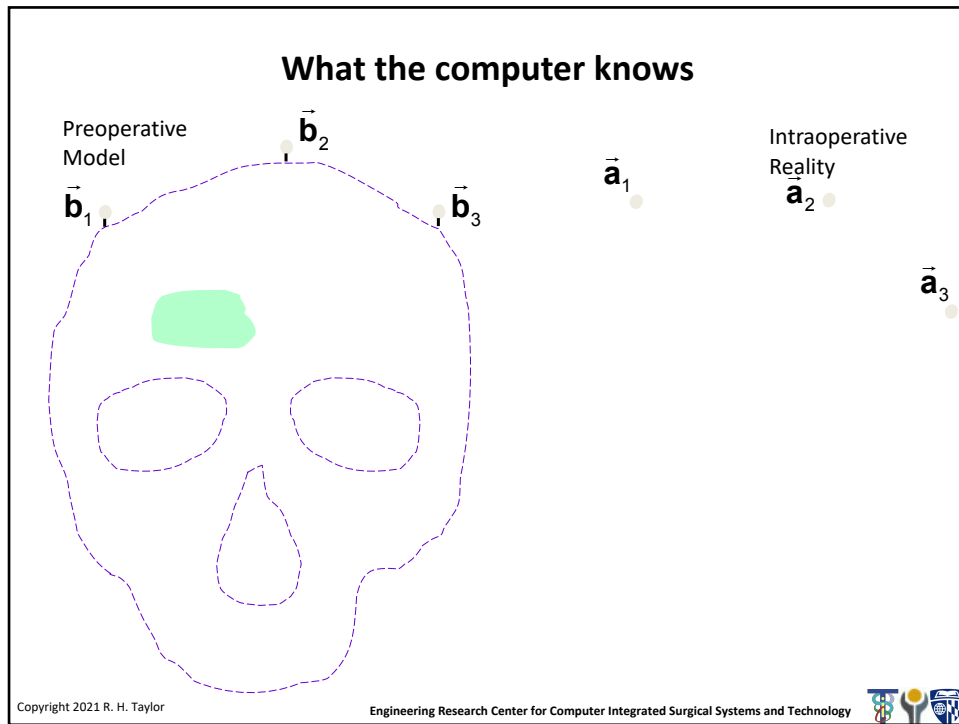
Engineering Research Center for Computer Integrated Surgical Systems and Technology



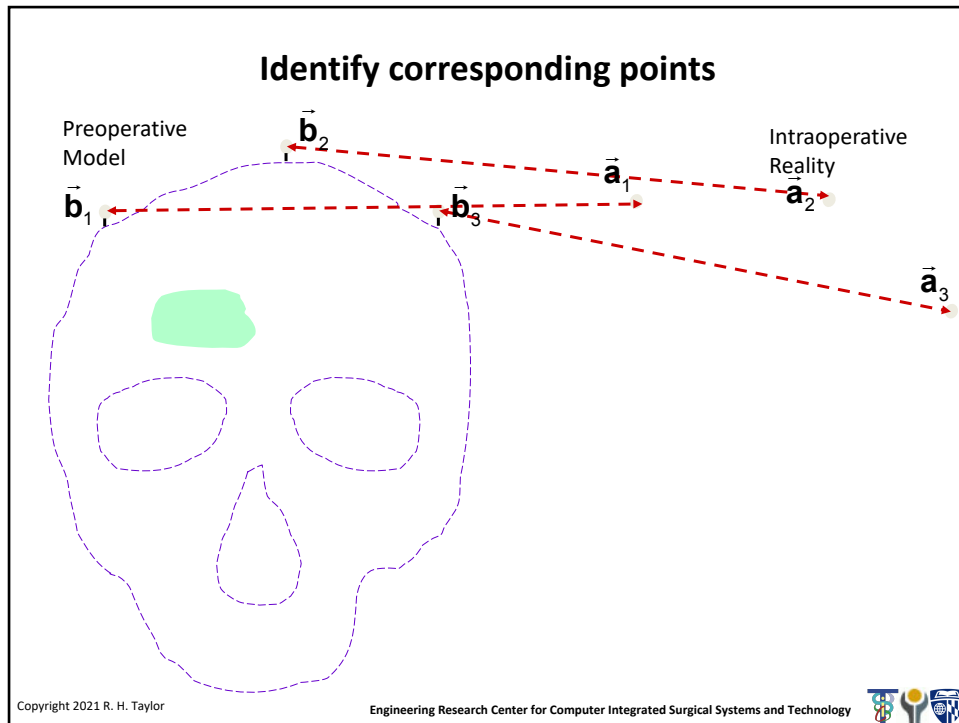
18



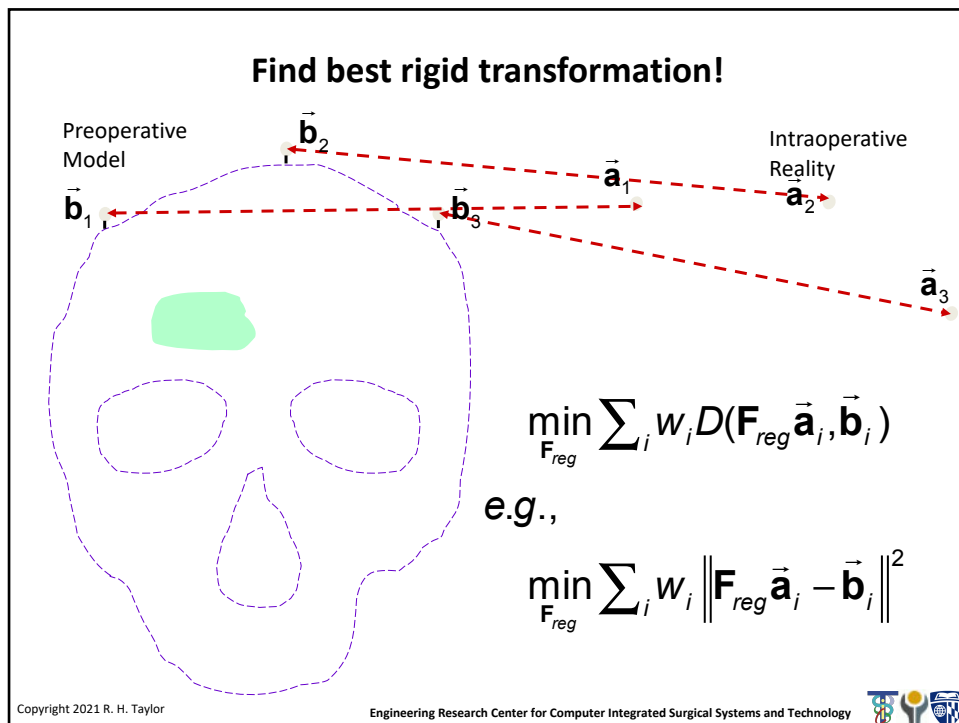
19



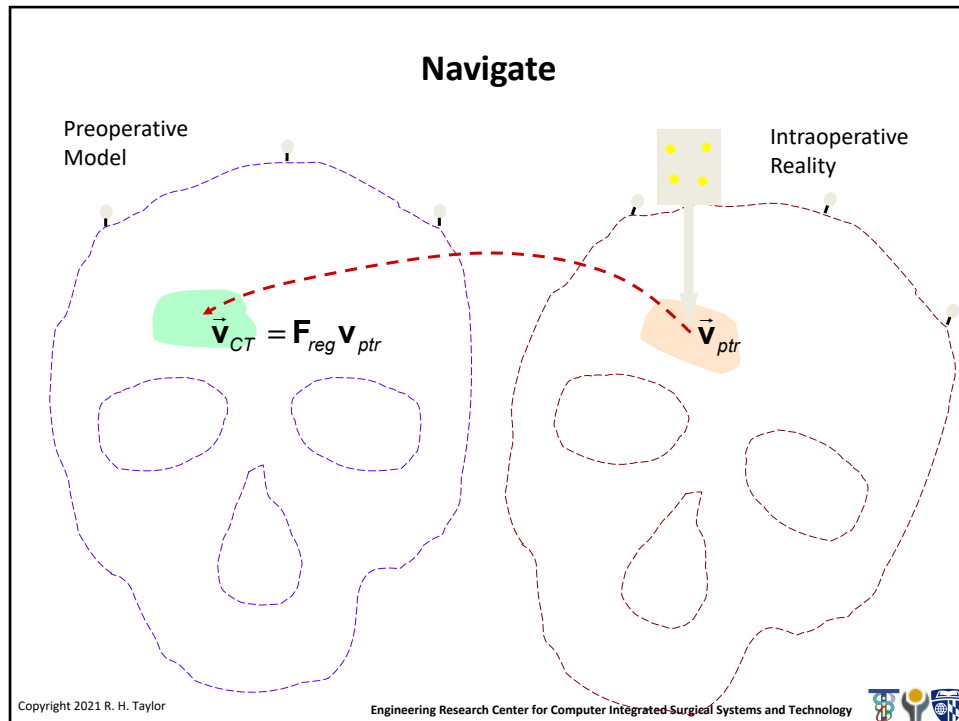
20



21



22



23

### Sampled 3D data to surface models

#### Outline:

- Select large number of sample points
- Determine distance function  $d_S(\mathbf{f}, \mathcal{F})$  for a point  $\mathbf{f}$  to a surface feature  $\mathcal{F}$ .
- Use  $d_S$  to develop disparity function  $D$ .

