# Image Registration (=Absolute Orientation) for 2D Images or 3D Medical Scans

Lecture by Margrit Betke, CS 585, April 11, 2024

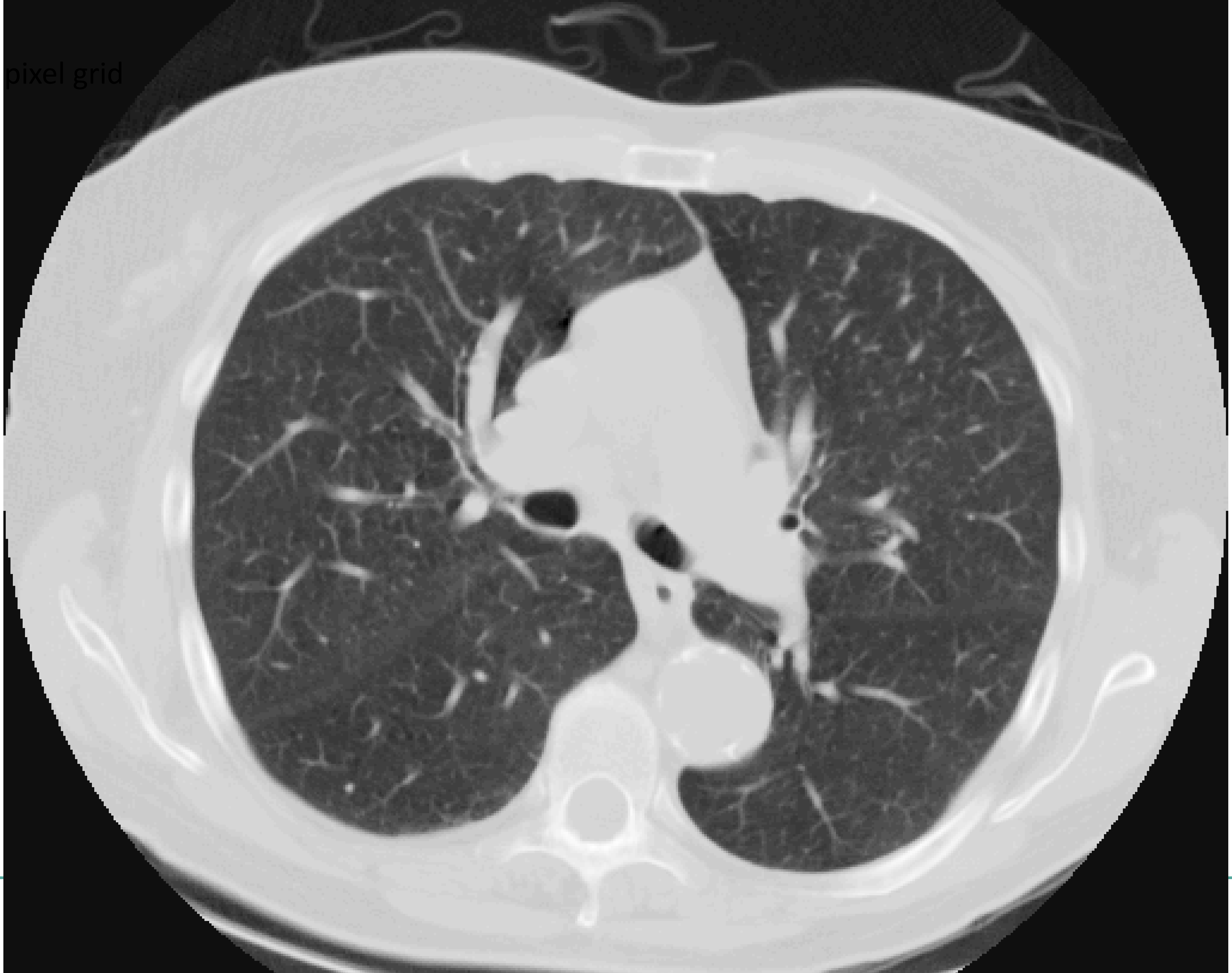
# BU Today, April 14, 2020: Life on the Front Lines – Stories of Doctors at BU Medical Center


Evan Berg, BMC ER physician:

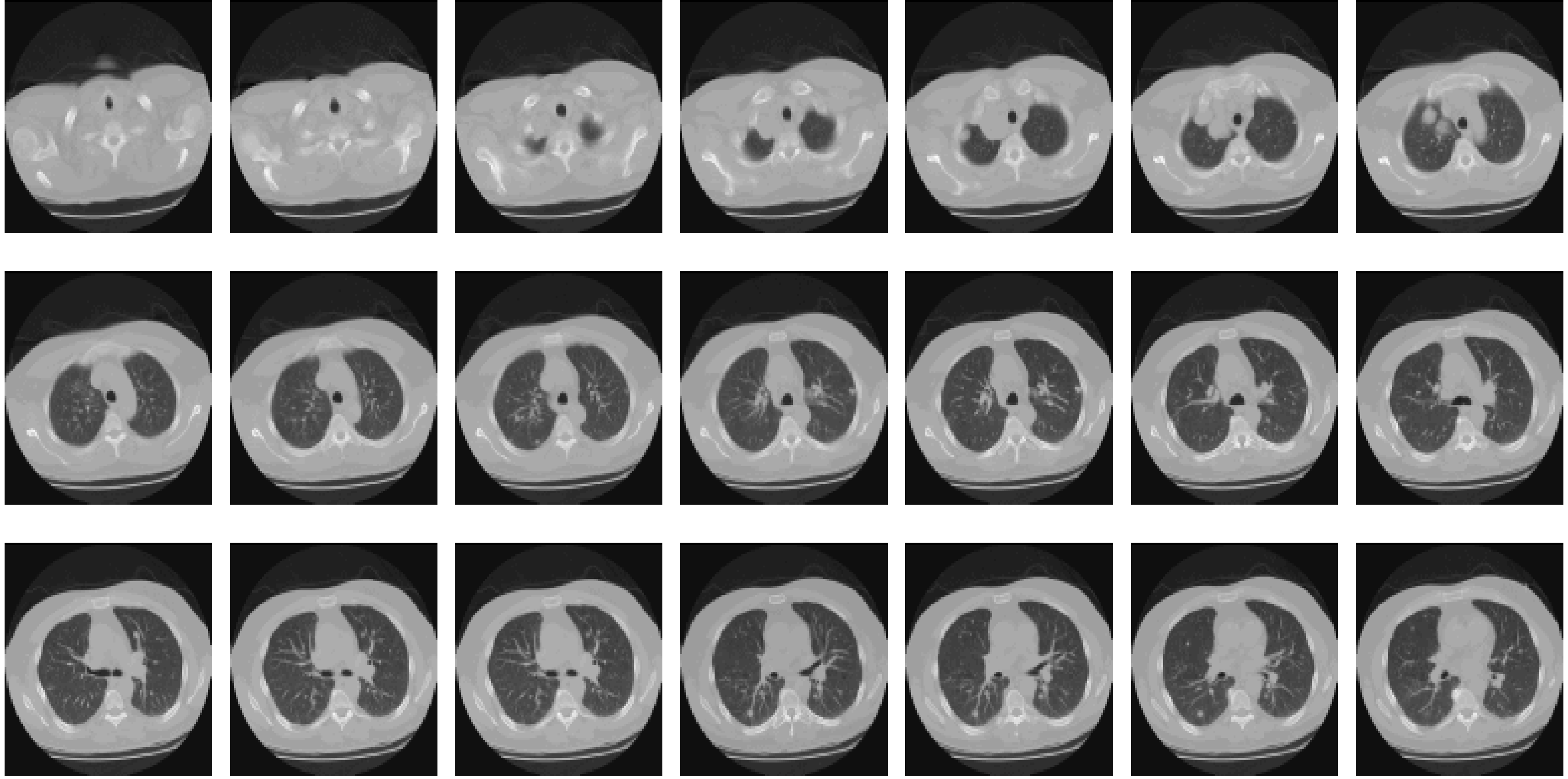
“There is a lot of uncertainty about the [COVID-19] disease process. [..]

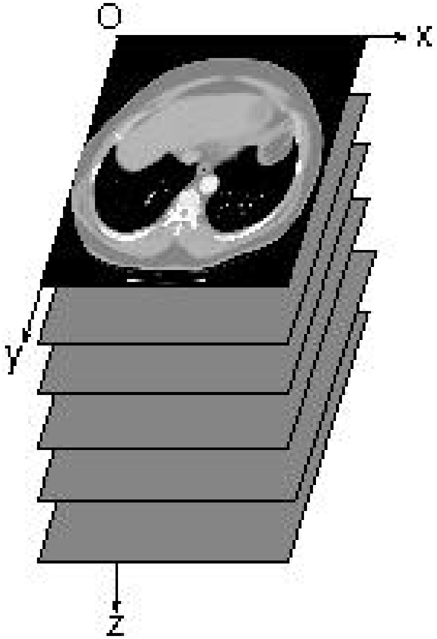
Our understanding around the [..] best practices around treatment, and the outcomes for those needing hospitalization continue to evolve as we learn more and gain experience.”

Image = 2D pixel grid

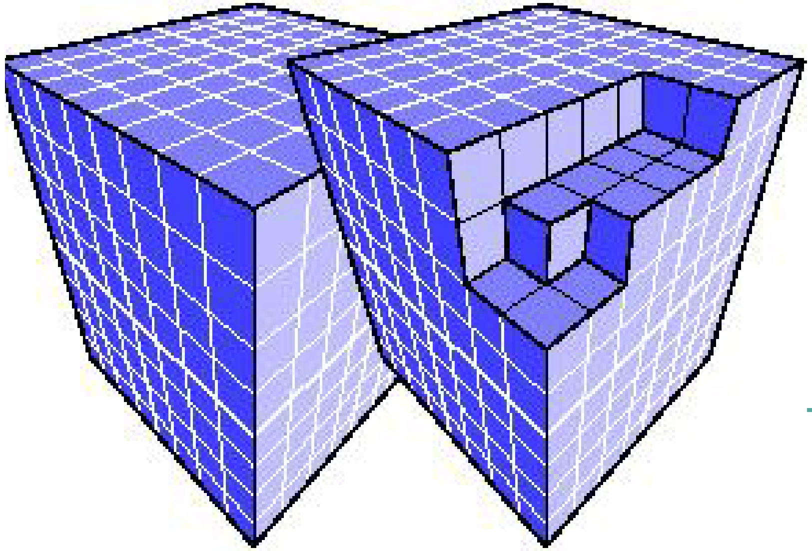


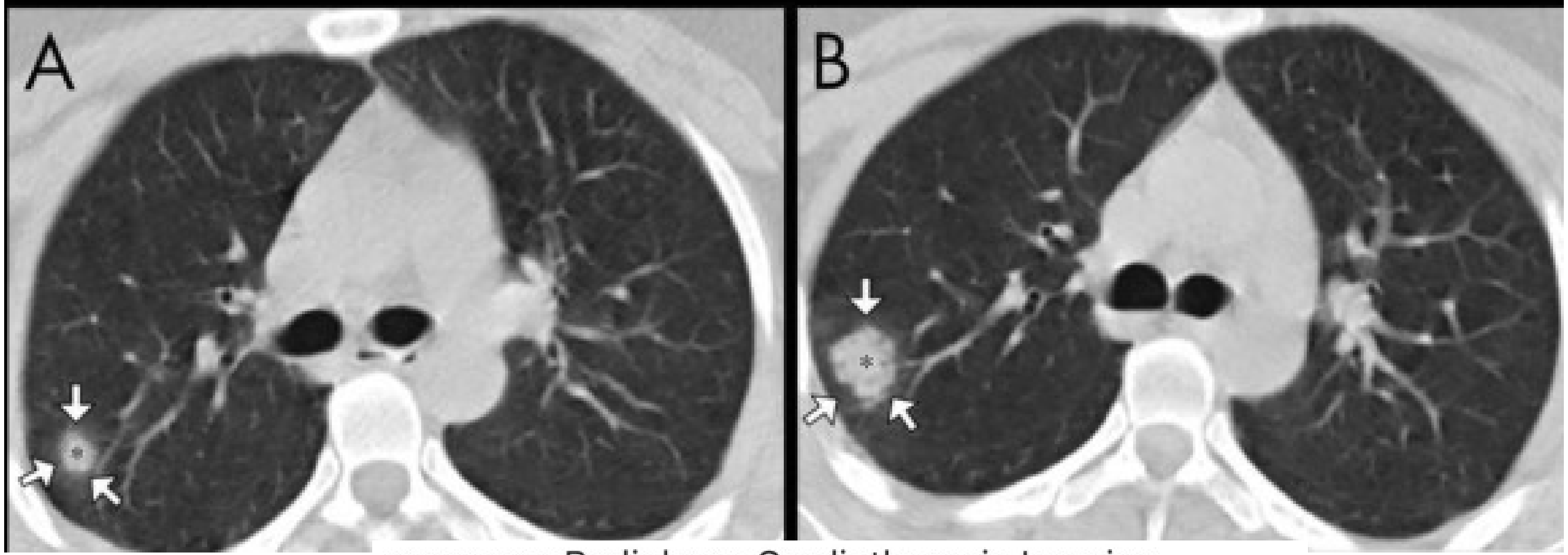
CT scan
=
3D Dataset of
3D voxels that can
be visualized in the
axial plane (as here)






or in the
coronal plane (frontal view)
or
sagittal plane (side view)



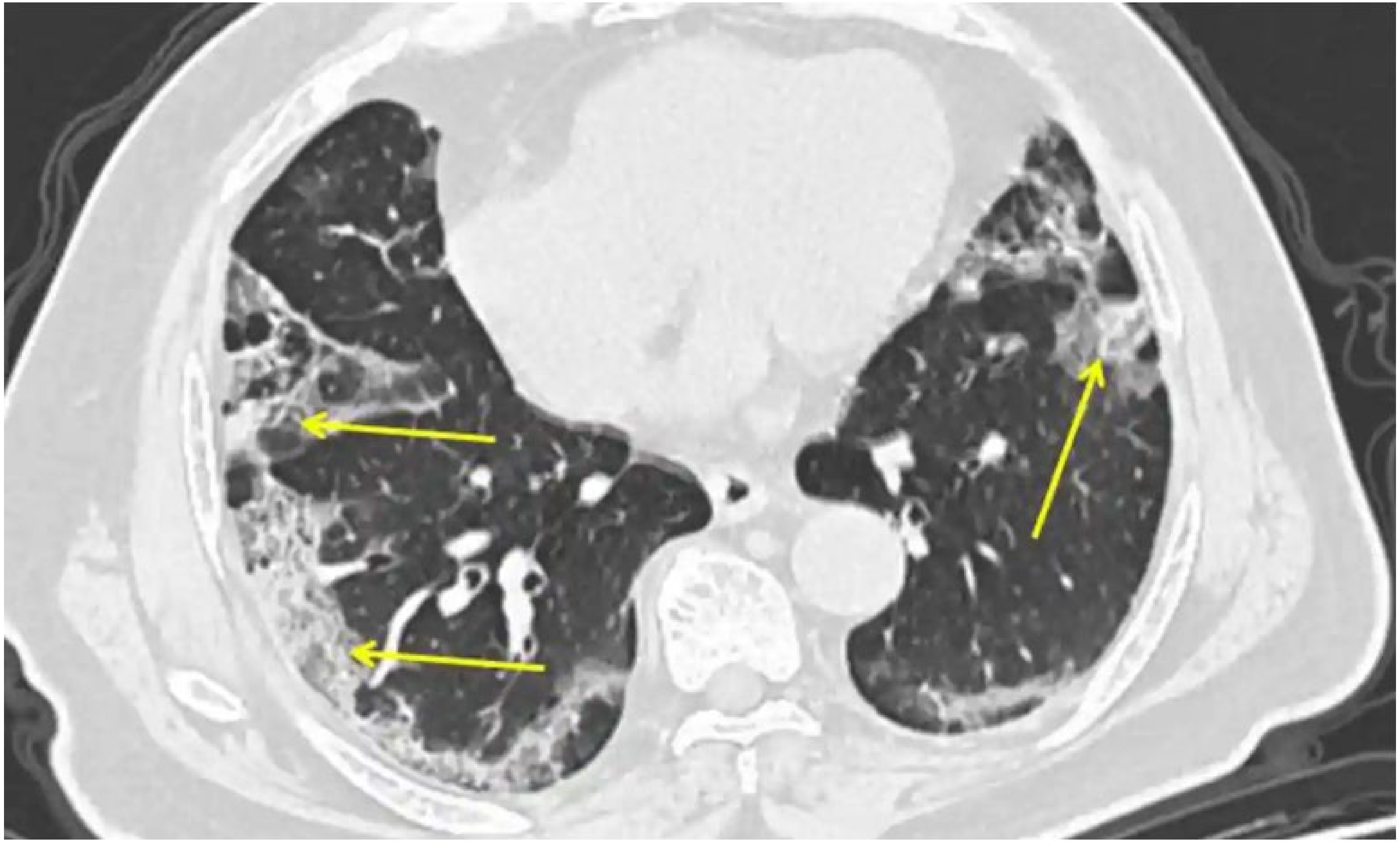



**rsnagram** Radiology: Cardiothoracic Imaging - COVID-19 Infection Presenting with CT Halo Sign

February 12, 2020

Nonenhanced axial chest CT images in a 27-year-old woman. A, Image shows a solid nodule (*) surrounded by a ground-glass halo in the posterior right upper lobe segment (arrows). B, Image at the same level as in A, obtained 4 days after, shows increase in size of the solid nodule (*), with development of small peripheral air bronchograms.