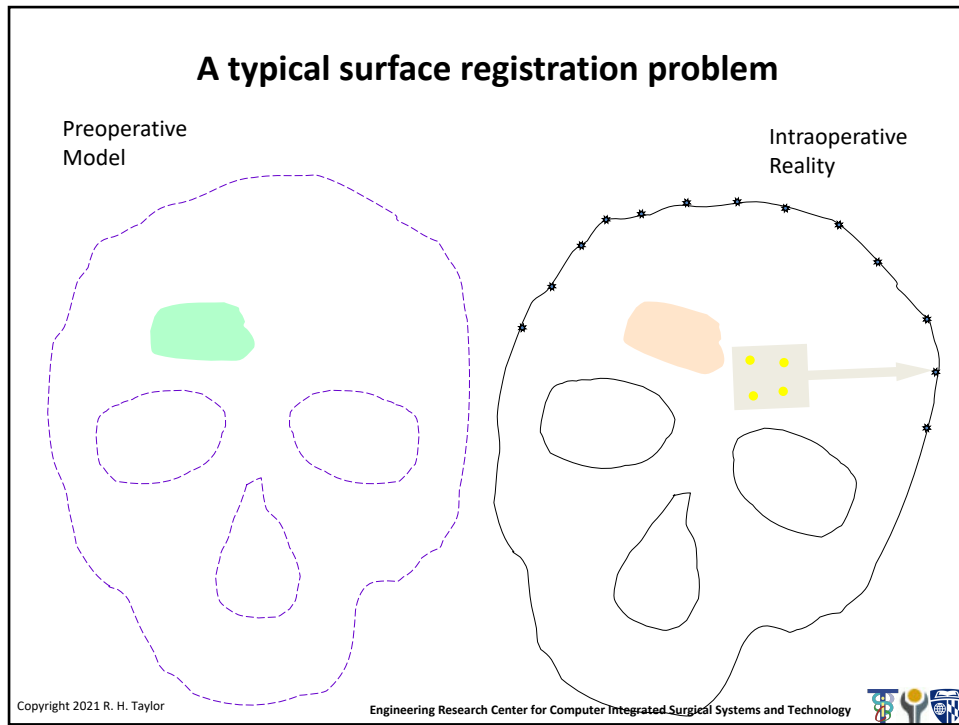
#### Examples

- Head-in-hat algorithm [Levin et al., 1988; Pelizzari et al., 1989]
- Distance maps [e.g., Lavallee et al.]
- Iterative closest point [Besl and McKay, 1992]

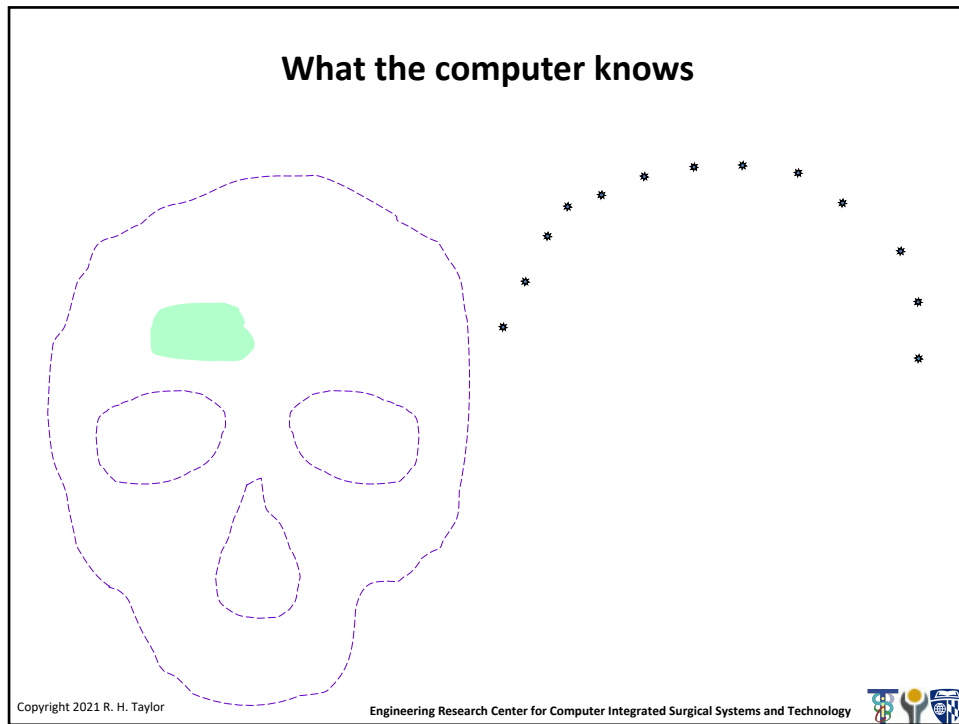
Copyright

Engineering Research Center for Computer Integrated Surgical Systems and Technology

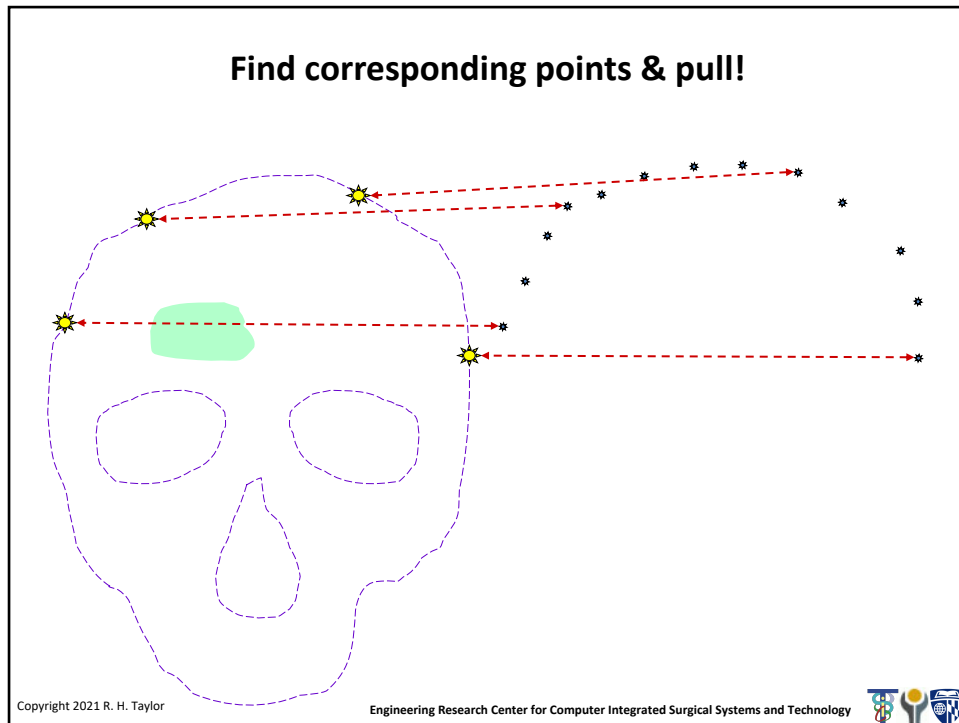
24



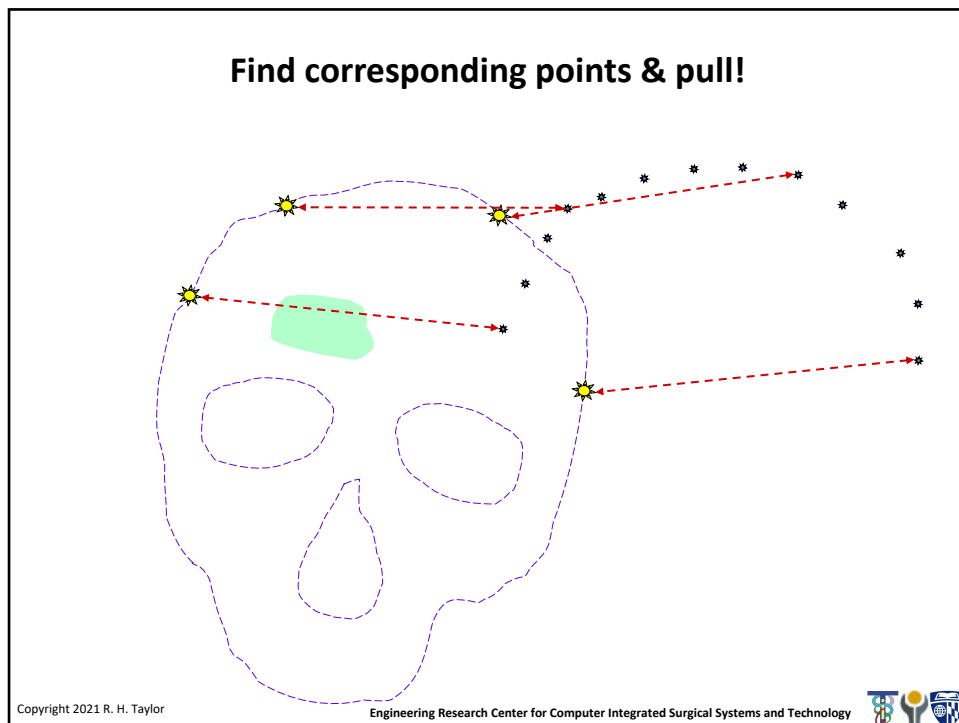
25



26



27



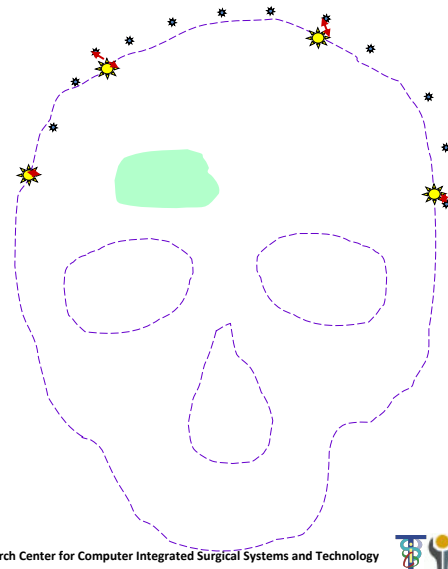
28

## Find corresponding points & pull!

Iterate this until converge

Find new point pairs every iteration

Key challenge is finding point pairs efficiently.



29

## Iterative Closest Point

- Besl and McKay, 1992
- Start with an initial guess,  $\mathbf{T}_0$ , for  $\mathbf{T}$ .
- At iteration  $k$ 
  1. For each sampled point  $\mathbf{f}_i \in \mathcal{F}_A$ , find the point  $\mathbf{v}_i \in \mathcal{F}_B$  that is closest to  $\mathbf{T}_k \cdot \mathbf{f}_i$ .
  2. Then compute  $\mathbf{T}_{k+1}$  as the transformation that minimizes

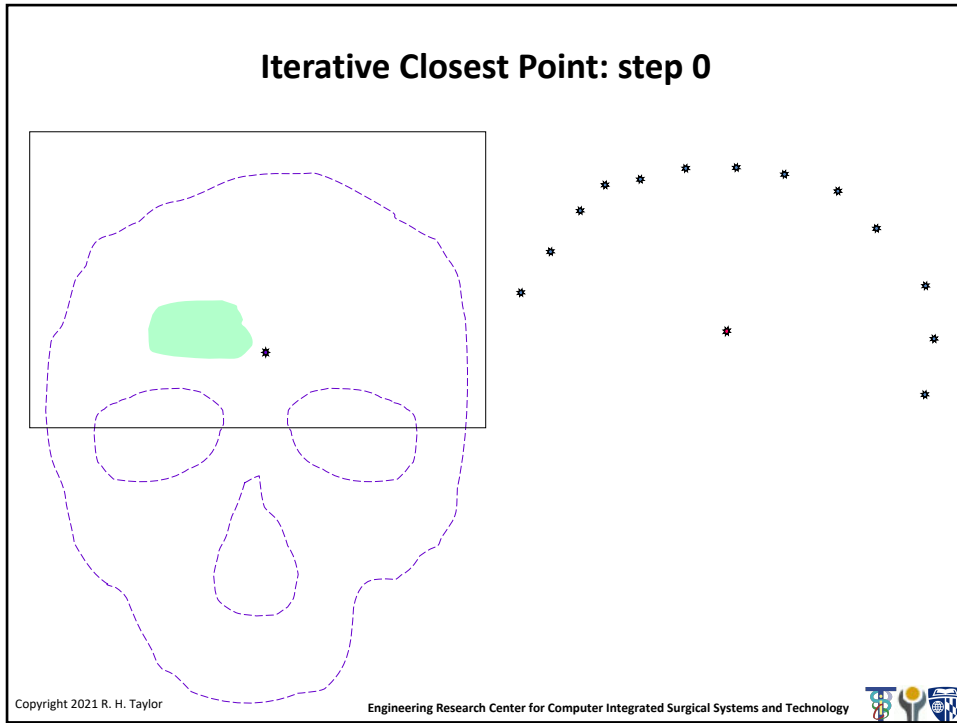
$$D_{k+1} = \sum_i \|\mathbf{v}_i - \mathbf{T}_{k+1} \cdot \mathbf{f}_i\|^2$$