# CT Image of COV-19 Patient

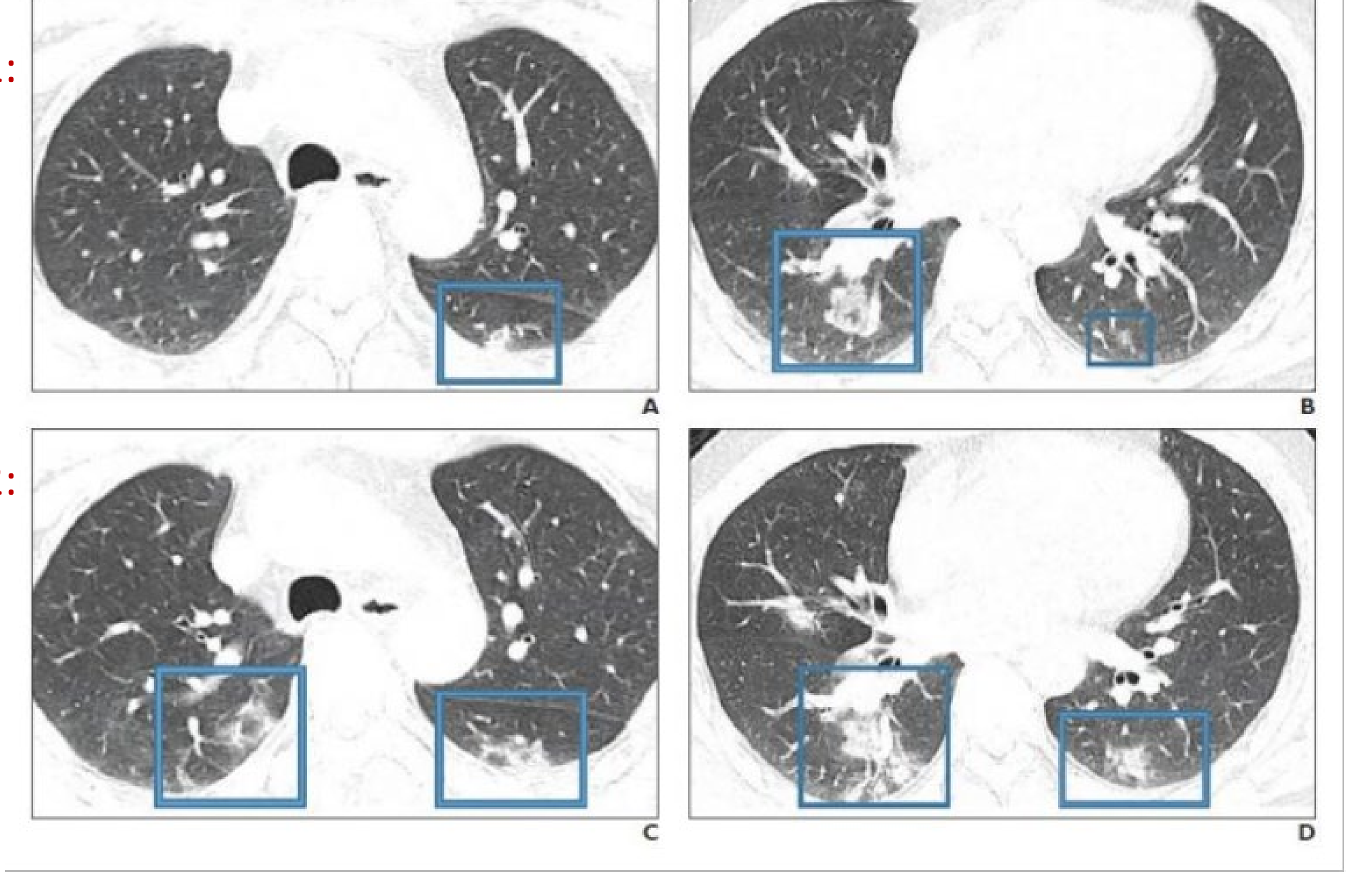


▲ Respiratory physician John Wilson explains the range of Covid-19 impacts. This image shows a CT scan from a man with Covid-19. Pneumonia caused by the new severe acute respiratory coronavirus 2 can show up as distinctive hazy patches on the outer edges of the lungs, indicated by arrows. Photograph: AP

# COVID-19 disease progression visualized on consecutive CT scans

Scan 1:

Scan 2:




CAPTION

A and B, Initial CT images obtained show small round areas of mixed ground-glass opacity and consolidation (rectangles) at level of aortic arch (A) and ventricles (B) in right and left lower lobe posterior zones. C and D, Follow-up CT images obtained 2 days later show progression of abnormalities (rectangles) at level of aortic arch (C) and ventricles (D), which now involve right upper and right and left lower lobe posterior zones.

CREDIT

American Journal of Roentgenology (AJR)

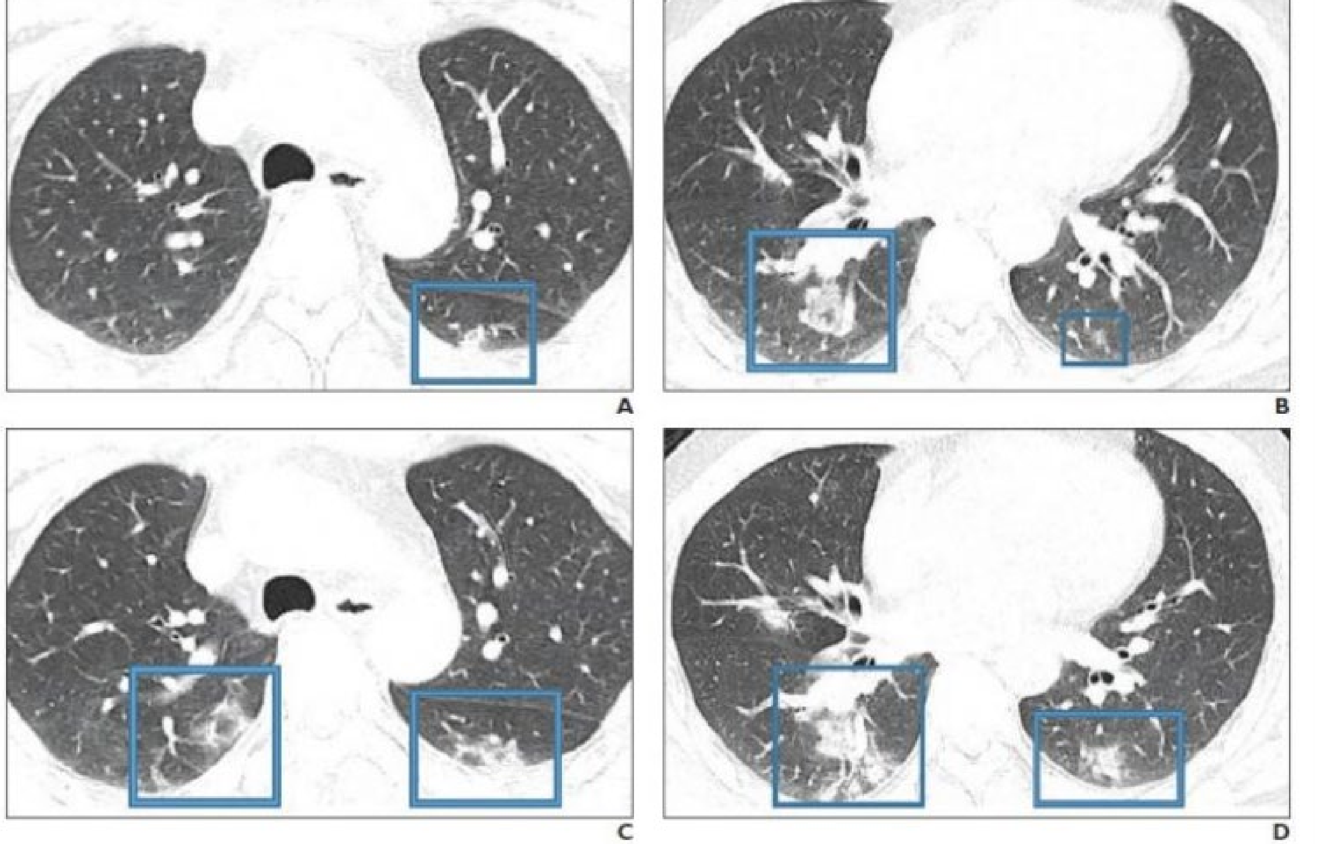
Definition:

A pulmonary consolidation is a region of lung tissue that has filled with liquid instead of air.

Normally soft tissue found in the aerated lung has hardened (white regions in images).

Scan 1:

Scan 2:




CAPTION

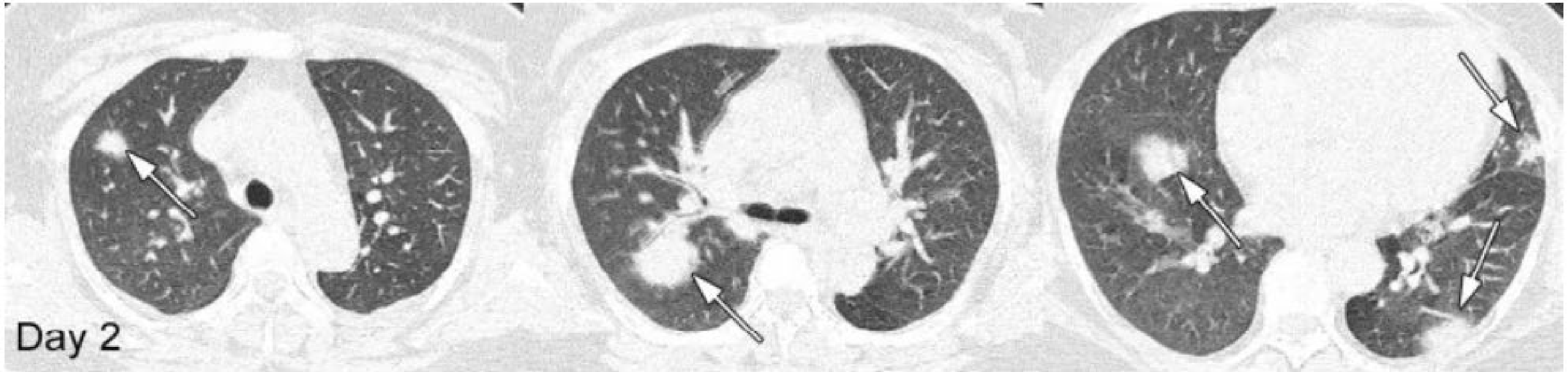
A and B, Initial CT images obtained show small round areas of mixed ground-glass opacity and consolidation (rectangles) at level of aortic arch (A) and ventricles (B) in right and left lower lobe posterior zones. C and D, Follow-up CT images obtained 2 days later show progression of abnormalities (rectangles) at level of aortic arch (C) and ventricles (D), which now involve right upper and right and left lower lobe posterior zones.

CREDIT

American Journal of Roentgenology (AJR)

# COV-19 Patient improves on follow-up CT scan:

## Ground-glass opacities




Lungs of a 54-year-old woman who presented with fever on day 2 of symptoms.

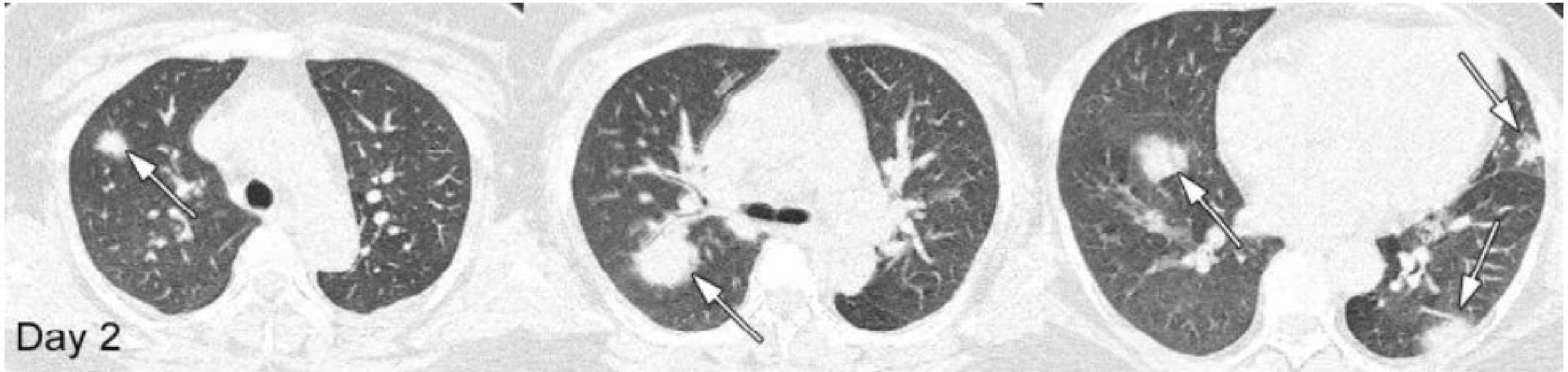
WU Y, XIE Y-I, WANG X. LONGITUDINAL CT FINDINGS IN COVID-19 PNEUMONIA: CASE PRESENTING ORGANIZING PNEUMONIA PATTERN. RADIOLOGY: CARDIOTHORACIC IMAGING DOI: 10.1148/RYCT.2020200031. PUBLISHED ONLINE FEBRUARY 1, 2020. © RADIOLOGICAL SOCIETY OF NORTH AMERICA.


A follow-up CT on day 11, three days after initiation of antiviral treatment, showed significant improvement of the ground-glass opacities.

WU Y, XIE Y-I, WANG X. LONGITUDINAL CT FINDINGS IN COVID-19 PNEUMONIA: CASE PRESENTING ORGANIZING PNEUMONIA PATTERN. RADIOLOGY: CARDIOTHORACIC IMAGING DOI: 10.1148/RYCT.2020200031. PUBLISHED ONLINE FEBRUARY 1, 2020. © RADIOLOGICAL SOCIETY OF NORTH AMERICA.

**Ground-glass opacities** on CT are hazy regions that do not obscure the underlying bronchial structures or pulmonary vessels.

A ground-glass opacity indicates a partial filling of air spaces in the lungs by **exudate** (seeped out fluid) or **transudate** (pushed out fluid), as well as interstitial thickening or partial collapse of lung **alveoli** (tiny air sacs of the lungs which allow for rapid gaseous exchange).




Lungs of a 54-year-old woman who presented with fever on day 2 of symptoms.

WU Y, XIE Y-I, WANG X. LONGITUDINAL CT FINDINGS IN COVID-19 PNEUMONIA: CASE PRESENTING ORGANIZING PNEUMONIA PATTERN. RADIOLOGY: CARDIOTHORACIC IMAGING DOI: 10.1148/RYCT.20202000031. PUBLISHED ONLINE FEBRUARY 1, 2020.



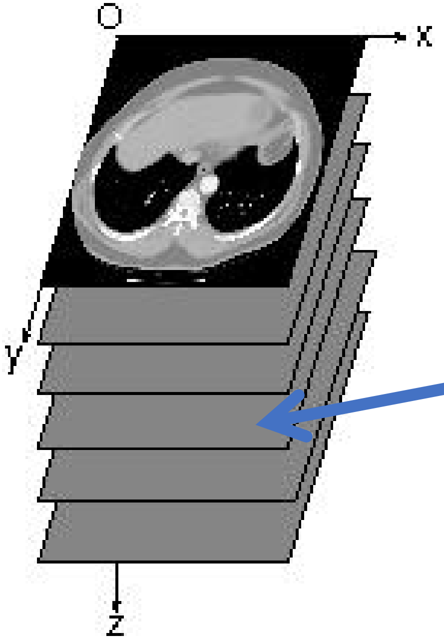
A follow-up CT on day 11, three days after initiation of antiviral treatment, showed significant improvement of the ground-glass opacities.

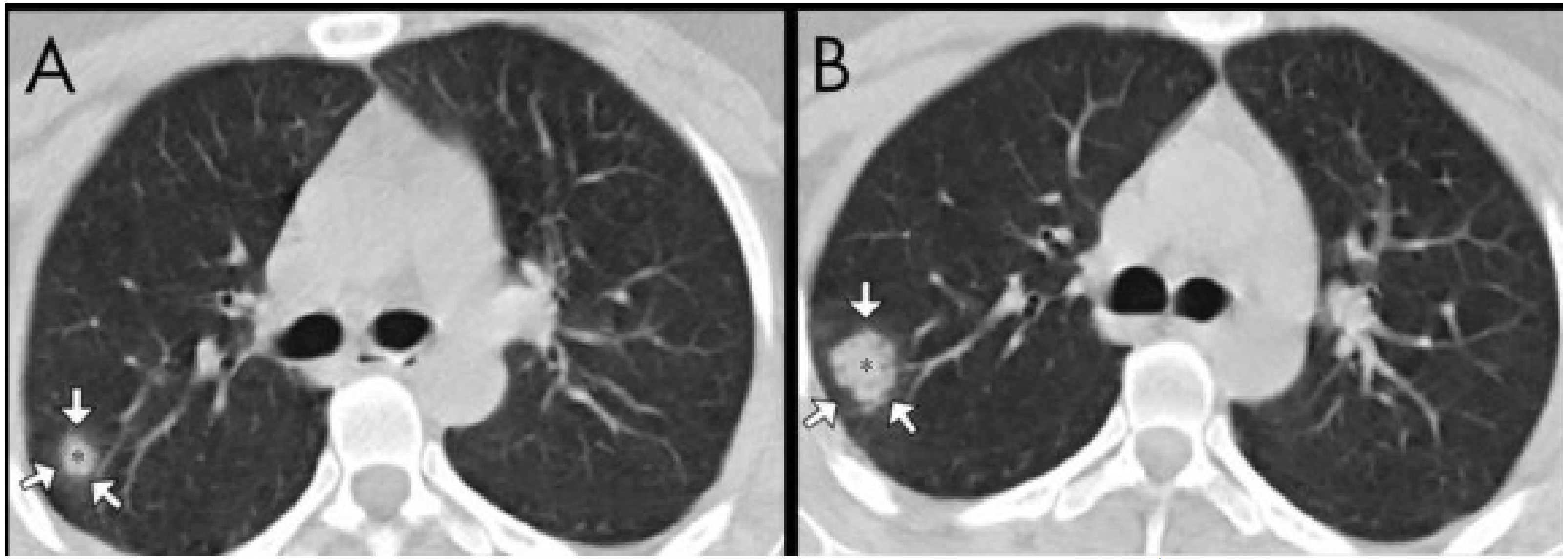
WU Y, XIE Y-I, WANG X. LONGITUDINAL CT FINDINGS IN COVID-19 PNEUMONIA: CASE PRESENTING ORGANIZING PNEUMONIA PATTERN. RADIOLOGY: CARDIOTHORACIC IMAGING DOI: 10.1148/RYCT.20202000031. PUBLISHED ONLINE FEBRUARY 1, 2020.
© RADIOLOGICAL SOCIETY OF NORTH AMERICA.