- Physical Analogy

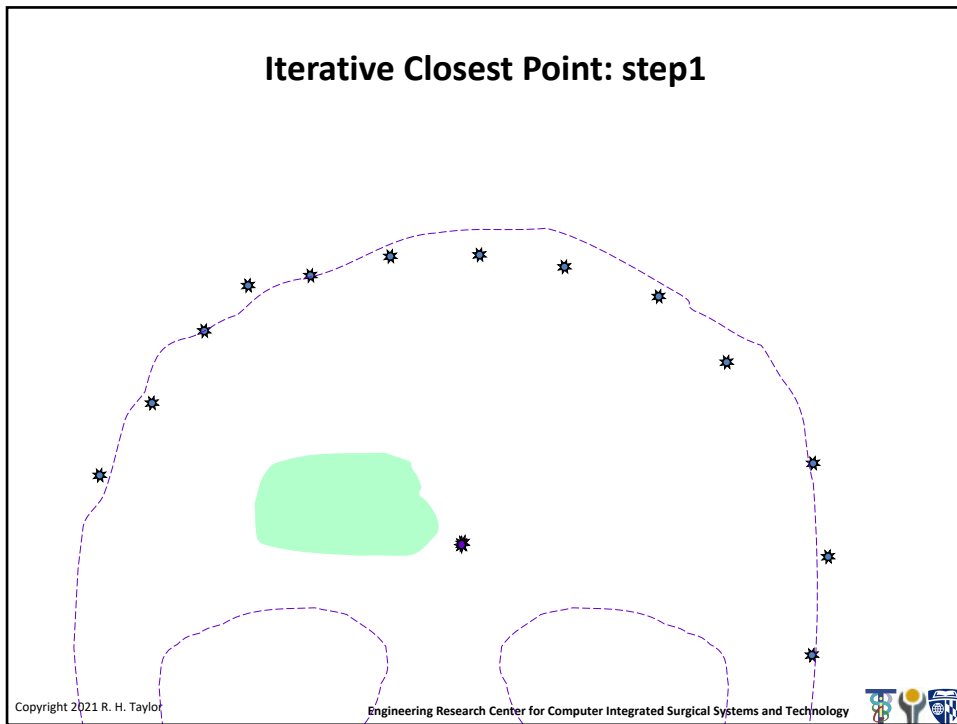
Cop



50

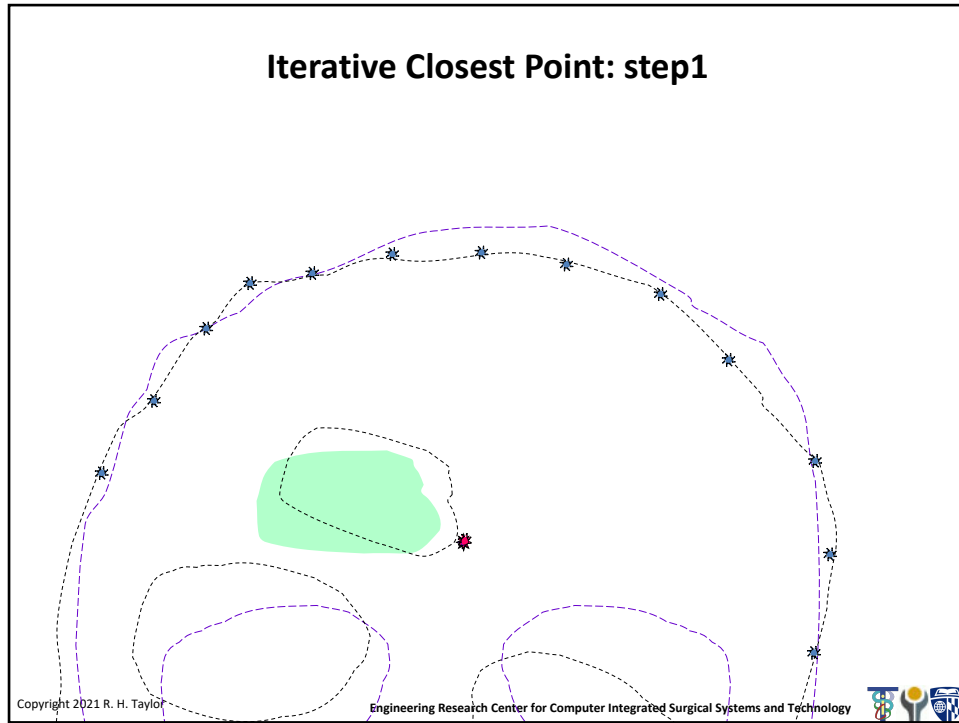


51

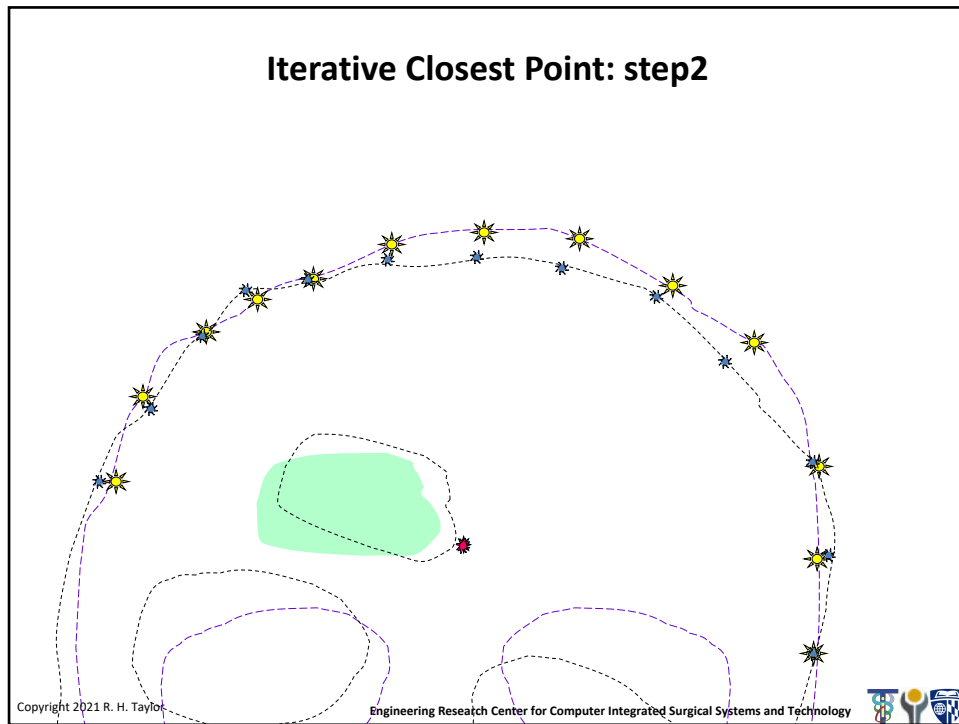


52

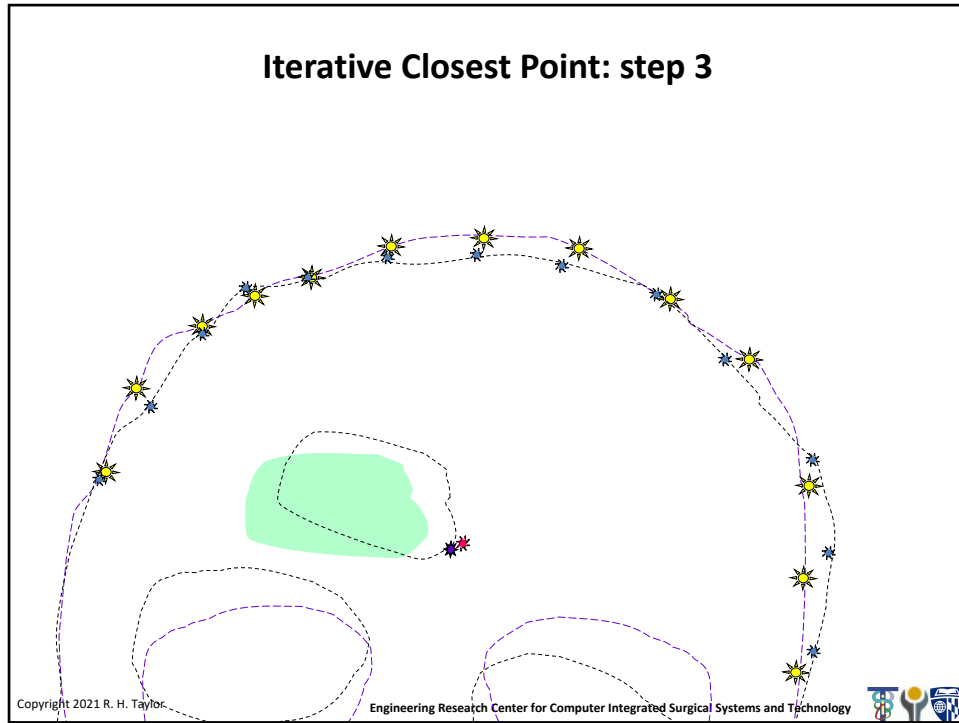




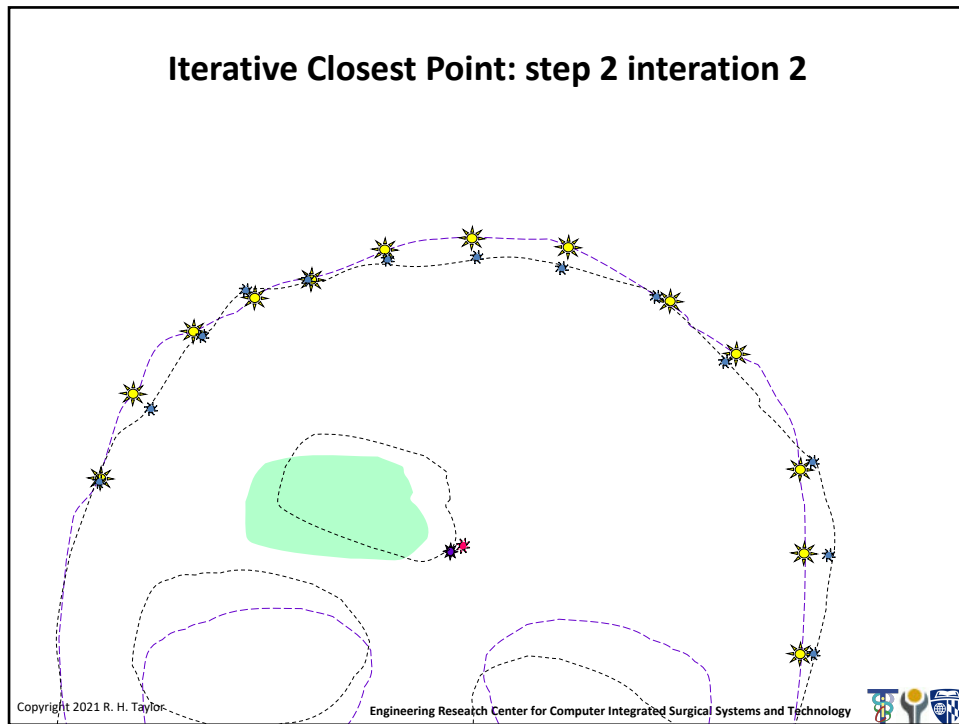
53



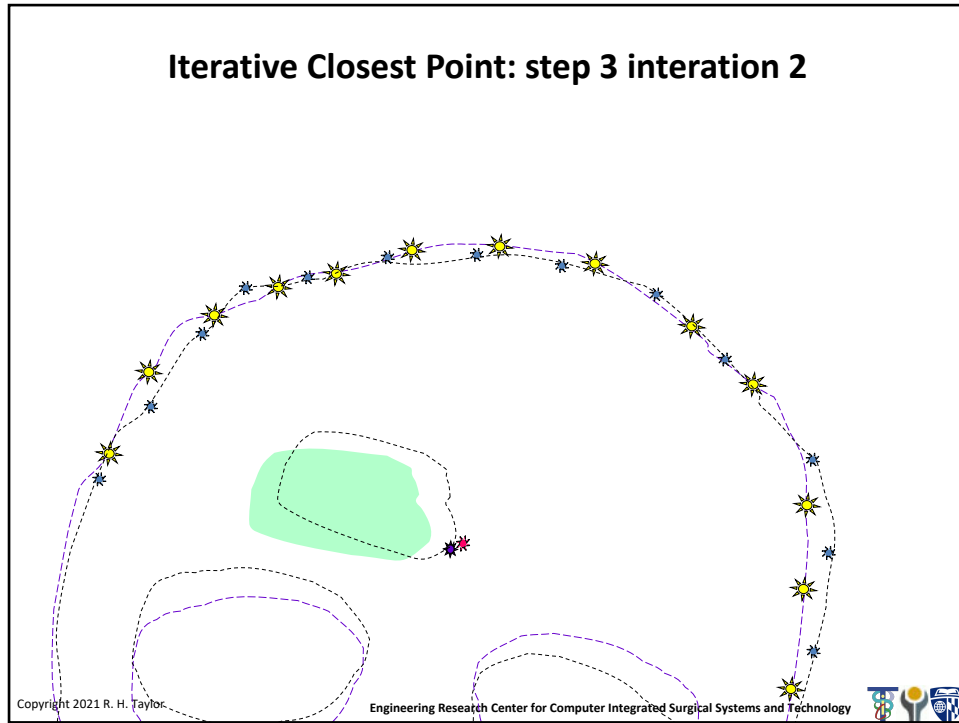
54



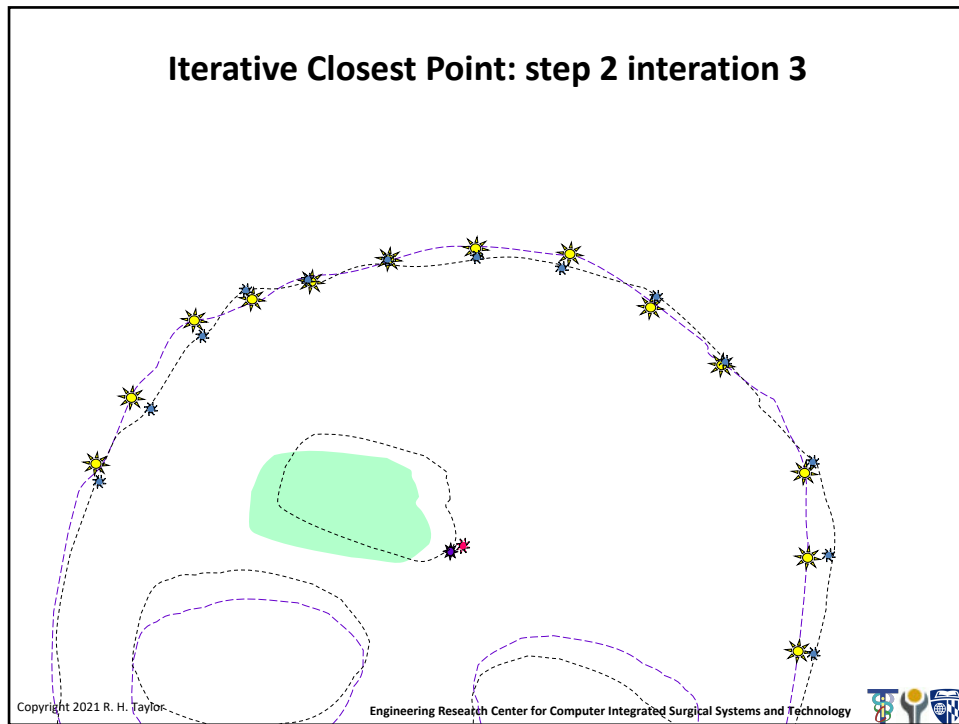
55



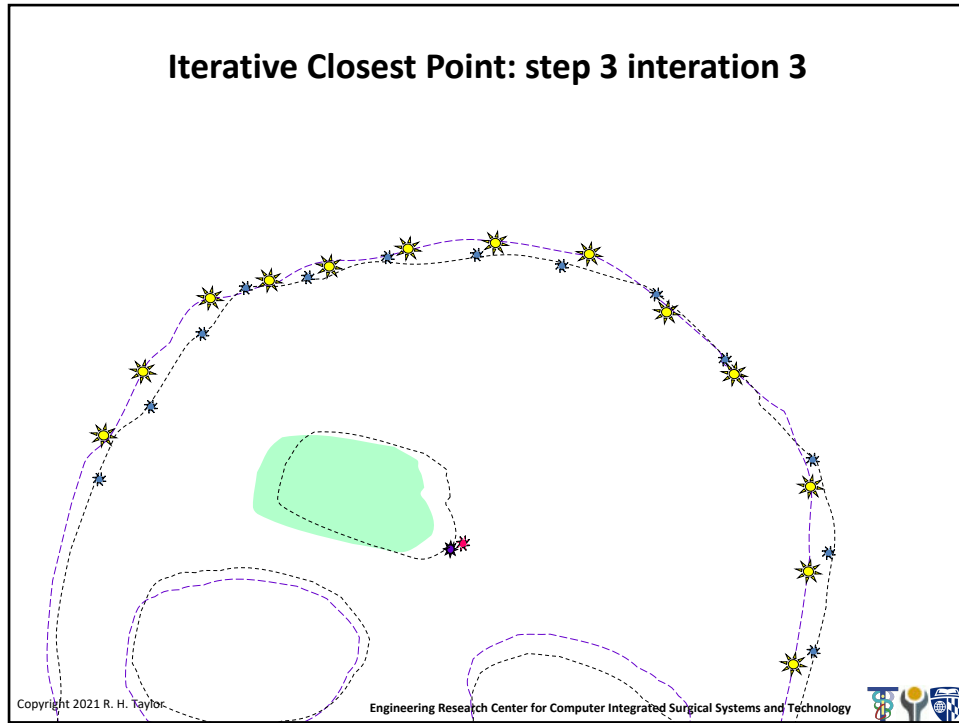
56



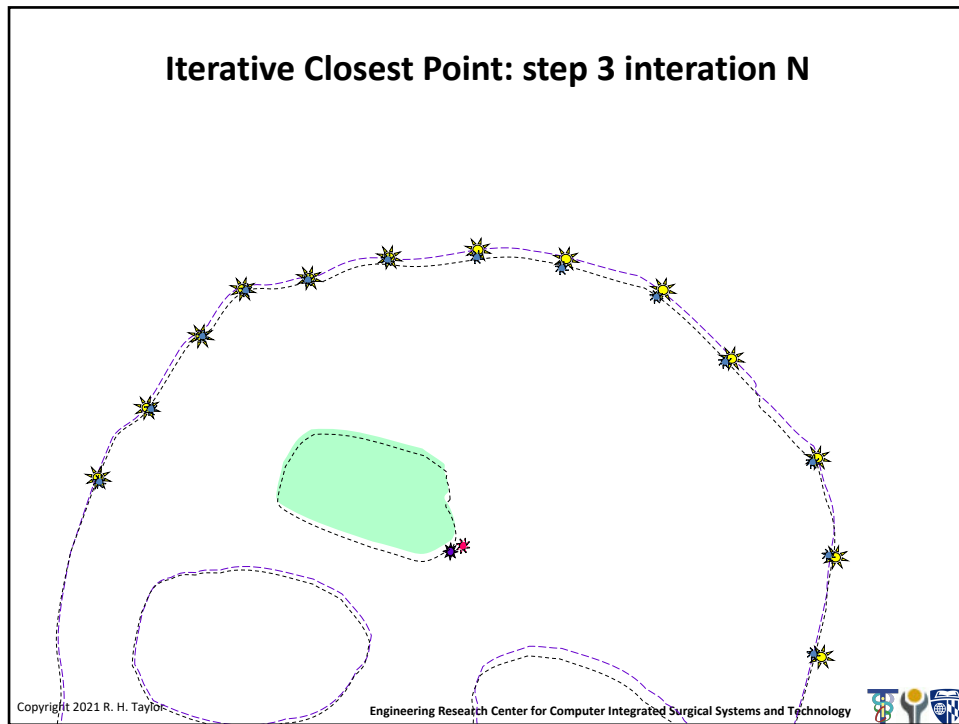
57



58



59



60

## Iterative Closest Point: Discussion

- Minimization step can be fast
- Crucially requires fast finding of nearest points
- Local minima still an issue
- Data overlap still an issue

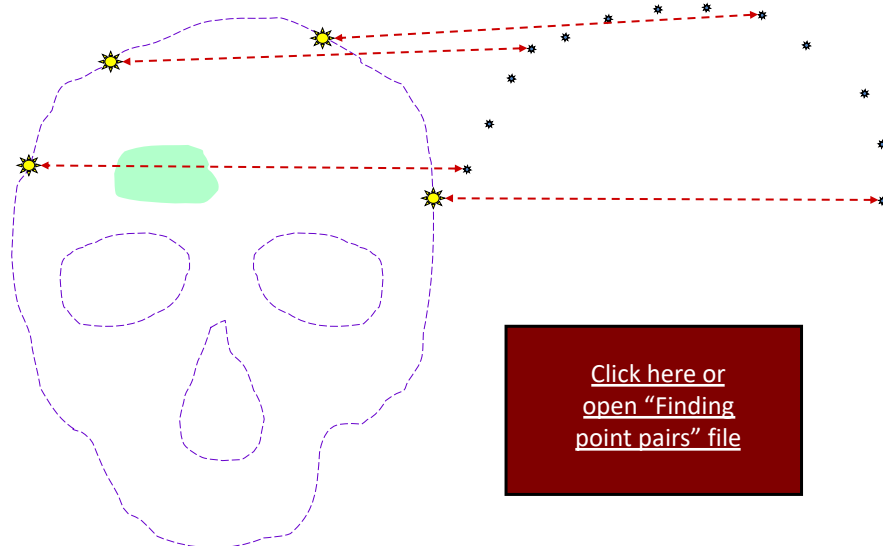
Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



61

## Digression: Finding Point Pairs



[Click here or  
open "Finding  
point pairs" file](#)

Copyright 2021 R. H. Taylor

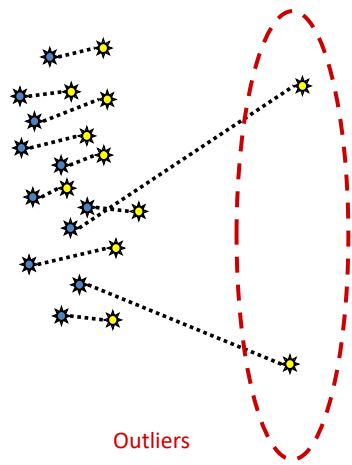
Engineering Research Center for Computer Integrated Surgical Systems and Technology



62

### Robust Pose Estimation ...

- Basic idea is to identify outliers and give them little or no weight.



Outliers

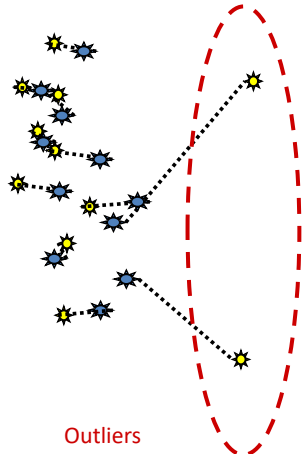
R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," *Comput. Vision, Graphics, Image Processing-IU*, vol. 60, no. 3, pp. 313–342, 1994.

Copyright 2021 R. H. Taylor Engineering Research Center for Computer Integrated Surgical Systems and Technology

63

### Robust Pose Estimation ...

- Basic idea is to identify outliers and give them little or no weight.



Outliers

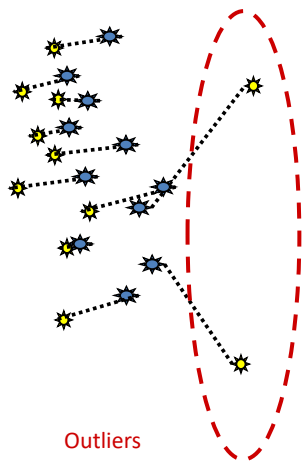
R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," *Comput. Vision, Graphics, Image Processing-IU*, vol. 60, no. 3, pp. 313–342, 1994.