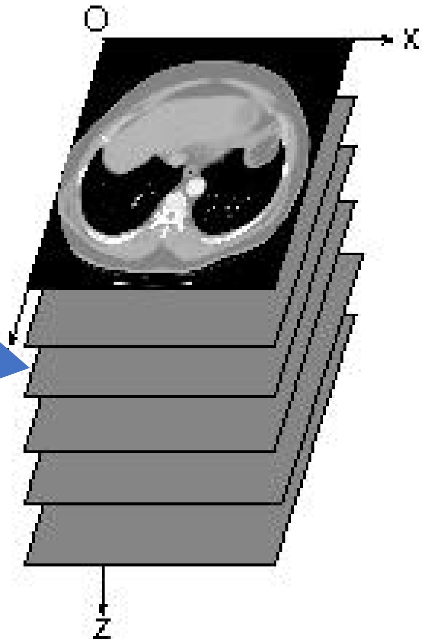
# AI-supported Evaluation of Consecutive CT Scans requires

- 2D Absolute Orientation = Image Registration = Image Alignment
- 3D Absolute Orientation = 3D Registration = 3D Scan Alignment

# Alignment of 2D Images vs 3D scans




Comparison more accurate if done in 3D

COVID-19 CT Scan Evaluation Process:

Given a pair of corresponding images:

Compute  Alignment

Semi-automatic Evaluation:

1. Radiologist annotates region of interest with consolidation in scan 1. AI system finds all pixels (voxels) with consolidation in scan 2

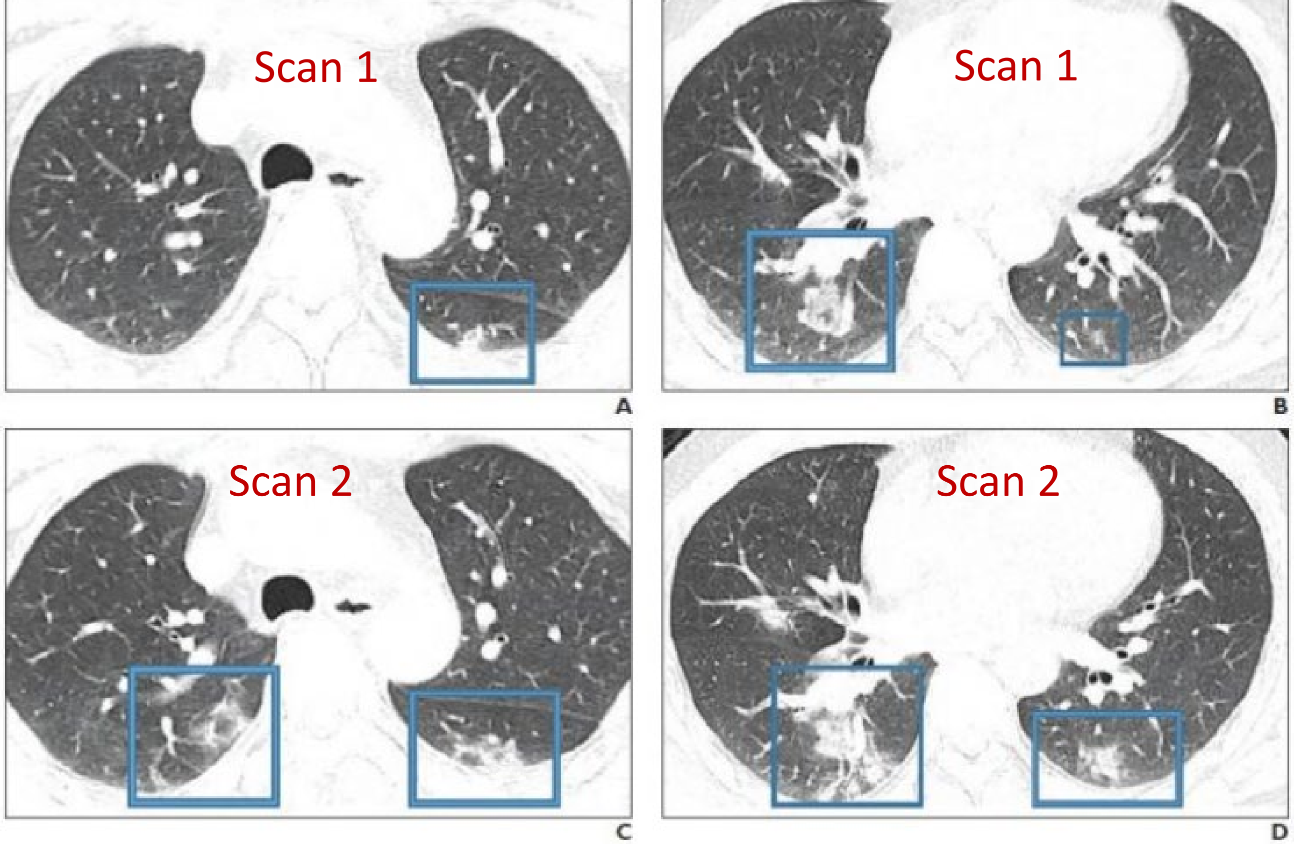
OR:

Automatic Evaluation:

AI system detects & measures the volume of the consolidation in scan 1 and scan 2

AI system evaluates change in size of region:  Increase: Patient fares worse

Decrease: Patient is getting better




CAPTION

A and B, Initial CT images obtained show small round areas of mixed ground-glass opacity and consolidation (rectangles) at level of aortic arch (A) and ventricles (B) in right and left lower lobe posterior zones. C and D, Follow-up CT images obtained 2 days later show progression of abnormalities (rectangles) at level of aortic arch (C) and ventricles (D), which now involve right upper and right and left lower lobe posterior zones.

CREDIT

American Journal of Roentgenology (AJR)

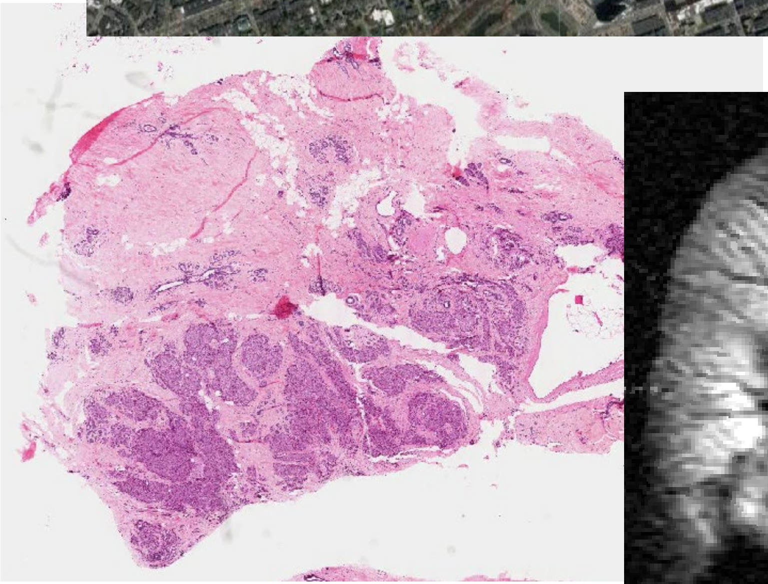
BOSTON UNIVERSITY

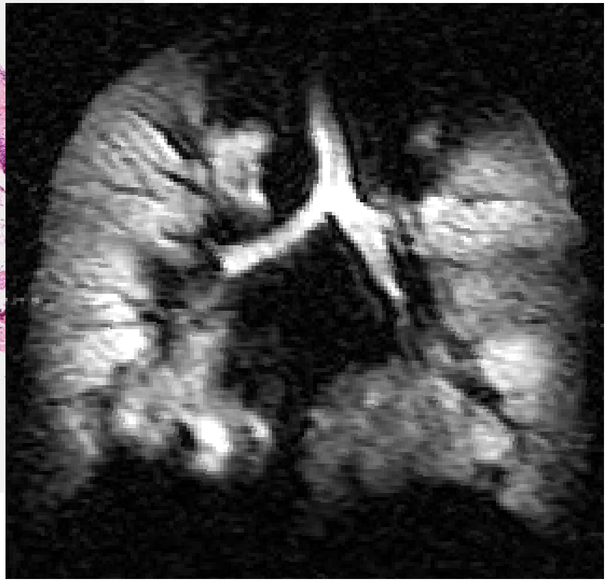
# Additional Terminology:

- 2D Landmarks = 2D Feature Locations in Image (pixels)
- 3D Landmarks = 3D Feature Locations in 3D Scan (voxels)

# Other Use Cases

Photogrammetry = Analysis of images (often photographs) to extract geometric information (often for map building) about (typically) physical objects and their environment





Biomedical Image Analysis = Computer Vision for Medicine (diagnosis, prognosis, treatment, and prevention of disease)

# 2D Alignment (= Absolute Orientation in 2D)

Definition:

Compute the parameters that describe a “2D rigid body alignment”

= Compute an angle that describes the rotation of the object in the image plane and a 2D translation vector that moves the object into the desired location.

Practical Use Cases:

- Registration of anatomical structures on images of the same patient taken at different times for measuring response to treatment
    - Lung cancer patients (Patent: "Method and system for the detection, comparison and volumetric quantification of pulmonary nodules on medical computed tomography scans." Inventors Margrit Betke and Jane P. Ko., US Patent 7,206,462, issued on April 17, 2007.)
- Match left and right images in a stereo camera system
- Photogrammetry = Build maps from overlapping photographs
- Compute rigid body motion of objects under the microscope

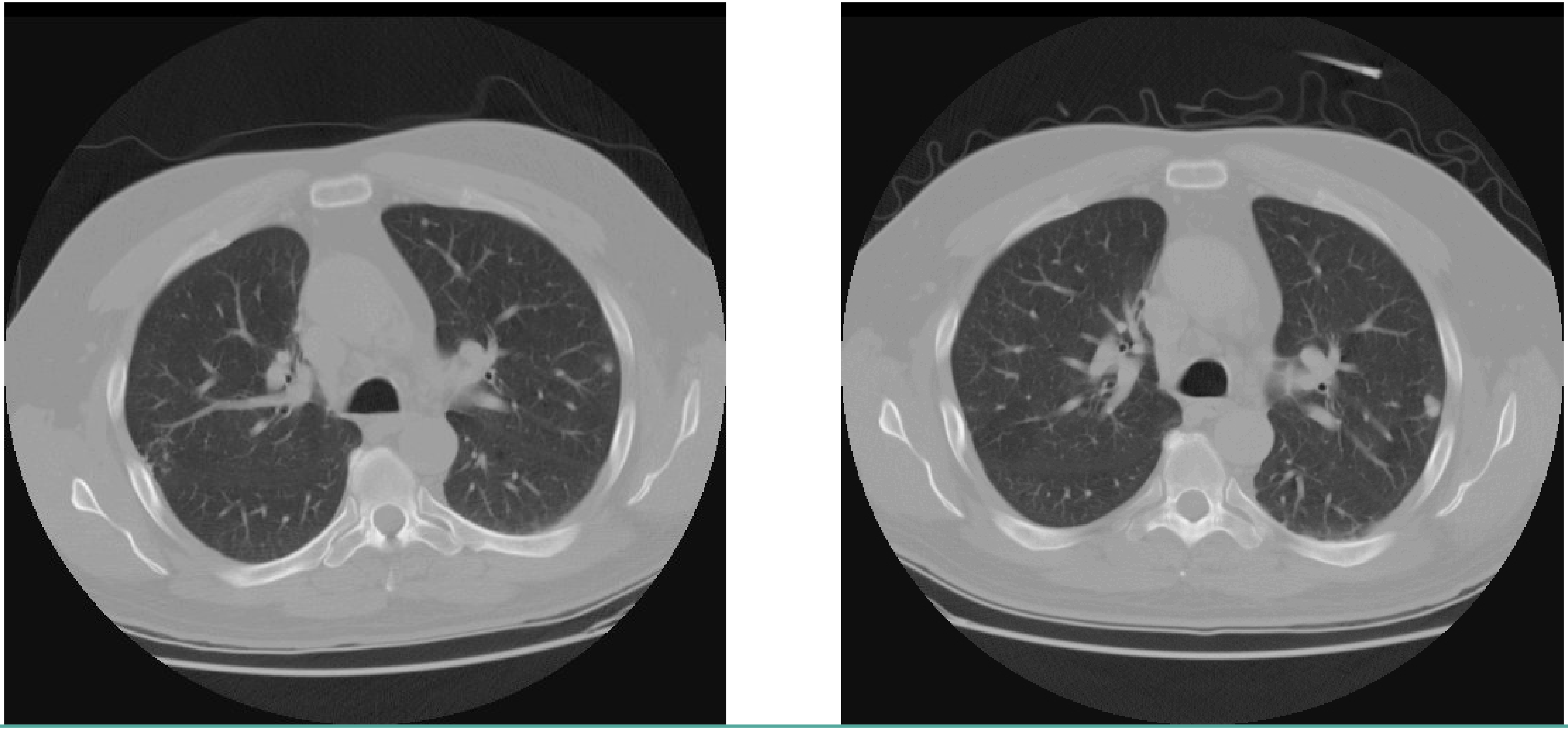
# Absolute Orientation in 2D

Specific Definition:

Compute the parameters that describe a 2D rigid body alignment

= Compute a rotation angle $\theta$ and a 2D translation vector $(x_o, y_o)^T$

# Example:  Registration of Lung Images



# Registration of lung images of a cancer patient taken three months apart to measure nodule growth

Scan 1

Scan 2

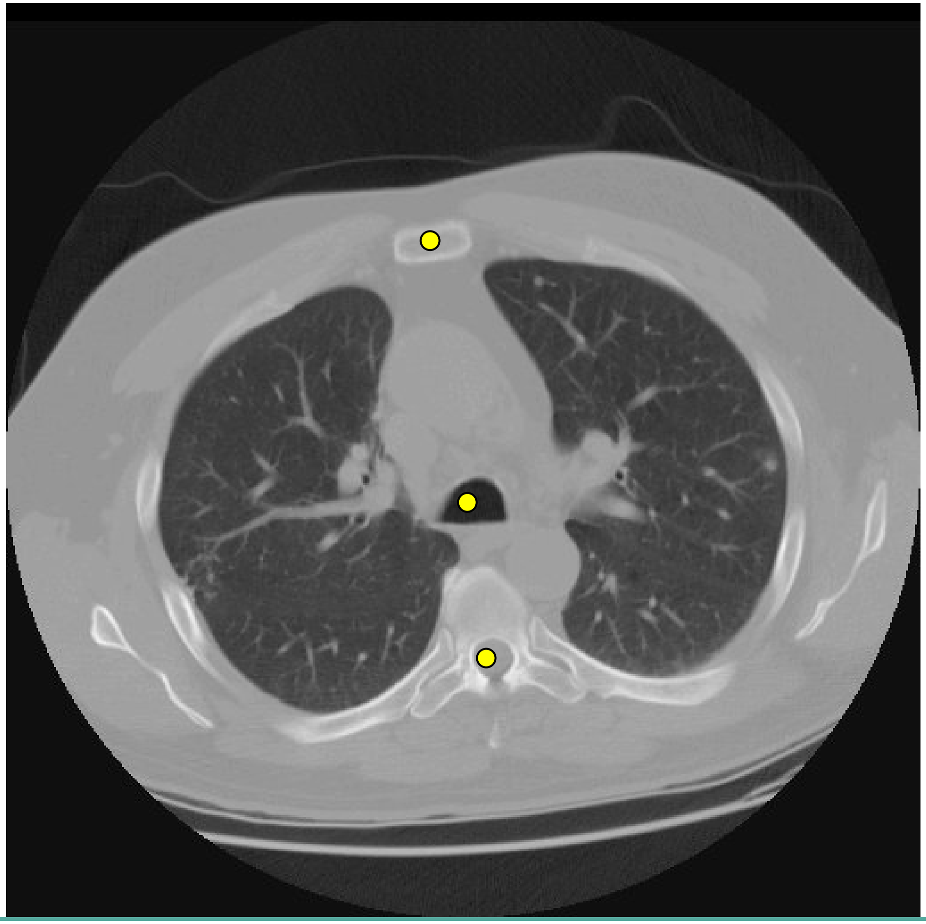
# 2D Absolute Orientation

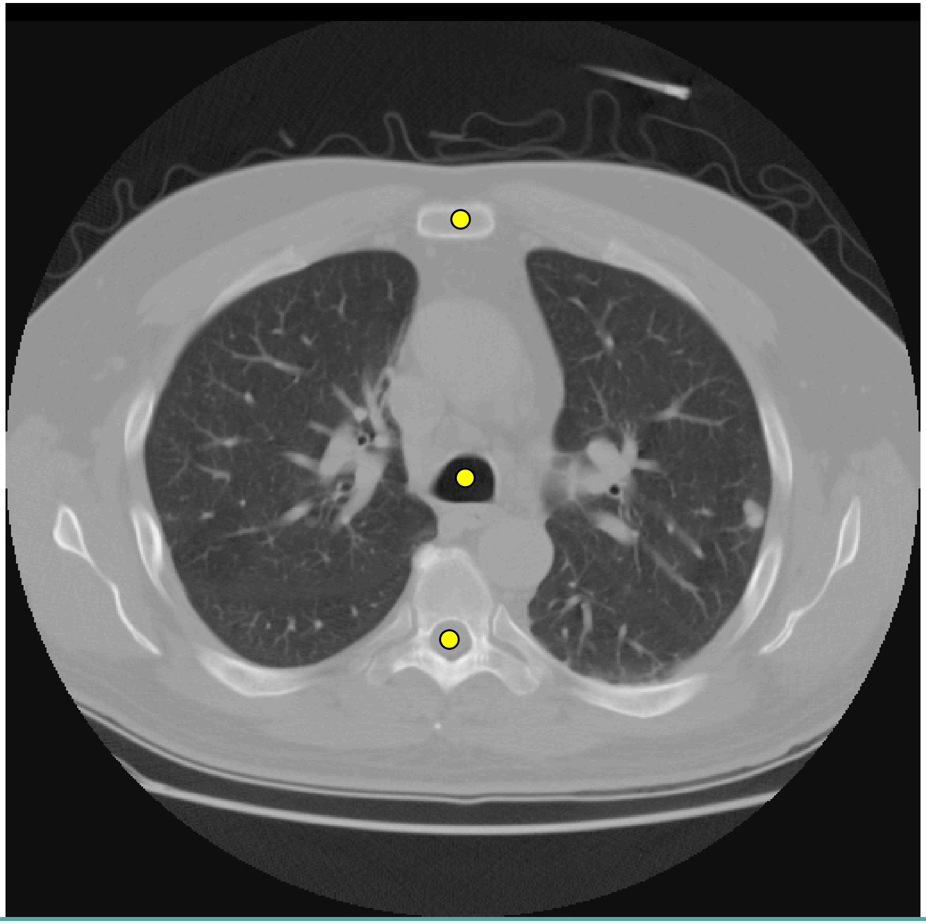
Unknowns: Rotation angle $\theta$ and translation vector $(x_o, y_o)^T$

Number of unknowns: 3

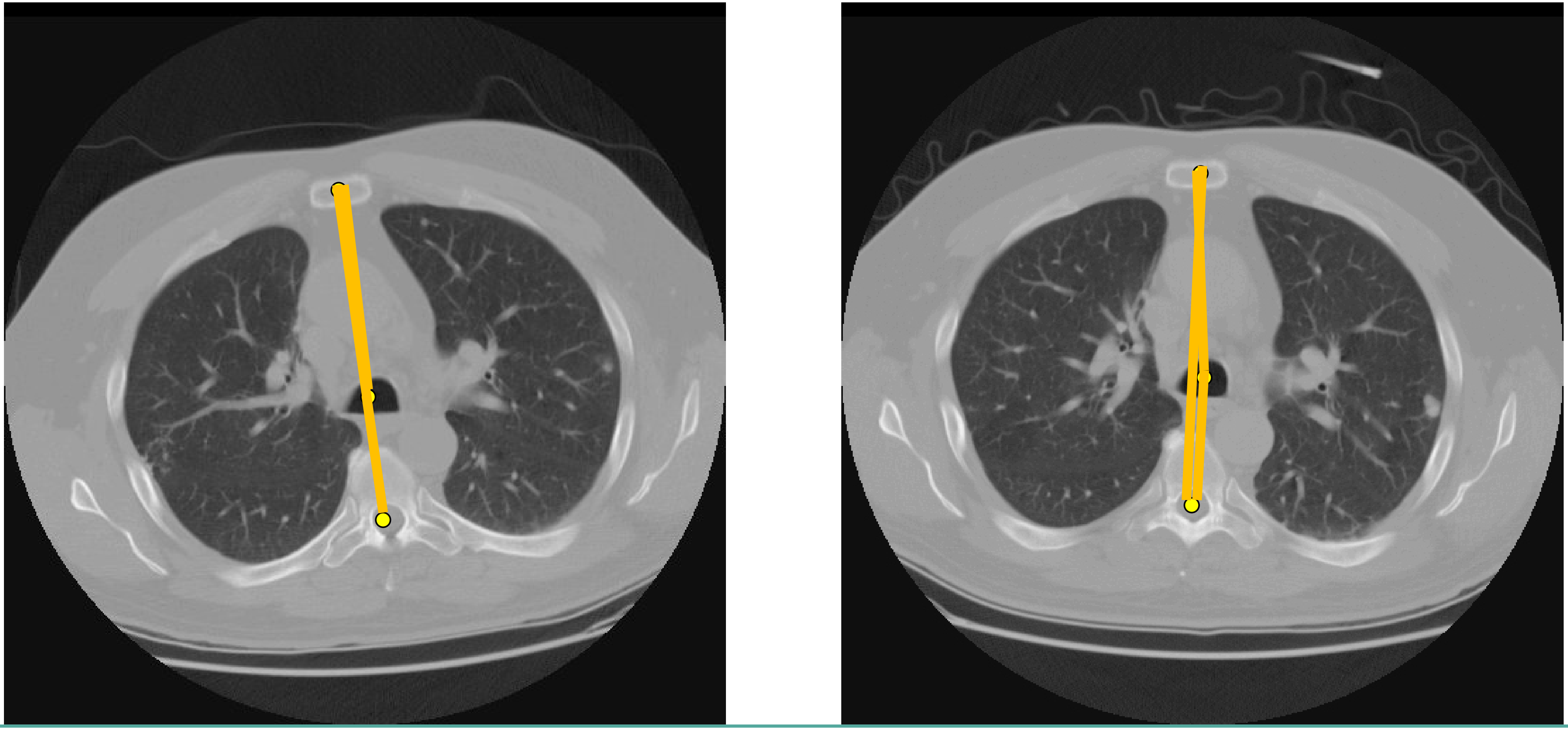
$\Rightarrow$ We need at least 3 equations and 3 landmarks

# How about these three landmarks?

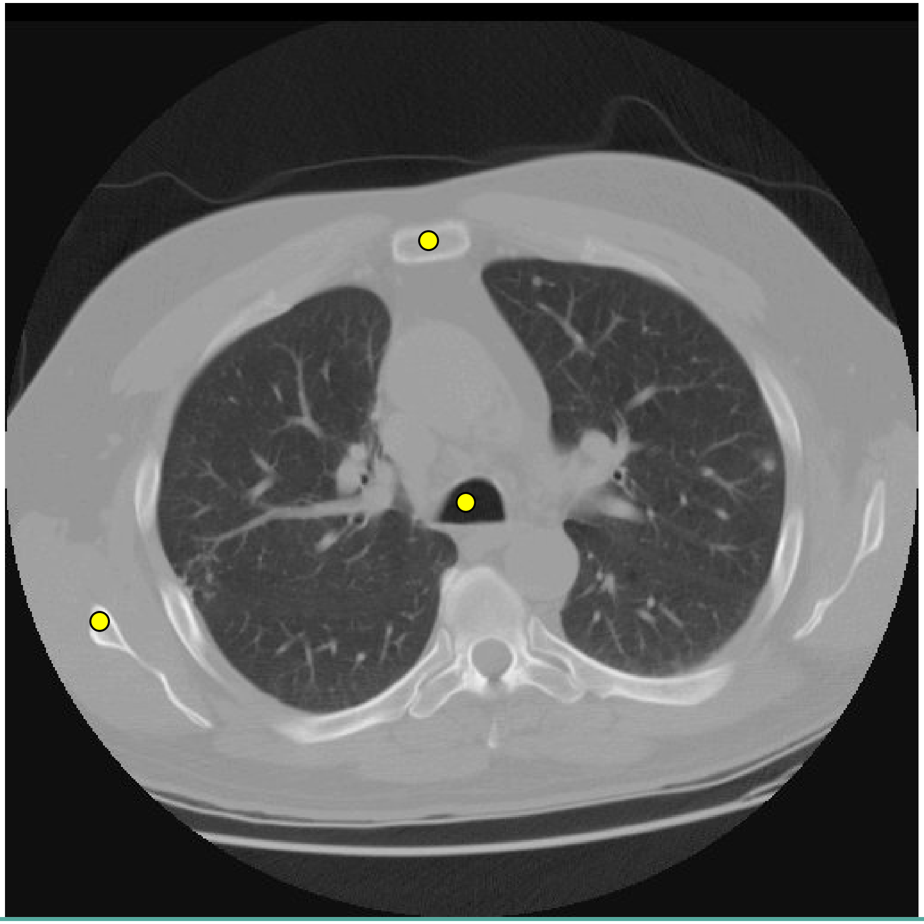


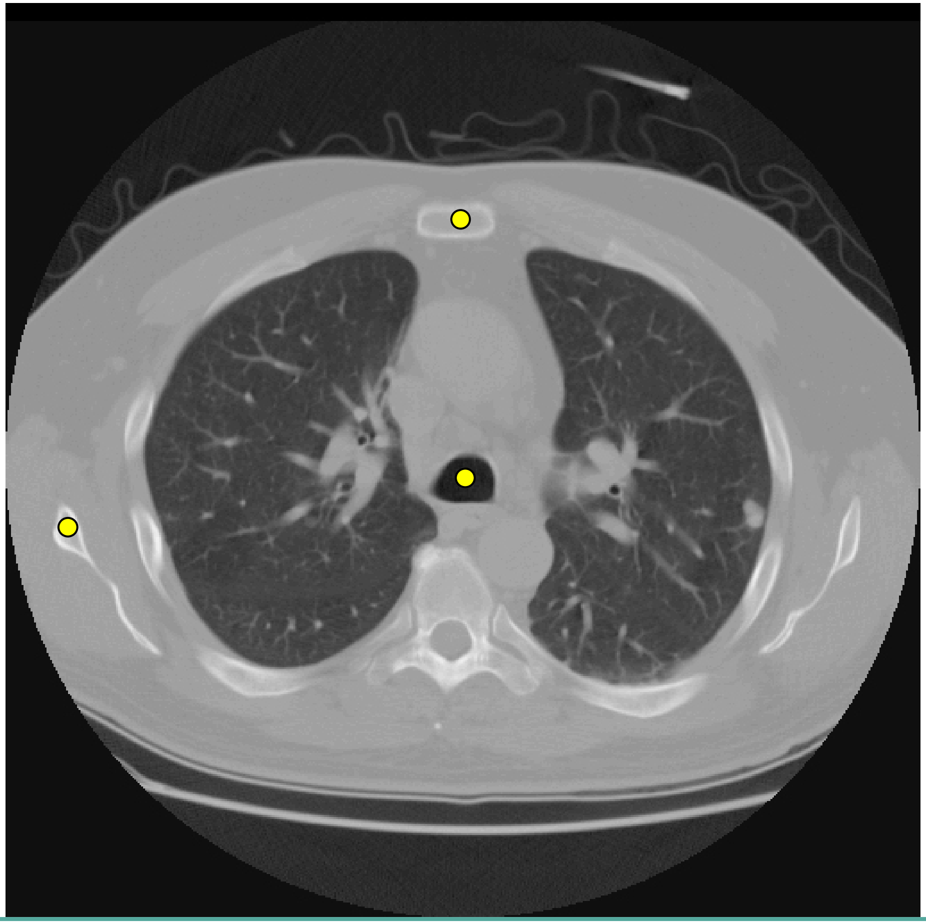


# Angle difficult to discern:

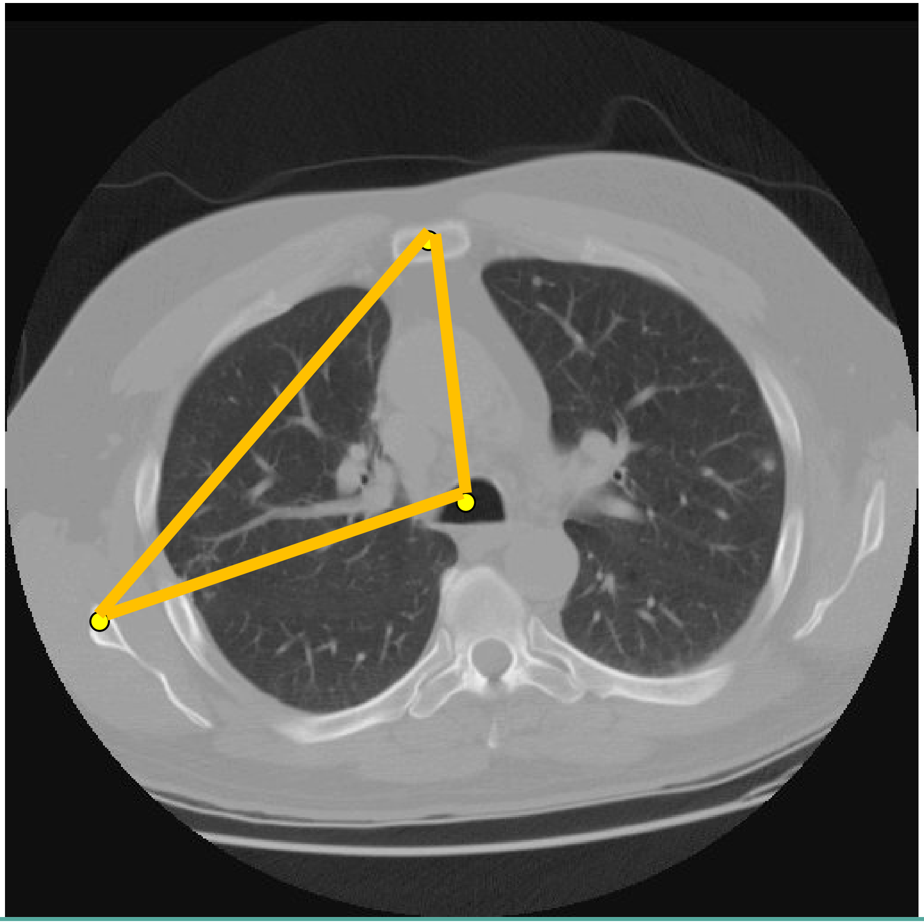


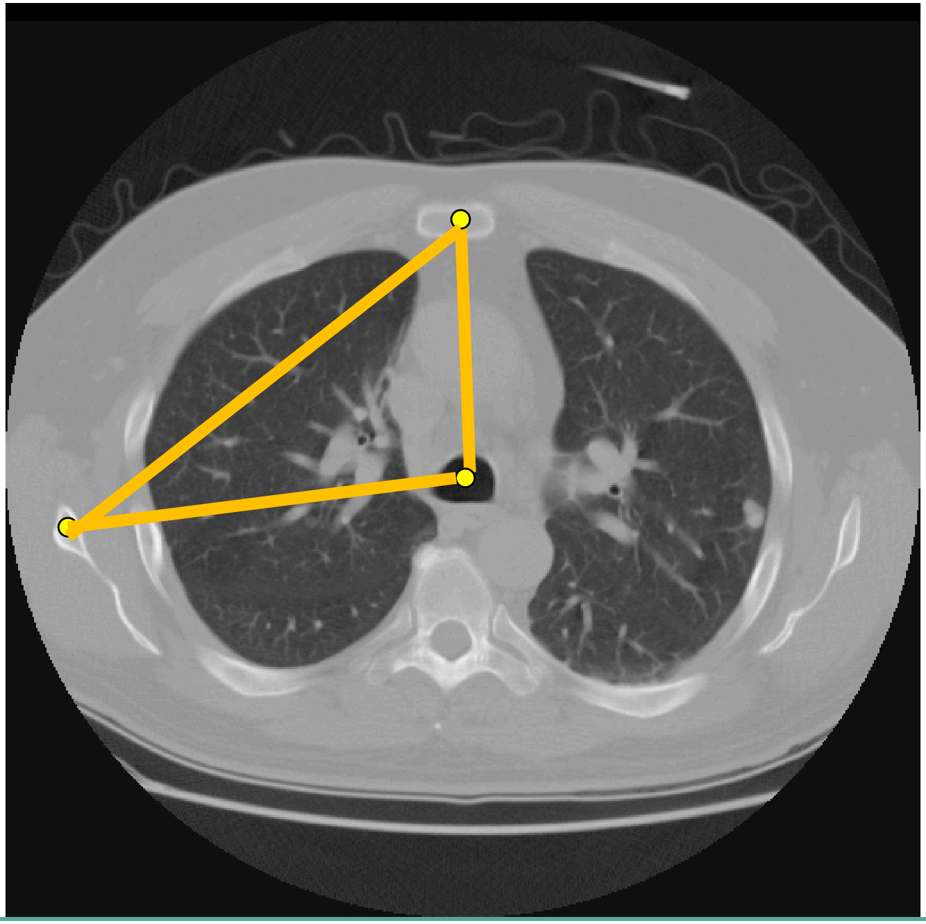
# Better these three landmarks:





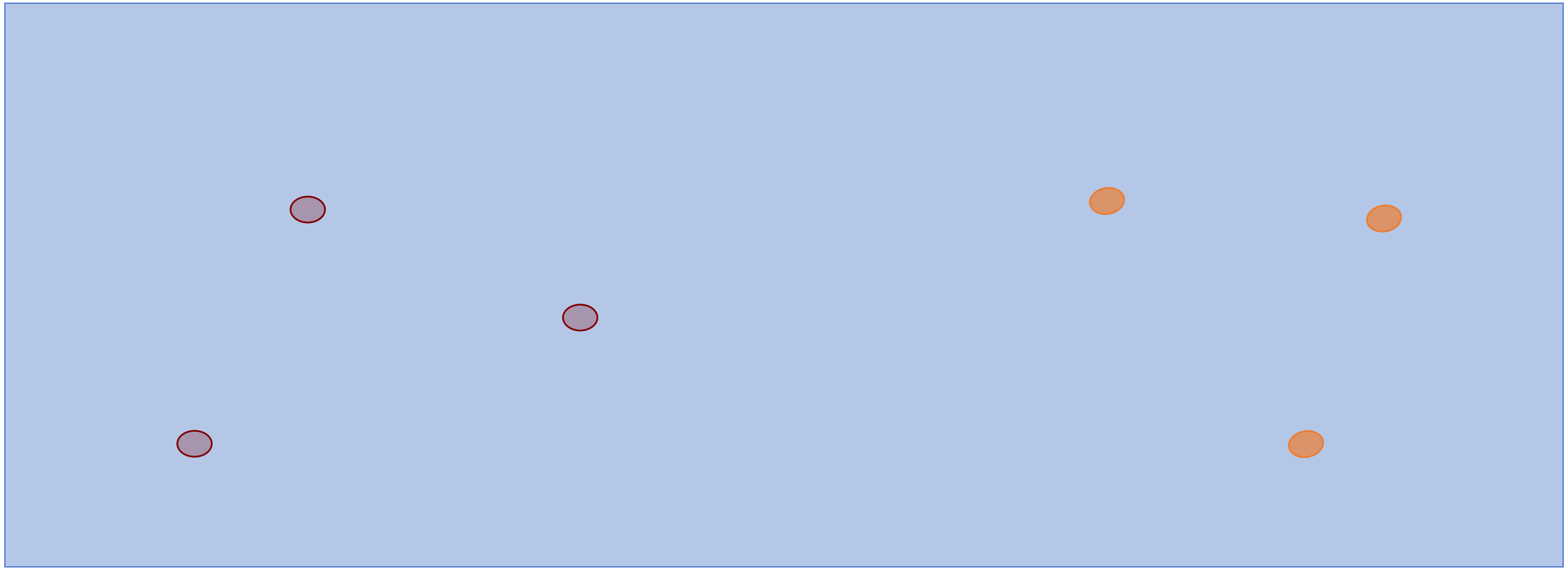
# Better these three landmarks:





# Algorithm Idea for 2D Alignment

First translation, then rotation:



# How can we represent rotation?

Derivation of Rotation Matrix:

Rotation in the image plane by angle $\theta$ :

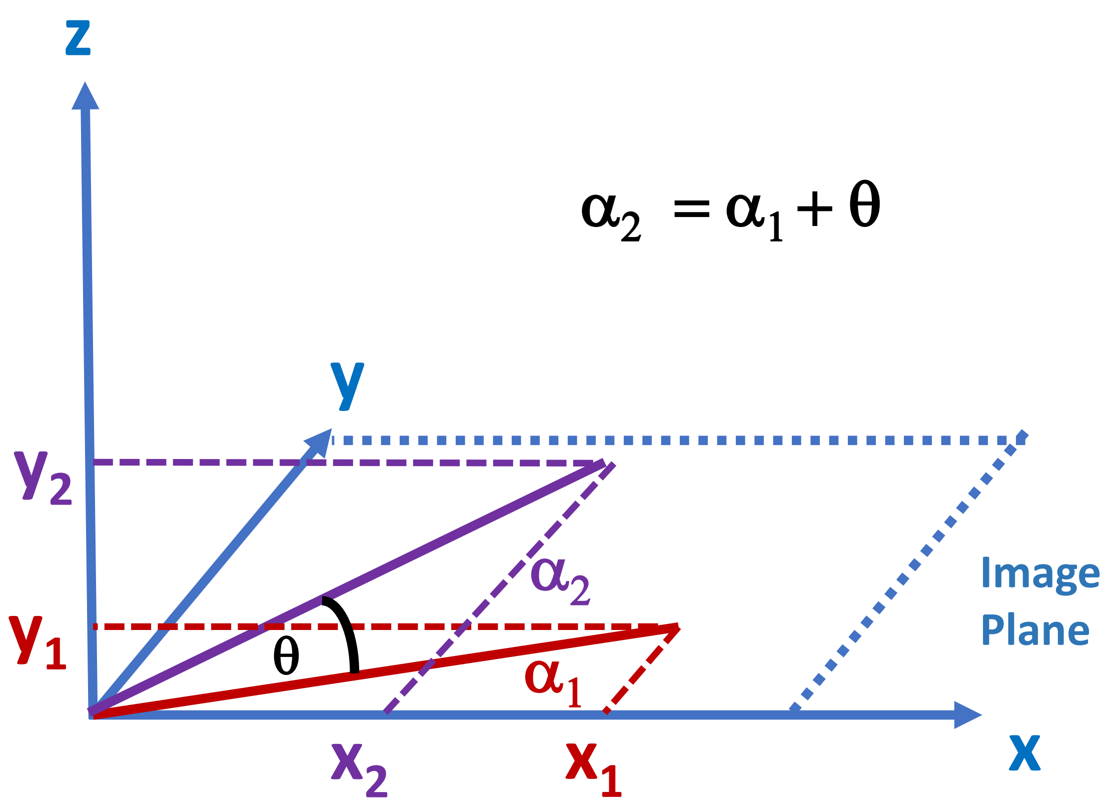
Original Position:

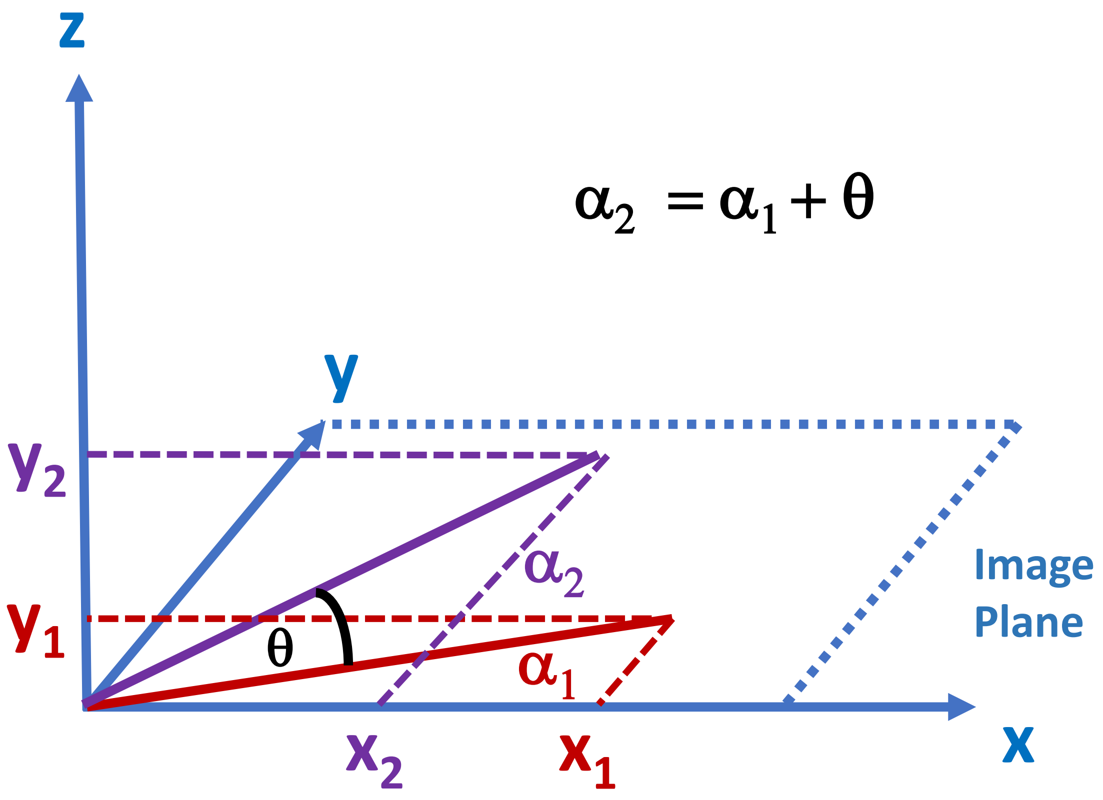
$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} r\cos\alpha_1 \\ r\sin\alpha_1 \\ 1 \end{pmatrix}$$

Rotated Position:

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} r\cos\alpha_2 \\ r\sin\alpha_2 \\ 1 \end{pmatrix}$$

$$\alpha_2 = \alpha_1 + \theta$$






$$\alpha_2 = \alpha_1 + \theta$$

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} r\cos\alpha_2 \\ r\sin\alpha_2 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} r\cos(\alpha_1+\theta) \\ r\sin(\alpha_1+\theta) \\ 1 \end{pmatrix} = \begin{pmatrix} r\cos\alpha_1\cos\theta - r\sin\alpha_1\sin\theta \\ r\sin\alpha_1\cos\theta + r\cos\alpha_1\sin\theta \\ 1 \end{pmatrix}$$

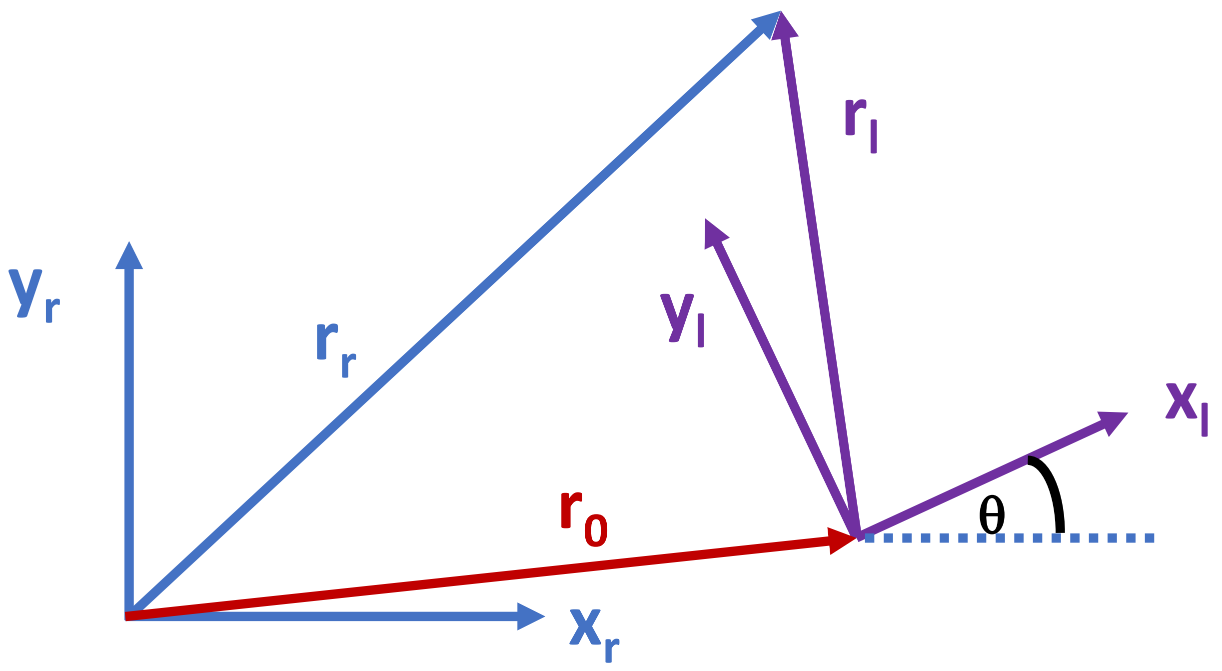
$$= \begin{pmatrix} x_1\cos\theta - y_1\sin\theta \\ y_1\cos\theta + x_1\sin\theta \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

$R^T R = RR^T = I$

R is an orthonormal matrix = columns (or rows) add up to 1 and are perpendicular to each other (dot product = 0)

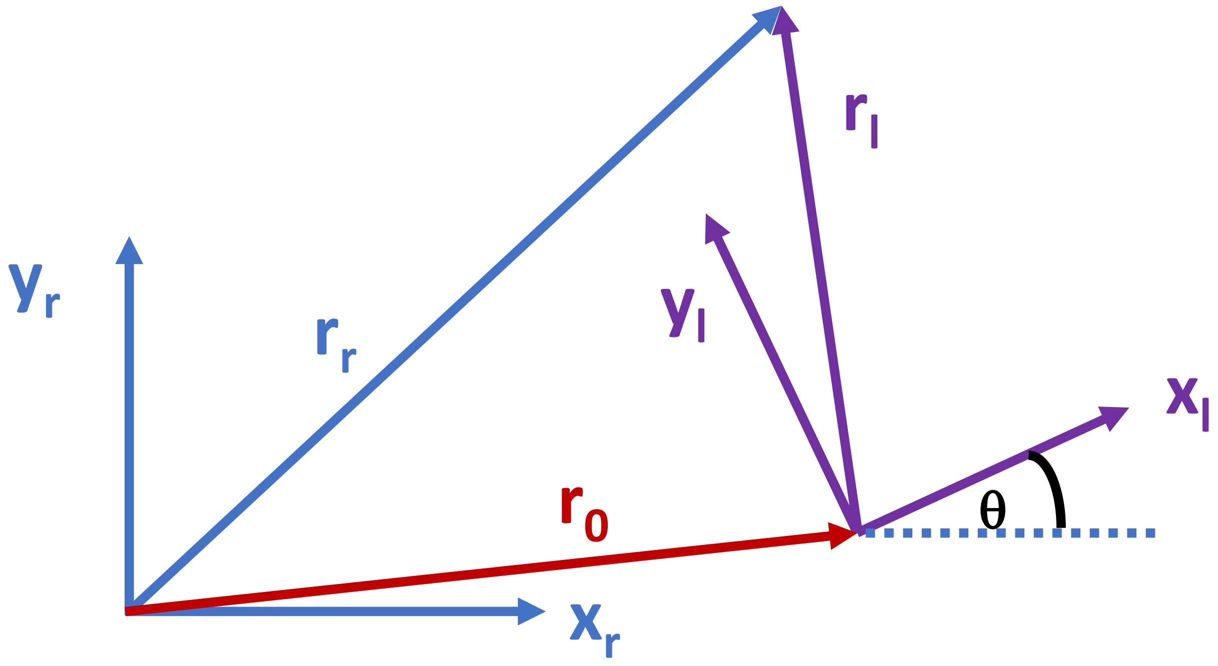
# Right & Left Coordinate Systems




$$R = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathrm{R}\,\boldsymbol{r}_l + \boldsymbol{r}_o = \boldsymbol{r}_r$$

# Which Equation Describes the 2D Alignment?




$$R = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathrm{R}\,\boldsymbol{r}_l + \boldsymbol{r}_o = \boldsymbol{r}_r$$

# How Many Unknowns, How Many Equations?

Unknowns: Rotation angle $\theta$ and translation vector $r_0 = (x_o, y_o)^T$

Number of unknowns: 3

We need at least 3 equations and 3 landmarks

R = rotation matrix, $r_l$ = left landmark, $r_r$ = right landmark

$$\mathrm{R}\,\boldsymbol{r}_l + \boldsymbol{r}_o = \boldsymbol{r}_r$$

# 2D Absolute Orientation

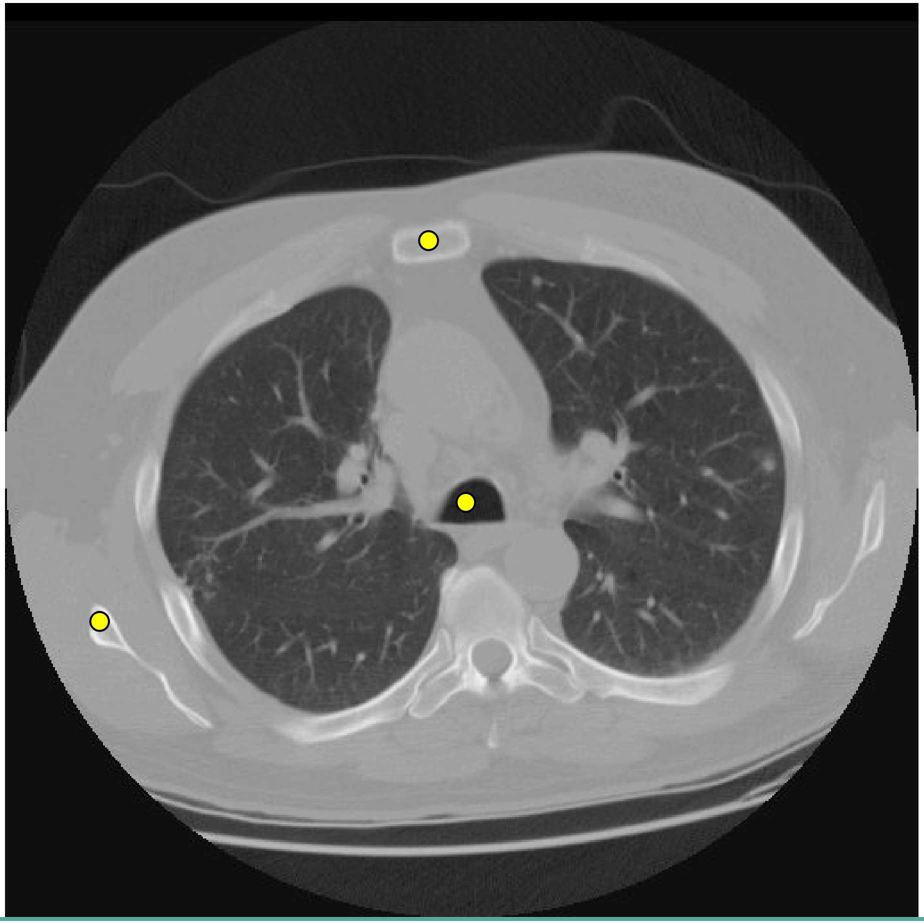
Unknowns: Rotation angle $\theta$ and translation vector $(x_o, y_o)^T$

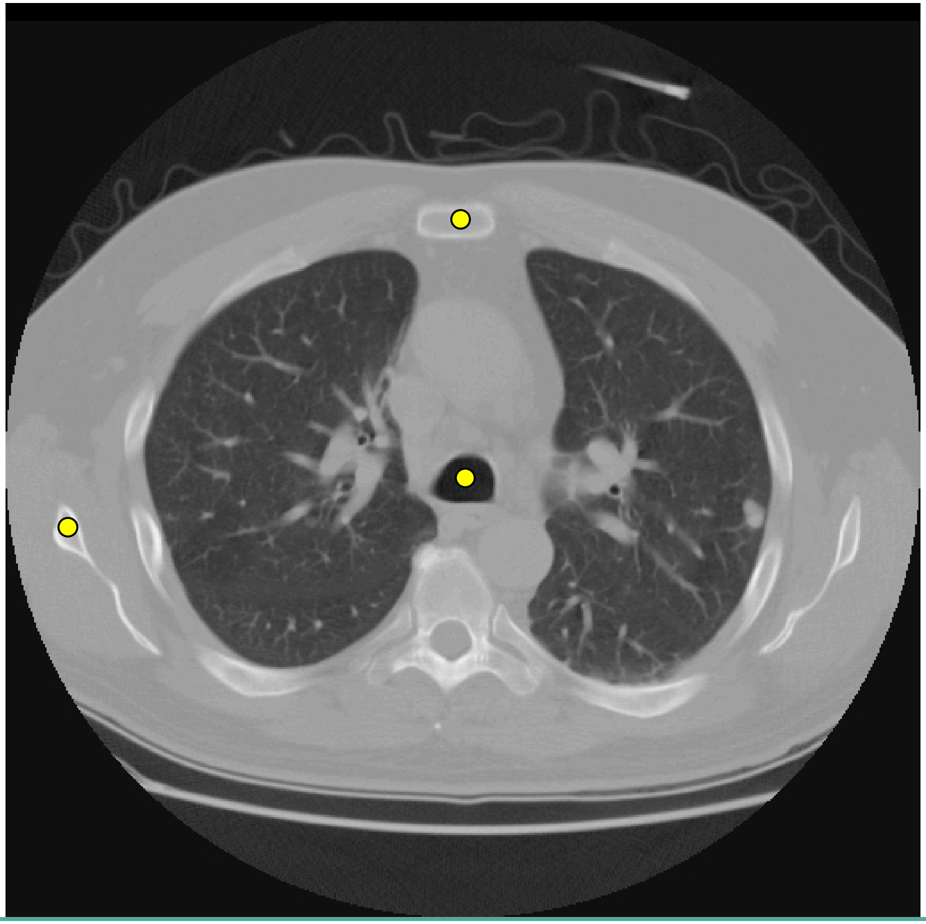
Number of unknowns = 3

Equation not linear: $$\mathrm{R}\,\boldsymbol{r}_l + \boldsymbol{r}_o = \boldsymbol{r}_r$$ $$\mathrm{R} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

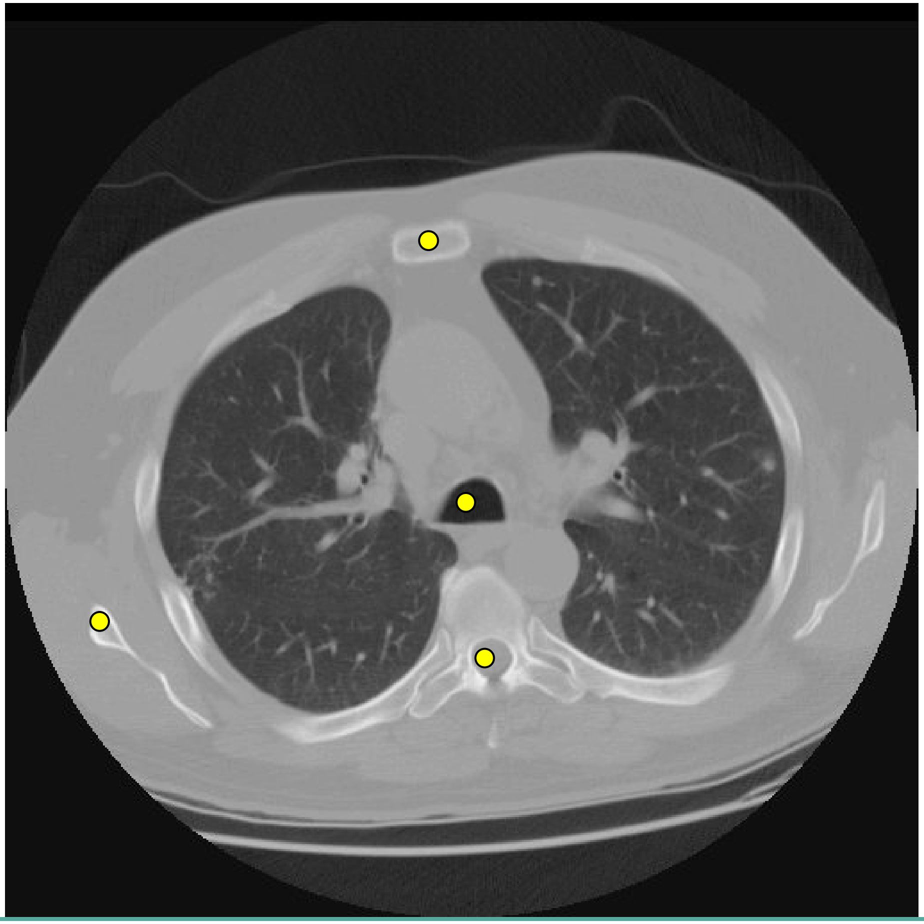
1 equation per landmark pair $\boldsymbol{r}_{l,i}, \boldsymbol{r}_{r,i}$ for i = 1, 2, and 3 ?