Copyright 2021 R. H. Taylor Engineering Research Center for Computer Integrated Surgical Systems and Technology

64

### Robust Pose Estimation ...

- Basic idea is to identify outliers and give them little or no weight.



Outliers

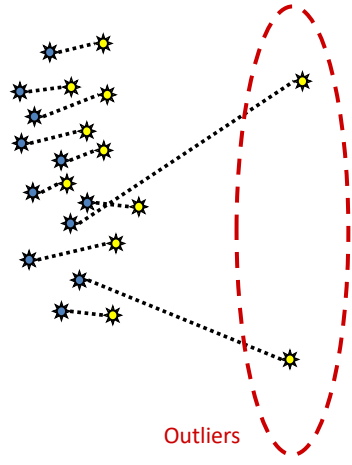
R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," *Comput. Vision, Graphics, Image Processing-IU*, vol. 60, no. 3, pp. 313–342, 1994.

Copyright 2021 R. H. Taylor Engineering Research Center for Computer Integrated Surgical Systems and Technology

65

### Robust Pose Estimation ...

- Basic idea is to identify outliers and give them little or no weight.



Outliers

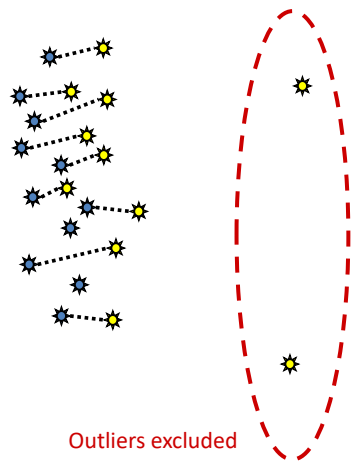
R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," *Comput. Vision, Graphics, Image Processing-IU*, vol. 60, no. 3, pp. 313–342, 1994.

Copyright 2021 R. H. Taylor Engineering Research Center for Computer Integrated Surgical Systems and Technology

66


### Robust Pose Estimation ...

- Basic idea is to identify outliers and give them little or no weight.



Outliers excluded

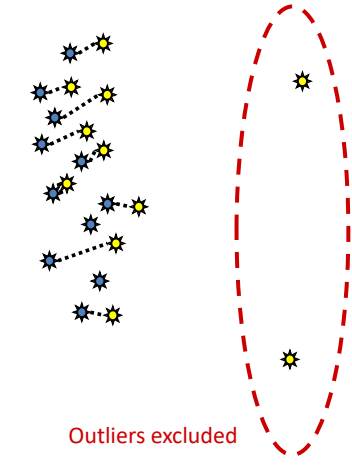
R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," *Comput. Vision, Graphics, Image Processing-IU*, vol. 60, no. 3, pp. 313–342, 1994.

Copyright 2021 R. H. Taylor Engineering Research Center for Computer Integrated Surgical Systems and Technology 

67


### Robust Pose Estimation ...

- Basic idea is to identify outliers and give them little or no weight.



Outliers excluded

R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," *Comput. Vision, Graphics, Image Processing-IU*, vol. 60, no. 3, pp. 313–342, 1994.

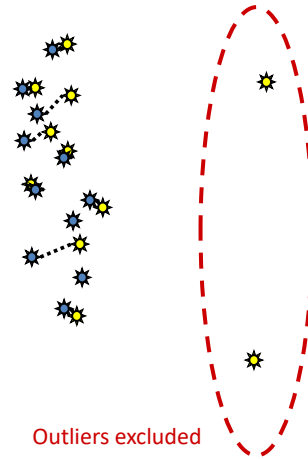
Copyright 2021 R. H. Taylor Engineering Research Center for Computer Integrated Surgical Systems and Technology 

68



## Robust Pose Estimation ...

- Basic idea is to identify outliers and give them little or no weight.



R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," *Comput. Vision, Graphics, Image Processing-IJ*, vol. 60, no. 3, pp. 313–342, 1994.

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



69

## Outline of a practical ICP code

### Given

1. Surface model  $M$  consisting of triangles  $\{m_i\}$
2. Set of points  $Q = \{\bar{q}_1, \dots, \bar{q}_N\}$  known to be on  $M$ .
3. Initial guess  $F_0$  for transformation  $F_0$  such that the points  $F_0 \cdot \bar{q}_k$  lie on  $M$ .
4. Initial threshold  $\eta_0$  for match closeness

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



70

## Outline of a practical ICP code

### Temporary variables

$n$	Iteration number
$\mathbf{F}_n = [\mathbf{R}, \bar{\mathbf{p}}]$	Current estimate of transformation
$\eta_n$	Current match distance threshold
$\mathbf{C} = \{\dots, \bar{\mathbf{c}}_k, \dots\}$	Closest points on $\mathbf{M}$ to $\mathbf{Q}$
$\mathbf{D} = \{\dots, d_k, \dots\}$	Distances $d_k = \ \bar{\mathbf{c}}_k - \mathbf{F}_n \cdot \bar{\mathbf{q}}_k\ $
$\mathbf{I} = \{\dots, i_k, \dots\}$	Indices of triangles $m_{i_k}$ corresp. to $\bar{\mathbf{c}}_k$
$\mathbf{A} = \{\dots, \bar{\mathbf{a}}_k, \dots\}$	Subset of $\mathbf{Q}$ with valid matches
$\mathbf{B} = \{\dots, \bar{\mathbf{b}}_k, \dots\}$	Points on $\mathbf{M}$ corresponding to $\mathbf{A}$
$\mathbf{E} = \{\dots, \bar{\mathbf{e}}_k, \dots\}$	Residual errors $\bar{\mathbf{b}}_k - \mathbf{F} \cdot \bar{\mathbf{a}}_k$
$[\sigma_n, (\epsilon_{\max})_n, \bar{\epsilon}_n]$	$= \left[ \frac{\sum_k \bar{\mathbf{e}}_k \cdot \bar{\mathbf{e}}_k}{\text{NumElts}(\mathbf{E})}; \max_k \sqrt{\bar{\mathbf{e}}_k \cdot \bar{\mathbf{e}}_k}; \frac{\sum_k \sqrt{\bar{\mathbf{e}}_k \cdot \bar{\mathbf{e}}_k}}{\text{NumElts}(\mathbf{E})} \right]$

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



71

## Outline of a practical ICP code

### Step 0: (initialization)

Input surface model  $\mathbf{M}$  and points  $\mathbf{Q}$ .

Build an appropriate data structure (e.g., octree, kD tree)  $\mathbf{T}$   
to facilitate finding the closest point matching search.

$n \leftarrow 0; \quad \eta_n \leftarrow \text{large number}$

$\mathbf{I} \leftarrow \{\dots, 1, \dots\}$

$\mathbf{C} \leftarrow \{\dots, \text{point on } m_1, \dots\}$

$\mathbf{D} \leftarrow \{\dots, \|\bar{\mathbf{c}}_k - \mathbf{F}_0 \cdot \bar{\mathbf{q}}_k\|, \dots\}$

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



72

## Outline of a practical ICP code

### Step 1: (matching)

```

A ← ∅; B ← ∅
For k ← 1 step 1 to N do
  begin
     $bnd_k = \|\mathbf{F}_n \cdot \bar{\mathbf{q}}_k - \bar{\mathbf{c}}_k\|$ 
     $[\bar{\mathbf{c}}_k, i, d_k] \leftarrow \text{FindClosestPoint}(\mathbf{F}_n \cdot \bar{\mathbf{q}}_k, \bar{\mathbf{c}}_k, i_k, bnd_k, \mathbb{T});$ 
    // Note: develop first with simple
    //       search. Later make more
    //       sophisticated, using  $\mathbb{T}$ 
    if ( $d_k < \eta_n$ ) then { put  $\bar{\mathbf{q}}_k$  into A; put  $\bar{\mathbf{c}}_k$  into B; };
    // See also subsequent notes
  end

```

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



73

## Outline of a practical ICP code

### Step 1: (matching)

```

A ← ∅; B ← ∅
For k ← 1 step 1 to N do
  begin
     $bnd_k = \|\mathbf{F}_n \cdot \bar{\mathbf{q}}_k - \bar{\mathbf{c}}_k\|$ 
     $[\bar{\mathbf{c}}_k, i, d_k] \leftarrow \text{FindClosestPoint}(\mathbf{F}_n \cdot \bar{\mathbf{q}}_k, \bar{\mathbf{c}}_k, i_k, bnd_k, \mathbb{T});$ 
    // Note: If using a tree search, you can use
    //       previous match to get a reasonable initial
    //       bound. E.g.,
    //        $bnd_k = \|\bar{\mathbf{c}}_k - \mathbf{F}_n \cdot \bar{\mathbf{q}}_k\|$ 
    //       and then pass that to the tree search.
    //       Alternatively, you can find the closest point
    //       on triangle  $i_k$  and use that to get an initial
    //       bound  $bnd_k$  for the search
    if ( $d_k < \eta_n$ ) then { put  $\bar{\mathbf{q}}_k$  into A; put  $\bar{\mathbf{c}}_k$  into B; };
  end

```

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



74

### Outline of a practical ICP code

#### Step 2 : (transformation update)

$$n \leftarrow n + 1$$

$$\mathbf{F}_n \leftarrow \text{FindBestRigidTransformation}(\mathbf{A}, \mathbf{B})$$

$$\sigma_n \leftarrow \frac{\sqrt{\sum_k \vec{e}_k \cdot \vec{e}_k}}{\text{NumElts}(\mathbb{E})}; \quad (\epsilon_{\max})_n \leftarrow \max_k \sqrt{\vec{e}_k \cdot \vec{e}_k}; \quad \bar{\epsilon}_n \leftarrow \frac{\sum_k \sqrt{\vec{e}_k \cdot \vec{e}_k}}{\text{NumElts}(\mathbb{E})}$$

#### Step 3 : (adjustment)

Compute  $\eta_n$  from  $\{\eta_0, \dots, \eta_{n-1}\}$  // see notes next page

// May also update  $\mathbf{F}_n$  from  $\{\mathbf{F}_0, \dots, \mathbf{F}_n\}$  (see Besl & McKay)

#### Step 4 : (iteration)

if TerminationTest( $\{\sigma_0, \dots, \sigma_n\}, \{(\epsilon_{\max})_0, \dots, (\epsilon_{\max})_n\}, \{\bar{\epsilon}_0, \dots, \bar{\epsilon}_n\}$ )

then stop. Otherwise, go back to step 1 // see notes

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



75

### Outline of practical ICP code

#### Threshold $\eta_n$ update

The threshold  $\eta_n$  can be used to restrict the influence of clearly wrong matches on the computation of  $\mathbf{F}_n$ .

Generally, it should start at a fairly large value and then decrease after a few iterations. One not unreasonable value might be something like  $3\bar{\epsilon}_n$ . If the number of valid matches begins to fall significantly, one can increase it adaptively. Too tight a bound may encourage false minima

Also, if the mesh is incomplete, it may be advantageous to exclude any matches with triangles at the edge of the mesh.

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



76

## Outline of practical ICP code

### Termination test

There are no hard and fast rules for deciding when to terminate the procedure. One criterion might be to stop when  $\sigma_n, \bar{\epsilon}_n$  and/or  $(\epsilon_{\max})_n$  are less than desired thresholds and  $\gamma \leq \frac{\bar{\epsilon}_n}{\epsilon_{n-1}} \leq 1$  for some value  $\gamma$  (e.g.,  $\gamma \cong .95$ ) for several iterations.

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



77

### Short further note: ICP related methods

- There is an extensive literature on methods based on ideas similar to ICP. Surveys and tutorials describing some of them may be found at
  - [http://www.cs.princeton.edu/~smr/papers/fasticp/fasticp\\_paper.pdf](http://www.cs.princeton.edu/~smr/papers/fasticp/fasticp_paper.pdf)
  - [http://www.mrpt.org/Iterative\\_Closest\\_Point\\_%28ICP%29\\_and\\_other\\_matching\\_algorithms](http://www.mrpt.org/Iterative_Closest_Point_%28ICP%29_and_other_matching_algorithms)
- There are also a number of methods that incorporate a probabilistic framework. One example is the “Generalized-ICP” method of Segal, Haehnel, and Thrun
  - Aleksandr V. Segal, Dirk Haehnel, and Sebastian Thrun, “Generalized-ICP”, in *Robotics: Science and Systems*, 2009.
  - [http://www.robots.ox.ac.uk/~avsegal/resources/papers/Generalized\\_ICP.pdf](http://www.robots.ox.ac.uk/~avsegal/resources/papers/Generalized_ICP.pdf)

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



78

### Typical Generalized ICP Algorithm

Outline below is based mostly on from paper by A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP", in *Robotics: Science and Systems*, 2009.

$n \leftarrow 0$ ; initialize  $\mathbf{F}_0$ , threshold value  $\eta_0$ , distribution parameters  $\Phi$

Step 1: (matching)

$\mathbf{A} \leftarrow \emptyset$ ;  $\mathbf{B} \leftarrow \emptyset$

For  $k \leftarrow 1$  step 1 to  $N$  do

begin

$[\vec{\mathbf{c}}_k, i_k, d_k] \leftarrow \text{FindClosestPoint}(\mathbf{F}_n \cdot \vec{\mathbf{q}}_k, \vec{\mathbf{c}}_k, i_k, \mathbf{T})$ ;

if ( $d_k < \eta_n$ ) then { put  $\vec{\mathbf{q}}_k$  into  $\mathbf{A}$ ; put  $\vec{\mathbf{c}}_k$  into  $\mathbf{B}$ ; };

\ \ alternative: test if  $\text{prob}(\vec{\mathbf{q}}_k \sim \vec{\mathbf{c}}_k) > \eta_n$

end

Step 2: (transformation update)

$n \leftarrow n + 1$

$$\begin{aligned} \mathbf{F}_n &\leftarrow \arg \max_{\mathbf{F}} \text{prob}(\mathbf{F} \cdot \mathbf{A} \sim \mathbf{B}; \Phi) = \arg \max_{\mathbf{F}} \prod_i \text{prob}(\mathbf{F} \cdot \vec{\mathbf{a}}_i \sim \vec{\mathbf{b}}_i; \Phi) \\ &= \arg \min_{\mathbf{F}} \sum_i -\log \text{prob}(\mathbf{F} \cdot \vec{\mathbf{a}}_i \sim \vec{\mathbf{b}}_i; \Phi) \end{aligned}$$

Step 3: (adjustment)

update threshold  $\eta_n$  and distribution parameters  $\Phi$

Step 4: (iteration)

if TerminationTest(...) then stop. Otherwise, go back to step 1 // see notes

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



79

### Related concept: Estimation with Uncertainty

Suppose you know something about the uncertainty of the sample data at each point pair (e.g., from sensor noise and/or model error). I.e.,

$$\vec{\mathbf{a}}_k \in \mathbf{A}_k; \vec{\mathbf{b}}_k \in \mathbf{B}_k; \text{cov}(\mathbf{A}_k, \mathbf{B}_k) = \mathbf{C}_k = \mathbf{Q}_k \Lambda_k \mathbf{Q}_k^T$$

Then an appropriate distance metric is the Mahalabonis distance

$$D(\vec{\mathbf{a}}_k, \vec{\mathbf{b}}_k) = (\vec{\mathbf{a}}_k - \vec{\mathbf{b}}_k)^T \mathbf{C}_k^{-1} (\vec{\mathbf{a}}_k - \vec{\mathbf{b}}_k) = \vec{\mathbf{d}}_k^T \Lambda_k^{-1} \vec{\mathbf{d}}_k$$

where

$$\vec{\mathbf{d}}_k = \mathbf{Q}_k^T (\vec{\mathbf{a}}_k - \vec{\mathbf{b}}_k)$$

This approach is readily extended to the case where the samples are not independent.

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



80

## Distance Maps

- Many authors
- Somewhat related to ICP and also to level sets
- Basic idea is to precompute the distance to the surface for a dense sampling of the volume.
- Then use the gradient of the distance map to compute an incremental motion that reduces the sum of the distances of all the moving points to the surface.
- Then iterate

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



81

## Distance Maps

There are a number of very fast algorithms for computing the Euclidean Distance Transform (distance to surface of each point in an image at each point in a 3D volume grid). One example is:

J. C. Torelli, R. Fabbri, G. Travieso, and O. Bruno, "A High Performance 3D Exact Euclidean Distance Transform Algorithm for Distributed Computing", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 24- 6, pp. 897-915, 2010.

But a web search will disclose many others, together with open source code

Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



82

## Distance Maps

Given

a current registration transformation  $\mathbf{F}$

Euclidean distance map  $d(\vec{\mathbf{p}})$

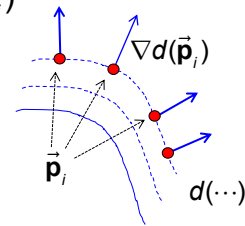
For each sample point  $\vec{\mathbf{f}}_i$  compute  $\vec{\mathbf{p}}_i = \mathbf{F} \cdot \vec{\mathbf{f}}_i$

Compute a small motion  $\Delta\mathbf{F}$

$$\Delta\mathbf{F} = \operatorname{argmin}_{\Delta\mathbf{F}} \sum_i (\Delta\mathbf{F} \cdot \vec{\mathbf{p}}_i - \vec{\mathbf{p}}_i) \cdot \nabla d(\vec{\mathbf{p}}_i)$$

Update  $\mathbf{F} \leftarrow \Delta\mathbf{F} \cdot \mathbf{F}$

Iterate



Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology



84

## Distance Maps

Given

a current registration transformation  $\mathbf{F}$

Euclidean distance map  $d(\vec{\mathbf{p}})$

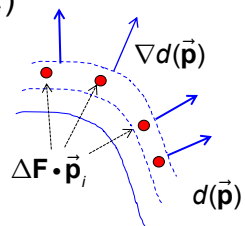
For each sample point  $\vec{\mathbf{f}}_i$  compute  $\vec{\mathbf{p}}_i = \mathbf{F} \cdot \vec{\mathbf{f}}_i$

Compute a small motion  $\Delta\mathbf{F}$

$$\Delta\mathbf{F} = \operatorname{argmin}_{\Delta\mathbf{F}} \sum_i (\Delta\mathbf{F} \cdot \vec{\mathbf{p}}_i - \vec{\mathbf{p}}_i) \cdot \nabla d(\vec{\mathbf{p}}_i)$$