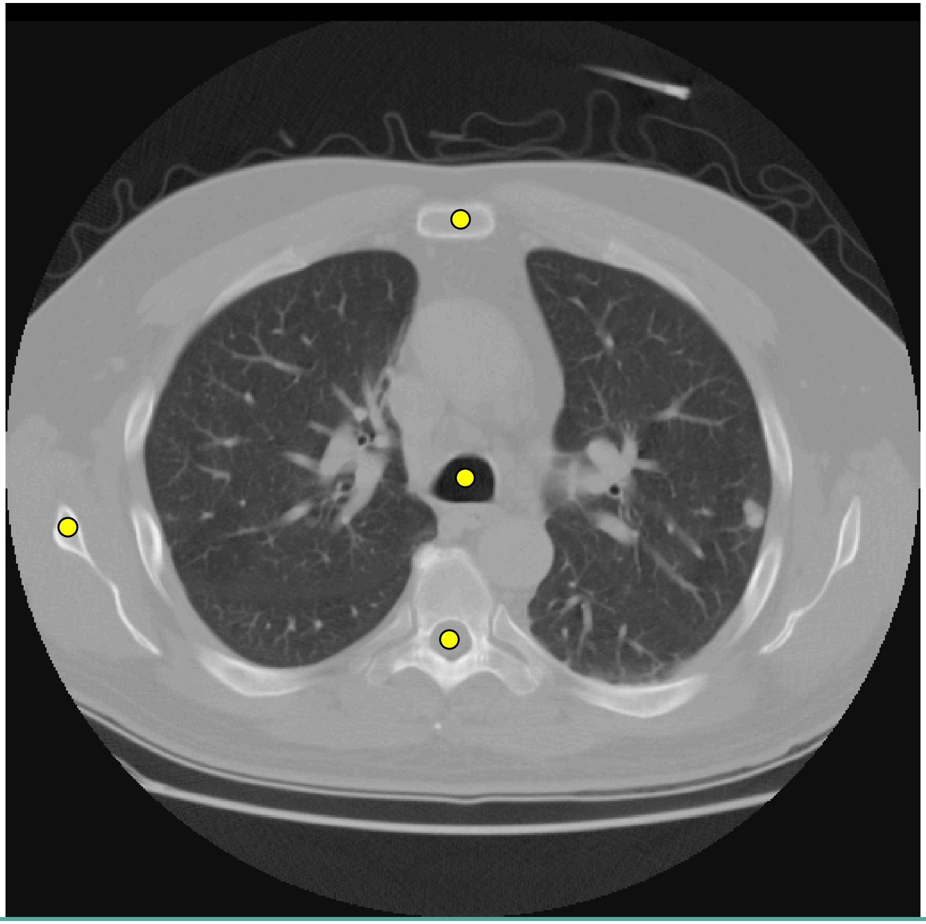
# Three landmarks to be found in both scans:





# Use > 3 Landmarks, e.g.:





# 2D Absolute Orientation

Unknowns: Rotation angle $\theta$ and translation vector $(x_o, y_o)^T$

We need at least 3 equations and 3 landmarks

$$\mathrm{R}\,\boldsymbol{r}_l + \boldsymbol{r}_o = \boldsymbol{r}_r$$

Better: Use n equations

1 equation per landmark pair $\boldsymbol{r}_{l,i}, \boldsymbol{r}_{r,i}$ for i = 1, …, n

*How do you combine n versions of the equation above?*

# Least Squares Method (LSM)

Unknowns: Rotation angle $\theta$ and translation vector $(x_o, y_o)^T$

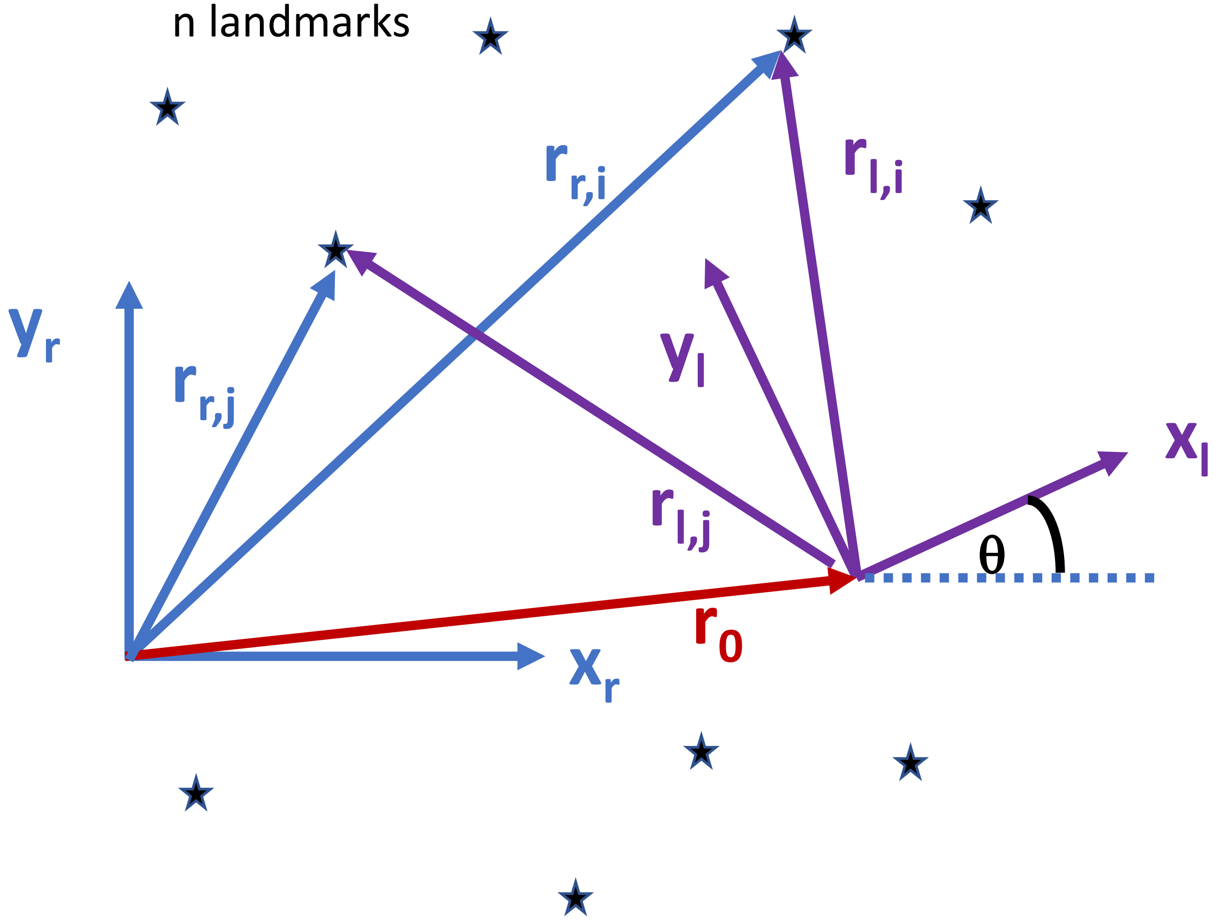
Algorithm to determine unknowns may yield errors in equations:

$$r_{r,i} - R\, r_{l,i} - r_0 = ERROR$$

Goal of LSM: Minimize the sum of the squared errors:

$$\min_{x_o,y_o,\theta} \sum_{i=1}^{N} \|\boldsymbol{r}_{r,i} - R\,\boldsymbol{r}_{l,i} - \boldsymbol{r}_o\|^2$$

# Right (= Scan 1) & Left (= Scan 2) Landmarks




*n* landmark pairs:

$$\boldsymbol{r}_{l,i}, \boldsymbol{r}_{r,i}$$

*for i=1,.., n*

# Least Squares Method (LSM)

Minimize the error:

$$\min_{x_o, y_o, \theta} \sum_{i=1}^{N} \|\boldsymbol{r}_{r,i} - R\,\boldsymbol{r}_{l,i} - \boldsymbol{r}_o\|^2$$

# Least Squares Method (LSM)

Minimize the error:

$$\min_{x_o,y_o,\theta} \sum_{i=1}^{N} \|\boldsymbol{r}_{r,i} - R\,\boldsymbol{r}_{l,i} - \boldsymbol{r}_o\|^2$$

which is equivalent to:

$$\min_{x_o,y_o,\theta} \sum_{i=1}^{N} \left\| \begin{pmatrix} x_{r,i} \\ y_{r,i} \end{pmatrix} - \left( \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_{l,i} \\ y_{l,i} \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \right) \right\|^2$$

# Least Squares Method (LSM)

Minimize the error:

$$\min_{x_o, y_o, \theta} \sum_{i=1}^{N} \left\| \boldsymbol{r}_{r,i} - R\, \boldsymbol{r}_{l,i} - \boldsymbol{r}_o \right\|^2$$

which is equivalent to:

$$\min_{x_o, y_o, \theta} \sum_{i=1}^{N} \left\| \begin{pmatrix} x_{r,i} \\ y_{r,i} \end{pmatrix} - \left( \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_{l,i} \\ y_{l,i} \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \right) \right\|^2$$

or

$$\min_{x_0, y_0, \theta} \sum_{i=1}^{N} \left( (x_{r,i} - (x_{l,i}\cos\theta - y_{l,i}\sin\theta + x_0))^2 + (y_{r,i} - (x_{l,i}\sin\theta + y_{l,i}\cos\theta + y_0))^2 \right)$$

# Trick to reduce number of unknowns:

- Compute centroids of landmarks in each image $(x_r, y_r)^T$ and $(x_l, y_l)^T$

  Left centroid $\bar{\boldsymbol{r}}_l = \sum_{i=1}^{n} \boldsymbol{r}_{l,i}$ and right centroid: $\bar{\boldsymbol{r}}_r = \sum_{i=1}^{n} \boldsymbol{r}_{r,i}$

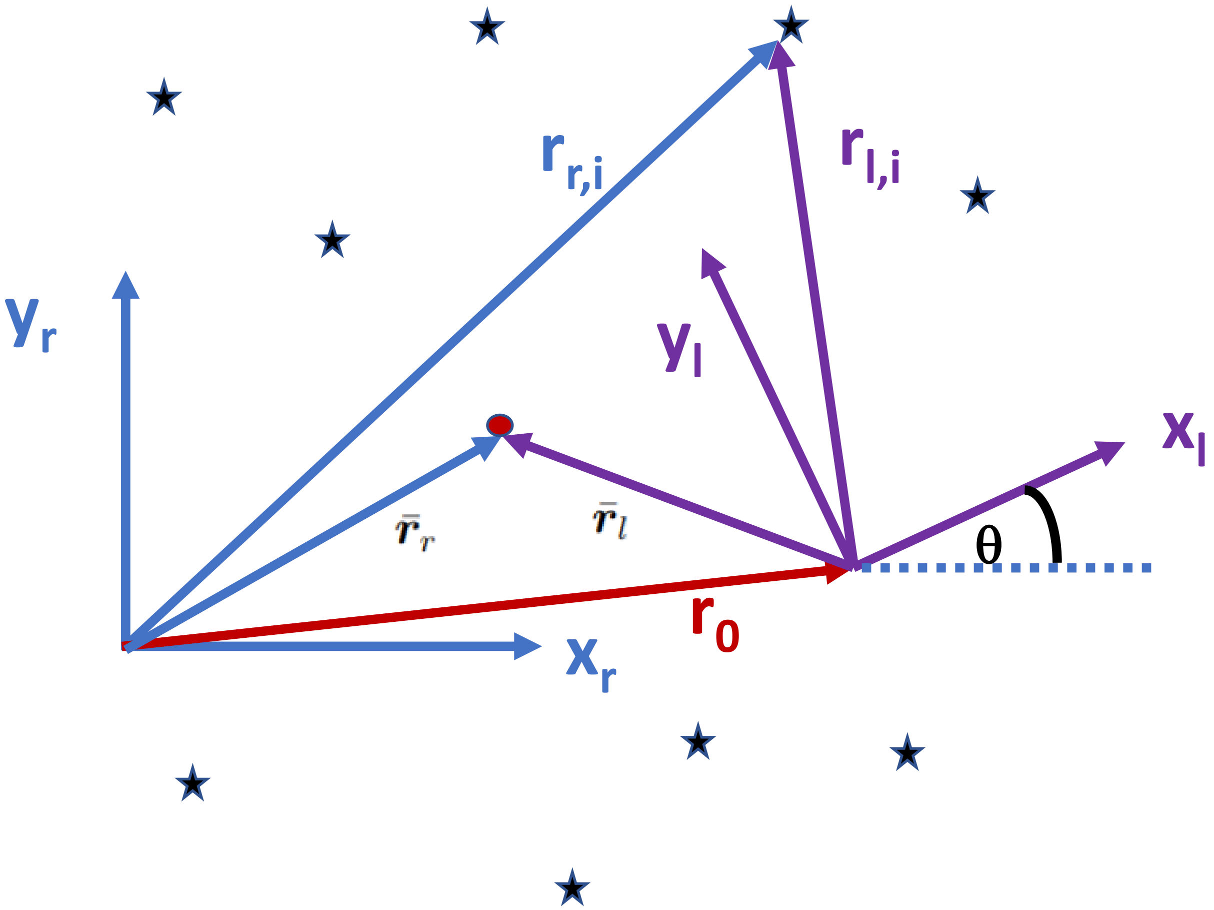
- Assume: $\bar{\boldsymbol{r}}_r = (\bar{x}_r, \bar{y}_r)^T \xleftrightarrow{\text{No Error!}} \bar{\boldsymbol{r}}_l = (\bar{x}_l, \bar{y}_l)^T$

- That means: $$\bar{\boldsymbol{r}}_r - R\,\bar{\boldsymbol{r}}_l - \boldsymbol{r}_0 = 0$$
- Or:

$$\bar{x}_r - (\bar{x}_l \cos\theta - \bar{y}_l \sin\theta + x_0) = 0$$

$$\bar{y}_r - (\bar{x}_l \sin\theta + \bar{y}_l \cos\theta + y_0) = 0$$

# Centroids = Origin of New Coordinate Systems

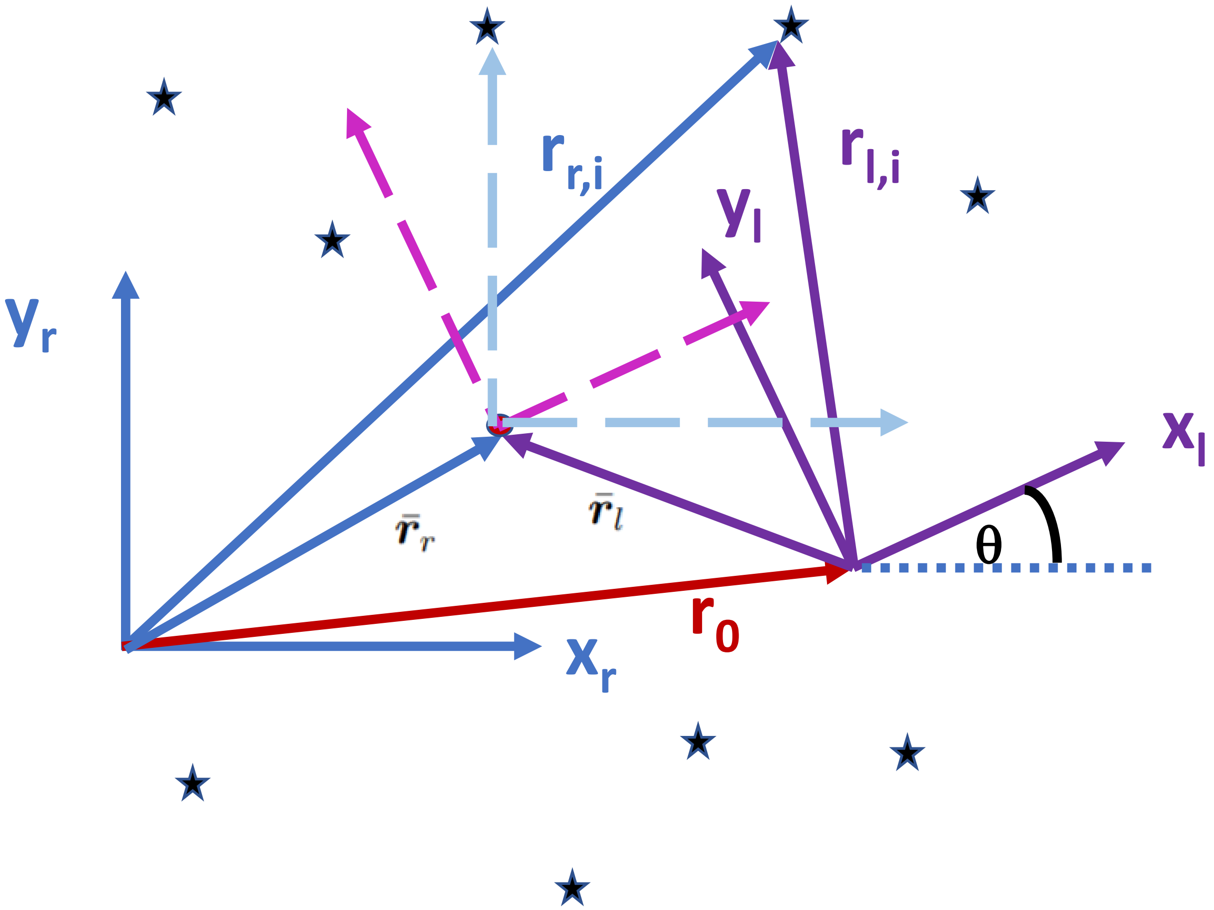



$$\bar{\boldsymbol{r}}_r = \sum_{i=1}^{n} \boldsymbol{r}_{r,i}$$

$$\bar{\boldsymbol{r}}_l = \sum_{i=1}^{n} \boldsymbol{r}_{l,i}$$

★2D Landmark

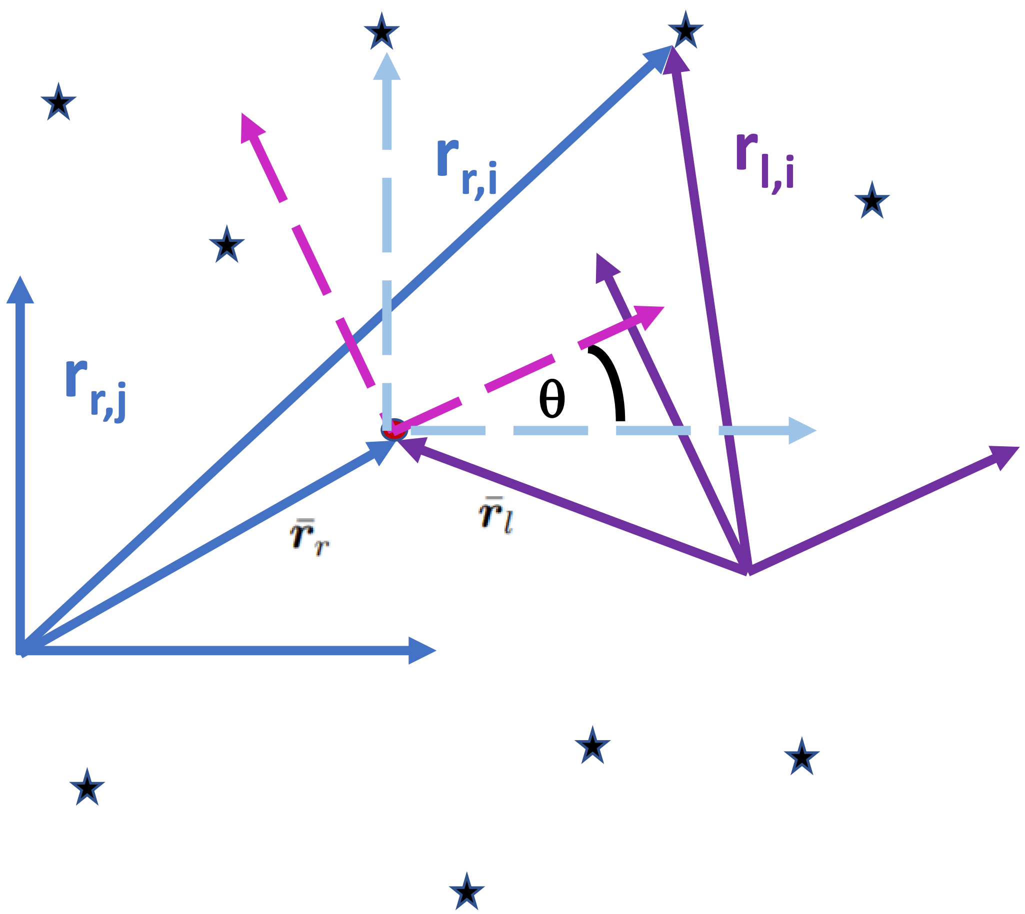
# Centroids = Origin of New Coordinate Systems



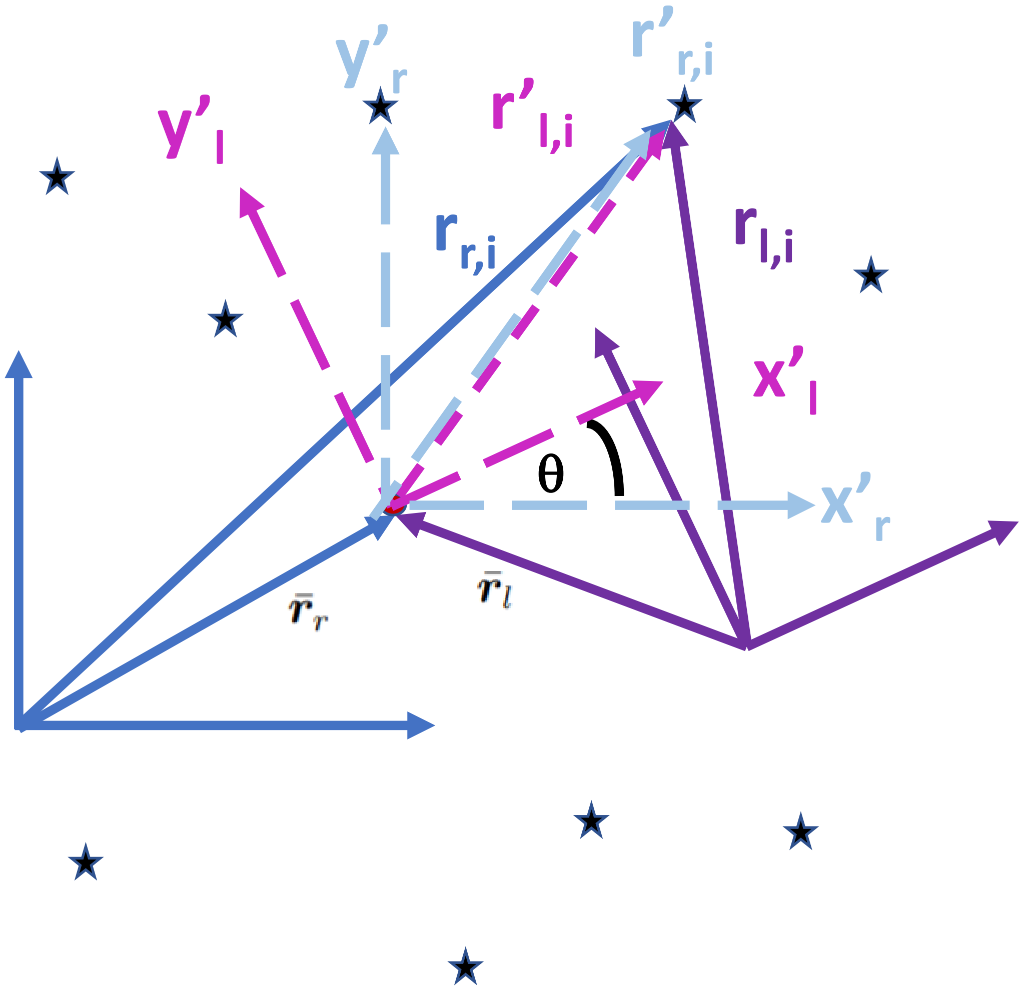

$$\bar{\boldsymbol{r}}_r = \sum_{i=1}^{n} \boldsymbol{r}_{r,i}$$

$$\bar{\boldsymbol{r}}_l = \sum_{i=1}^{n} \boldsymbol{r}_{l,i}$$

# Centroids = Origin of New Coordinate Systems



# Reaching Landmarks in New Coordinate Systems



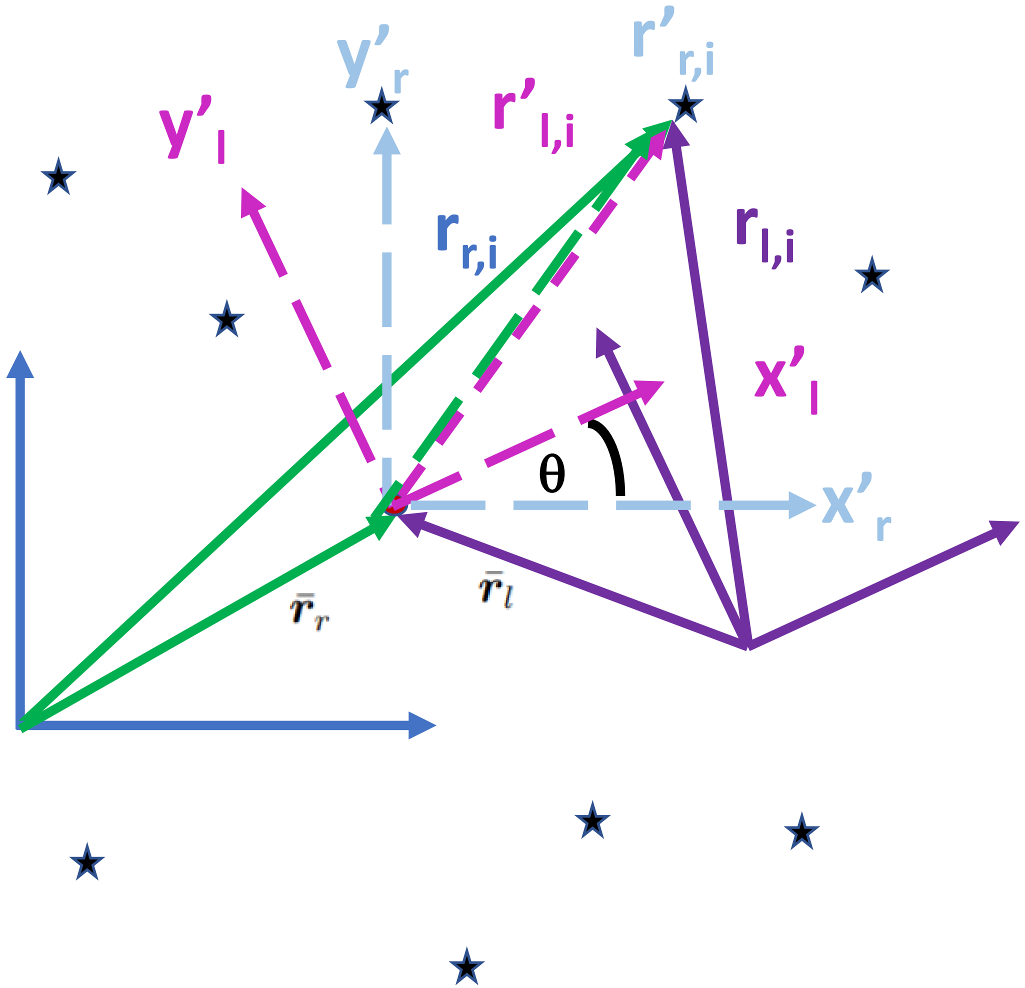

$$x_l = \bar{x}_l + x'_l$$

$$y_l = \bar{y}_l + y'_l$$

$$x_r = \bar{x}_r + x'_r$$

$$y_r = \bar{y}_r + y'_r$$

# Reaching Landmarks in New Coordinate Systems

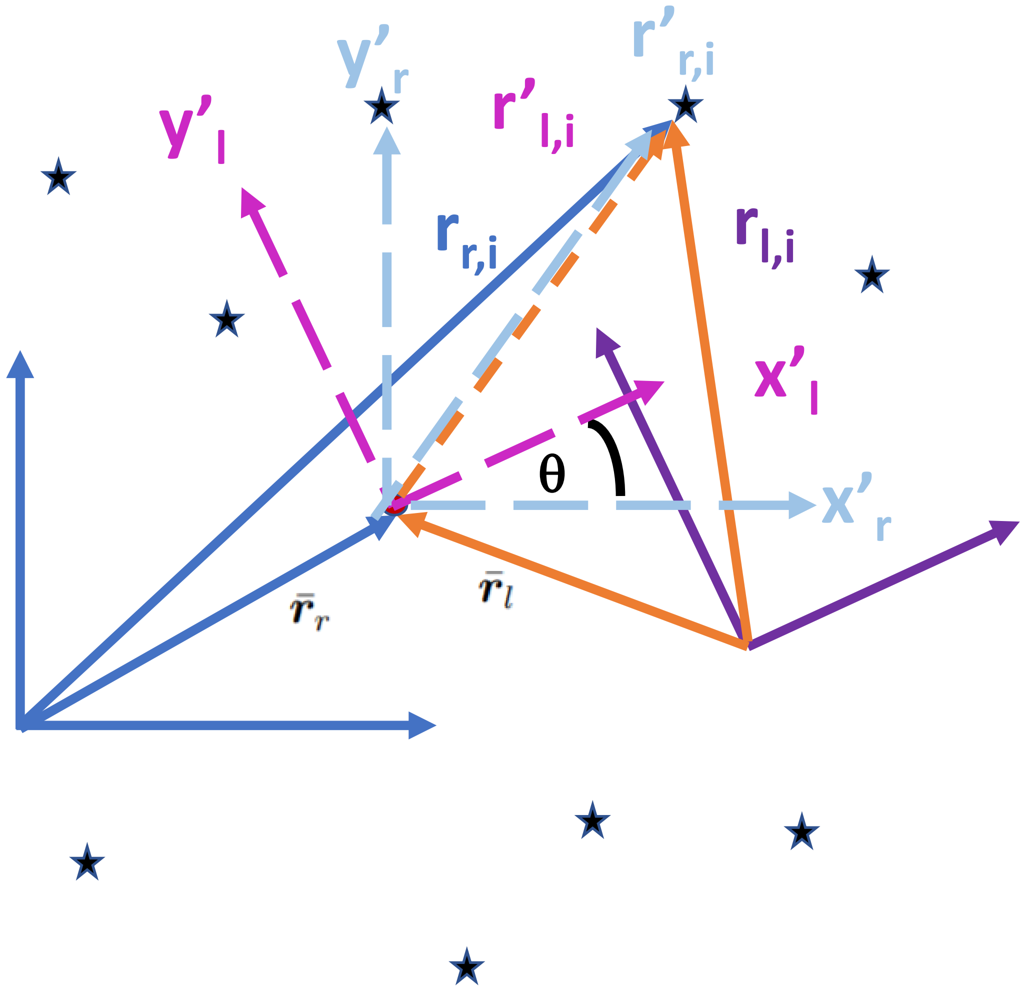



$$x_l = \bar{x}_l + x'_l$$

$$y_l = \bar{y}_l + y'_l$$

$$\boxed{\begin{aligned} x_r &= \bar{x}_r + x'_r \\ y_r &= \bar{y}_r + y'_r \end{aligned}}$$

# Reaching Landmarks in New Coordinate Systems




$$\boxed{\begin{aligned} x_l &= \bar{x}_l + x'_l \\ y_l &= \bar{y}_l + y'_l \end{aligned}}$$

$$x_r = \bar{x}_r + x'_r$$

$$y_r = \bar{y}_r + y'_r$$

# Convert Original LSM into Coordinate-Transformed LSM:

$$\min_{x_0,y_0,\theta} \sum_{i=1}^{N}((x_{r,i} - (x_{l,i}\cos\theta - y_{l,i}\sin\theta + x_0))^2 + (y_{r,i} - (x_{l,i}\sin\theta + y_{l,i}\cos\theta + y_0))^2)$$

$$\min_{\theta} \sum_{i=1}^{N}((x'_{r,i} - (x'_{l,i}\cos\theta - y'_{l,i}\sin\theta))^2 + (y'_{r,i} - (x'_{l,i}\sin\theta + y'_{l,i}\cos\theta))^2$$

# Advantage? No translation vector $(x_o, y_o)^T$
# Only one unknown variable $\theta$

$$\min_{x_0,y_0,\theta} \sum_{i=1}^{N}((x_{r,i} - (x_{l,i}\cos\theta - y_{l,i}\sin\theta + x_0))^2 + (y_{r,i} - (x_{l,i}\sin\theta + y_{l,i}\cos\theta + y_0))^2)$$

$$\min_{\theta} \sum_{i=1}^{N}((x'_{r,i} - (x'_{l,i}\cos\theta - y'_{l,i}\sin\theta))^2 + (y'_{r,i} - (x'_{l,i}\sin\theta + y'_{l,i}\cos\theta))^2$$

# Solve Transformed LSM Problem:

- Take derivative with respect to $\theta$.
- Set result equal to zero.
- The terms with factors $\sin\theta\cos\theta$, $\sin^2\theta$, and $\cos^2\theta$ cancel each other.
- The only remaining terms include $\sin\theta$ and $\cos\theta$.
- Collect the terms that include $\sin\theta$ on one side of the equation and $\cos\theta$ on the other.
- Divide by $\cos\theta$ to yield an expression with $\tan\theta$:
- Equation 1:

$$\tan\theta = \frac{\sum_{i=1}^{n}(y'_{r,i}x'_{l,i} - x'_{r,i}y'_{l,i})}{\sum_{i=1}^{n}(x'_{r,i}x'_{l,i} + y'_{r,i}y'_{l,i})}$$

Equation 2:

$$\boldsymbol{r}_0 = \bar{\boldsymbol{r}}_r - R\,\bar{\boldsymbol{r}}_l$$

# Algorithm for Image Alignment in 2D

Input: Landmarks in two images

1. Determine corresponding point pairs.
2. Compute centroids of landmarks in each image $(x_r, y_r)^T$ and $(x_l, y_l)^T$
3. Transform point coordinates into coordinate systems with centroids serving as the origins.
4. Compute $\theta$ with Equation 1 (above)
5. Compute $(x_o, y_o)^T$ with Equation 2 (above)

Output: Rotation angle $\theta$ and translation vector $(x_o, y_o)^T$

# Absolute Orientation in 3D

Definition:

Compute the parameters that describe a 3D rigid body alignment

= Determine 3 rotation angles and a 3D translation vector or

= Determine a 3x3 rotation matrix R and a 3D translation vector $r_o$ or

$$\mathrm{R}\,\boldsymbol{r}_l + \boldsymbol{r}_o = \boldsymbol{r}_r$$

= Determine a unit quaternion $\overset{o}{q}$ and a 3D translation vector

# What you need to know about Quaternions:

- Vector with 4 components:  1 scalar, 3 "imaginary" parts
- How multiplications of i, j, and k work
- How multiplications of quaternions work (not commutative)
- How represent a quaternion multiplication by a 4x4 orthogonal skew-symmetric matrix
- What is the magnitude of a quaternion?  The conjugate?
- The dot product of two quaternions is the regular 4D dot product.
- What is a unit quaternion?  Why does it represent a 3D rotation?
- Rotation preserves length of 3D vectors and angles between them
- Composition of rotations = multiplications of quaternions
- How to compute a rotation matrix from a quaternion

# Algorithm for Absolution Orientation in 3D

Input: Landmarks in two 3D scans

1. Determine corresponding point pairs.
2. Compute centroids of landmarks in each image $(x_r, y_r, z_r)^T$ and $(x_l, y_l, z_l)^T$
3. Transform point coordinates into coordinate systems with centroids serving as the origins.
4. Compute quaternion & rotation matrix
5. Compute translation

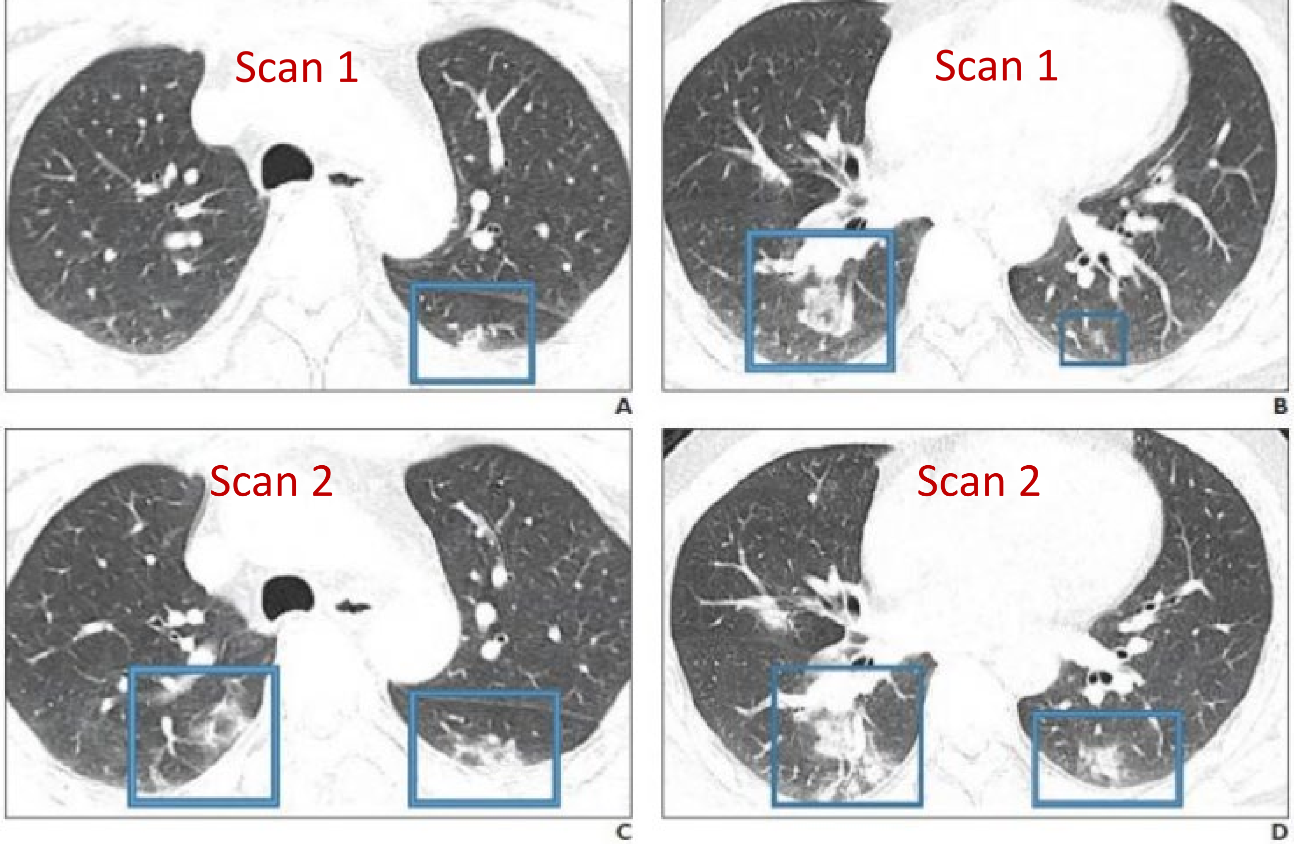
Output: Rotation matrix R and translation vector $(x_o, y_o, z_o)^T$

# COVID-19 CT Scan Evaluation Process:

Given a pair of CT scans:
Compute 3D Alignment

Semi-automatic Evaluation:
1. Radiologist annotates regions of interest with consolidation in scan 2. AI system finds all voxels with consolidation in scan 2
OR:
Automatic Evaluation:
AI system detects & measures the volume of the consolidation in scan 1 and scan 2

AI system evaluates change in volume of region: Increase: Patient fares worse
Decrease: Patient is getting better




CAPTION
A and B, Initial CT images obtained show small round areas of mixed ground-glass opacity and consolidation (rectangles) at level of aortic arch (A) and ventricles (B) in right and left lower lobe posterior zones. C and D, Follow-up CT images obtained 2 days later show progression of abnormalities (rectangles) at level of aortic arch (C) and ventricles (D), which now involve right upper and right and left lower lobe posterior zones.