Update  $\mathbf{F} \leftarrow \Delta\mathbf{F} \cdot \mathbf{F}$

Iterate



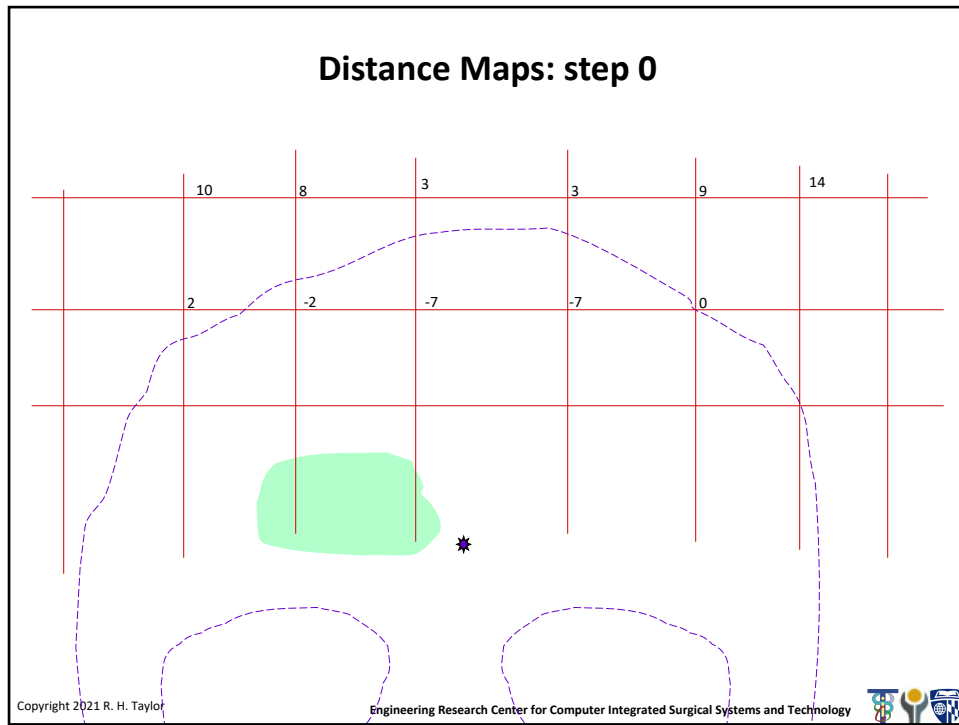
Copyright 2021 R. H. Taylor

Engineering Research Center for Computer Integrated Surgical Systems and Technology

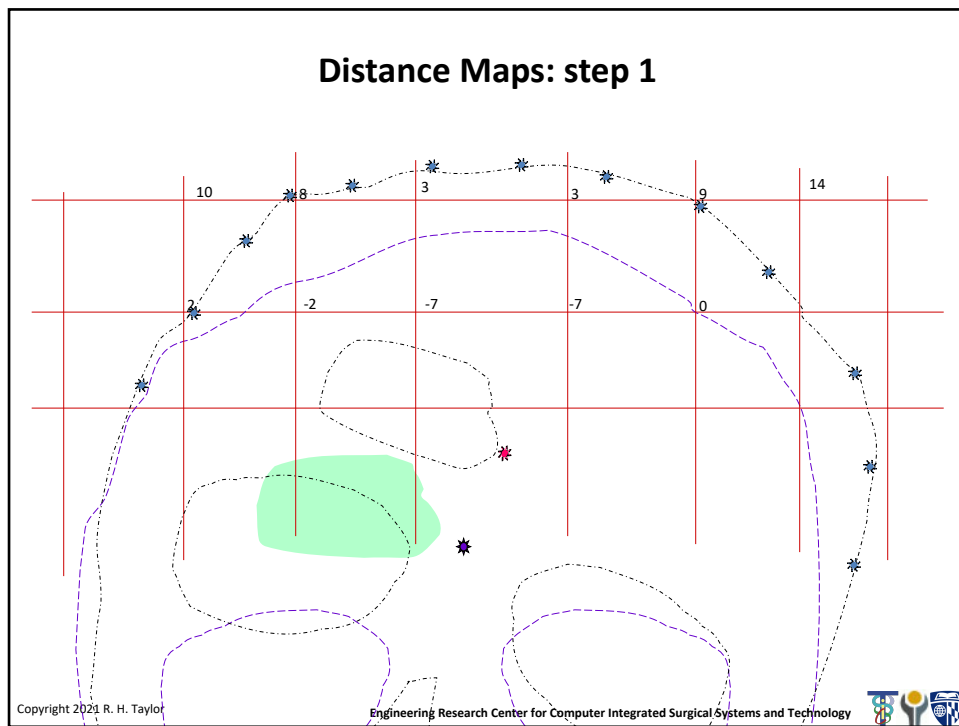


85

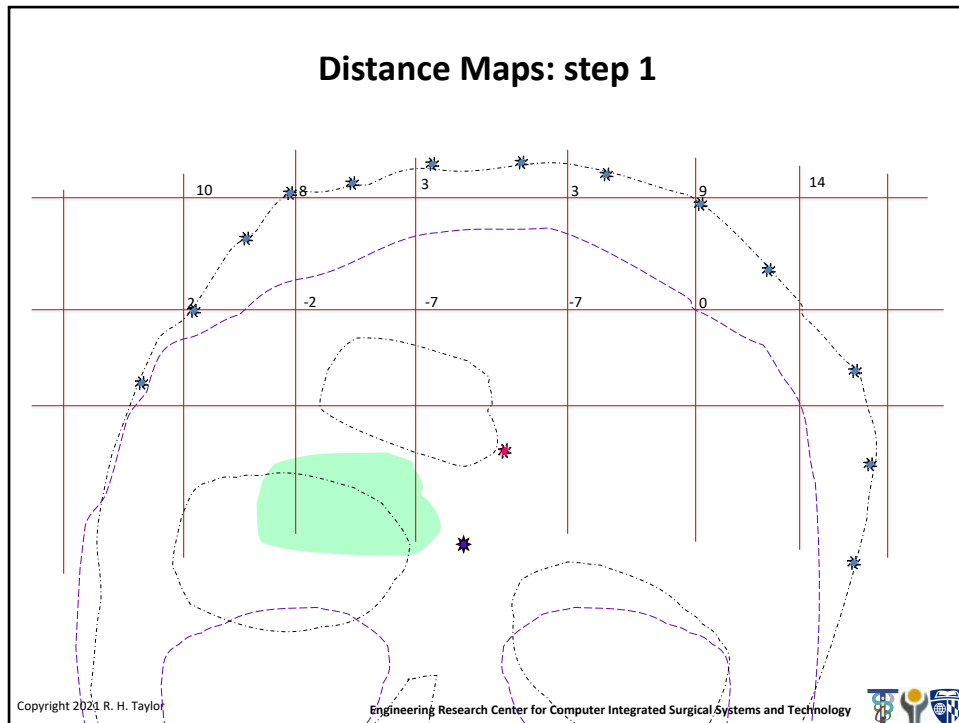




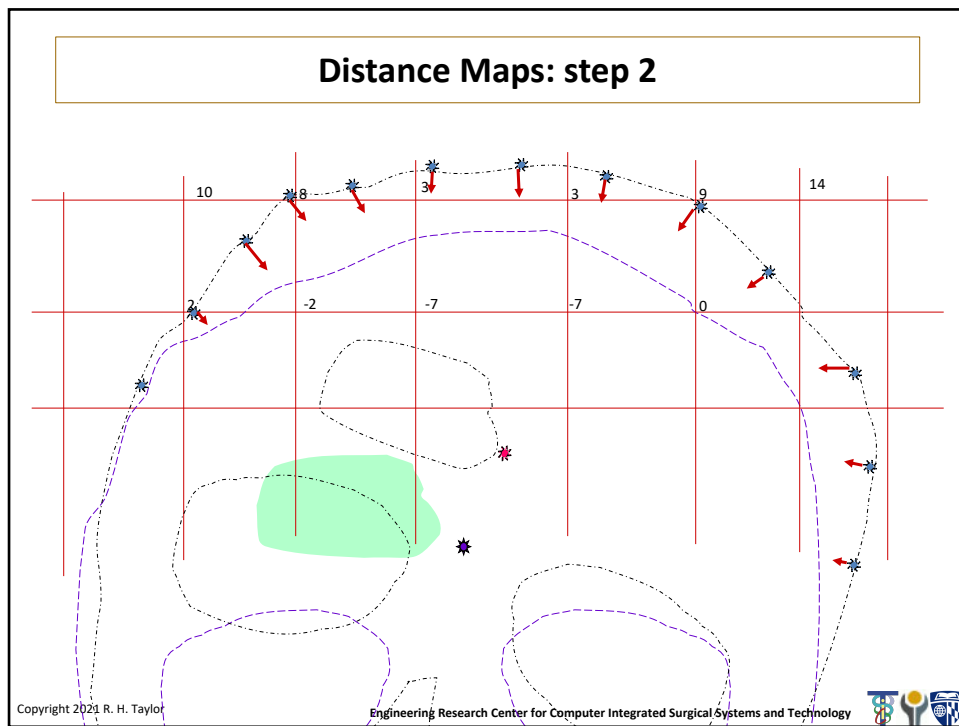
86



87



88



89