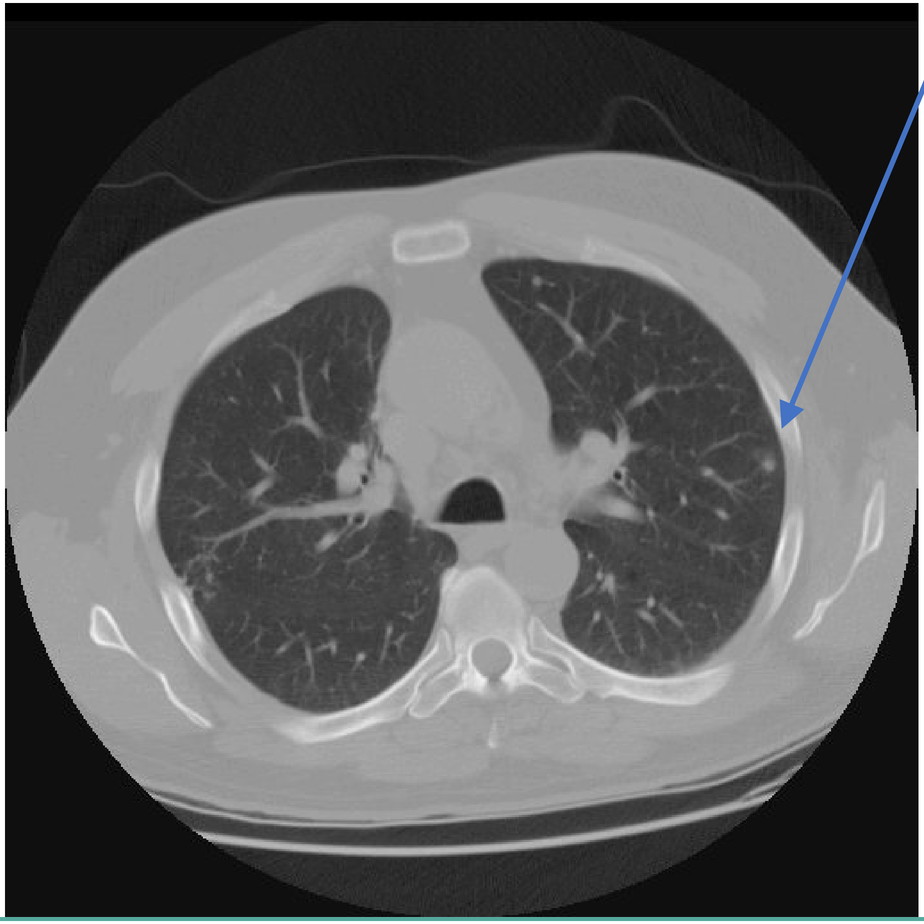
CREDIT
American Journal of Roentgenology (AJR)

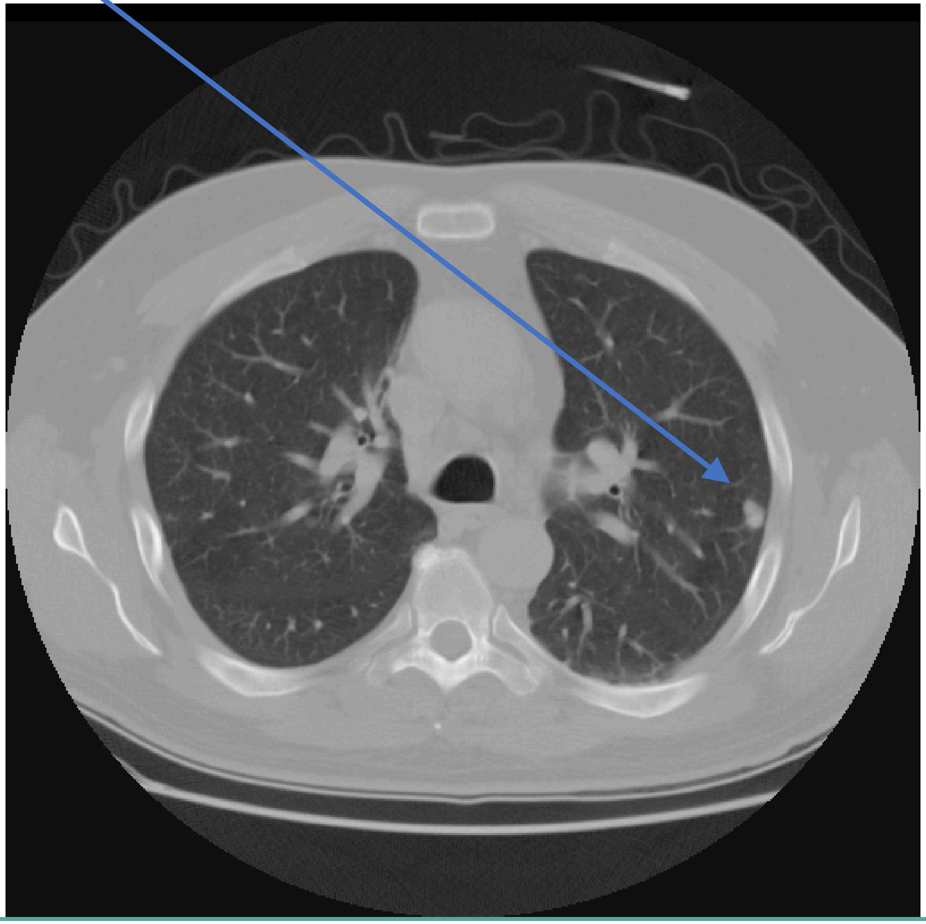
"Devil's Advocate:" Such an AI system may not matter – the radiologist can glance at the two CT scans and diagnose the patient's status.

Arguments for AI system:

- AI could quantify the (partial) filling of air spaces more accurately and timely
- AI may be able to predict patient outcome from scan 1 alone, helping to assess COVID severity and guide treatment
- AI may be able to combine the interpretation of CT scans with other features (patient's age, preexisting conditions, symptoms, etc)
- AI system could also be applied to cancer quantification and change in cancer quantification

# Quantification of Growth of Cancer Nodules





# What are Quaternions?

**Complex numbers versus quaternions**

To define complex numbers:

2D vector space over reals:

Elements have the form $x + iy$,

where $x$ is scalar and $y$ is imaginary part

One more axiom required: $i^2 = -1$.

To define quaternions:

4D vector space over reals:

Elements have the form $q_0 + q_1 i + q_2 j + q_3 k$,

where $q_0$ is scalar and $q_1, q_2, q_3$ are imaginary parts

Six more axioms:

$$i^2 = j^2 = k^2 = -1$$

$$ij = k$$

$$jk = i$$

$$ki = j$$

# Quaternion Properties

We can write a quaternion several ways:

$$q = q_0 + q_1 i + q_2 j + q_3 k$$

$$q = (q_0, q_1, q_2, q_3)$$

$$q = q_0 + \boldsymbol{q}$$

where $q_0$ is the *scalar part* and $\boldsymbol{q}$ is the *vector part*

# Quaternion Properties

We can write a quaternion product several ways:

$$
\begin{aligned}
pq &= (p_0 + p_1 i + p_2 j + p_3 k)(q_0 + q_1 i + q_2 j + q_3 k) \\
&= (p_0 q_0 - p_1 q_1 - p_2 q_2 - p_3 q_3) + \ldots i + \ldots j + \ldots k \\
pq &= (p_0 + \boldsymbol{p})(q_0 + \boldsymbol{q}) \\
&= (p_0 q_0 + p_0 \boldsymbol{q} + q_0 \boldsymbol{p} + \boldsymbol{pq})
\end{aligned}
$$

So what is $\boldsymbol{pq}$? Cross product? Dot product?

Both! Cross product minus dot product!

$$pq = (p_0 q_0 - \boldsymbol{p} \cdot \boldsymbol{q} + p_0 \boldsymbol{q} + q_0 \boldsymbol{p} + \boldsymbol{p} \times \boldsymbol{q})$$

# Quaternion Properties

Quaternion conjugate:

$$q^* = q_0 - q_1 i - q_2 j - q_3 k$$

Note that

$$\begin{aligned} qq^* &= (q_0 + \boldsymbol{q})(q_0 - \boldsymbol{q}) \\ &= q_0^2 + q_0\boldsymbol{q} - q_0\boldsymbol{q} - \boldsymbol{q}\boldsymbol{q} \\ &= q_0^2 + \boldsymbol{q} \cdot \boldsymbol{q} - \boldsymbol{q} \times \boldsymbol{q} \\ &= q_0^2 + q_1^2 + q_2^2 + q_3^2 \end{aligned}$$

Quaternion length:

$$|q| = \sqrt{qq^*} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$$

How do we compute rotation matrix R from quaternion q?

A rotation specified by a quaternion q transforms a 3D vector $\boldsymbol{x}$ into a 3D vector $\boldsymbol{x}_{rot}$ according to:

$$r_{rot} = (0 + \boldsymbol{x}_{rot}) = q(0 + \boldsymbol{x})q^* = qrq^* = (\mathbb{Q}\, r)q^* = \bar{\mathbb{Q}}^T \mathbb{Q}\, r$$

$$\bar{\mathbb{Q}}^T \mathbb{Q} = \begin{pmatrix} q_0 & q_x & q_y & q_z \\ -q_x & q_0 & q_z & -q_y \\ -q_y & -q_z & q_0 & q_x \\ -q_z & q_y & -q_x & q_0 \end{pmatrix} \begin{pmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & q_z & -q_y \\ q_y & -q_z & q_0 & q_x \\ q_z & q_y & -q_x & q_0 \end{pmatrix}$$

$$= \begin{pmatrix} q.q & 0 & 0 & 0 \\ 0 & q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 0 & 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 0 & 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix}$$

$$R = \begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix}$$

BOSTON UNIVERSITY

# What you need to know about Quaternions:

- Vector with 4 components:  1 scalar, 3 “imaginary” parts
- How multiplications of i, j, and k work
- How multiplications of quaternions work (not commutative)
- How represent a quaternion multiplication by a 4x4 orthogonal skew-symmetric matrix
- What is the magnitude of a quaternion?  The conjugate?
- The dot product of two quaternions is the regular 4D dot product.
- What is a unit quaternion?  Why does it represent a 3D rotation?
- Rotation preserves length of 3D vectors and angles between them
- Composition of rotations = multiplications of quaternions
- How to compute a rotation matrix from a quaternion

# Least Squares Method (LSM)

Unknowns: Rotation angle $\theta$ and translation vector $(x_o, y_o)^T$

Algorithm to determine unknowns may yield errors in equations:

$$r_{r,i} - R\, r_{l,i} - r_0 = ERROR$$

Goal of LSM: Minimize the error:

$$\min_{R, \boldsymbol{r}_0} \sum_{i=1}^{n} \|\boldsymbol{r}_{r,i} - R\,\boldsymbol{r}_{l,i} - \boldsymbol{r}_0\|^2$$

# LSM for 3D Absolute Orientation

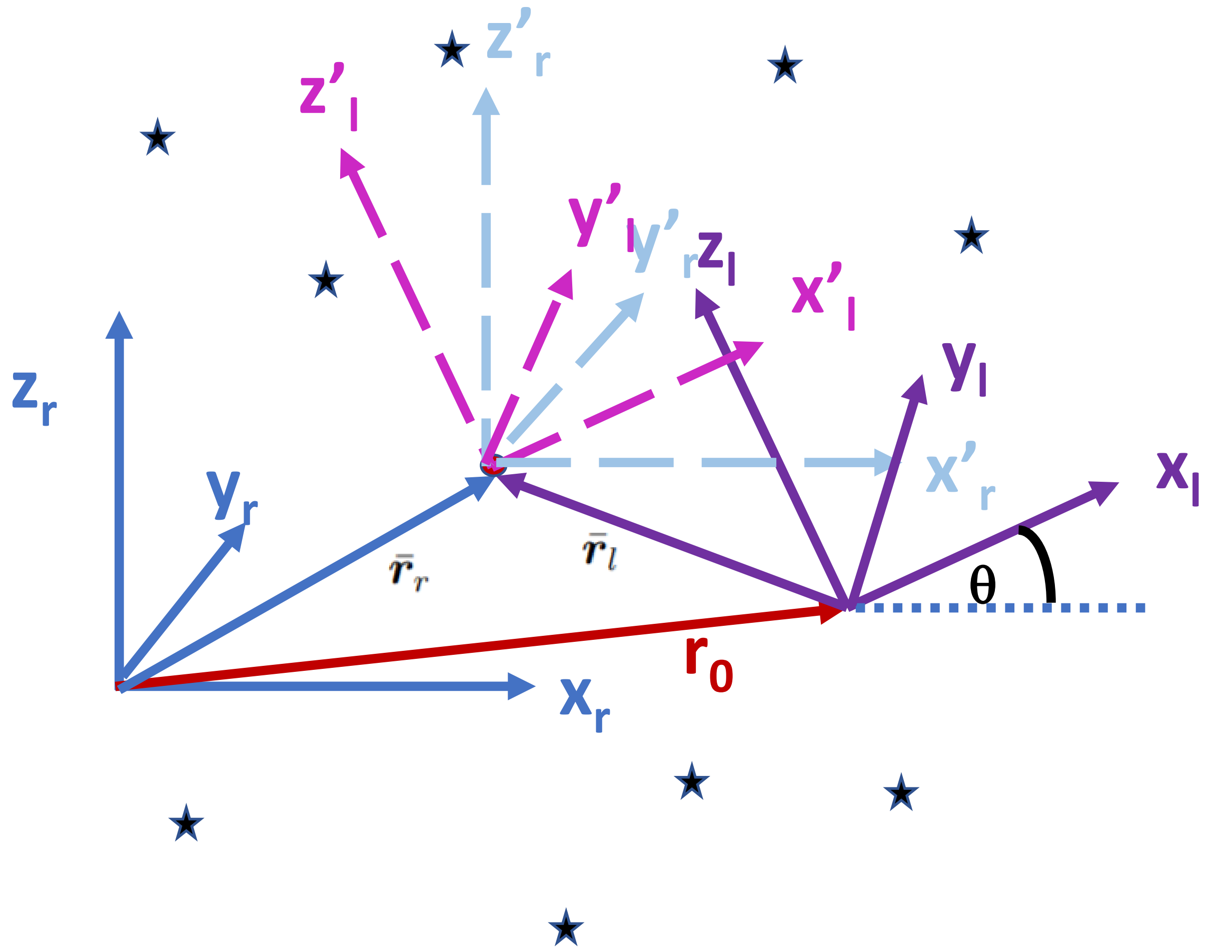
Same trick as in 2D: Use coordinate transform:

From
$$\min_{R,\boldsymbol{r}_0} \sum_{i=1}^{n} \left\| \boldsymbol{r}_{r,i} - R\,\boldsymbol{r}_{l,i} - \boldsymbol{r}_0 \right\|^2$$

To
$$= \min_{R} \sum_{i=1}^{n} \left\| \boldsymbol{r}'_{r,i} - R\,\boldsymbol{r}'_{l,i} \right\|^2$$

Advantage: The only unknown is R. Translation can be solved for later.

# Centroid = Origin of New Coordinate Systems




$$\bar{\boldsymbol{r}}_r = \sum_{i=1}^{n} \boldsymbol{r}_{r,i}$$

$$\bar{\boldsymbol{r}}_l = \sum_{i=1}^{n} \boldsymbol{r}_{l,i}$$

See 2D case for details.

# Simplifying the Minimization Problem:

Convert

$$\min_R \sum_{i=1}^{n} \left\| \boldsymbol{r}'_{r,i} - R\,\boldsymbol{r}'_{l,i} \right\|^2$$

To:

$$= \min_R \left( \sum_{i=1}^{n} \left\| \boldsymbol{r}'_{r,i} \right\|^2 + \sum_{i=1}^{n} \left\| \boldsymbol{r}'_{l,i} \right\|^2 - 2 \sum_{i=1}^{n} \boldsymbol{r}'_{r,i} \cdot R\boldsymbol{r}'_{l,i} \right)$$

since $R^T R = I$ is true for an orthonormal matrix

Then convert min to max:

$$= \max_R \sum_{i=1}^{n} \boldsymbol{r}'_{r,i} \cdot R\boldsymbol{r}'_{l,i}$$

Since the first two sums are fixed.

# Simplifying the Maximization Problem:

Convert from 3D vectors

$$= \max_R \sum_{i=1}^{n} \boldsymbol{r}'_{r,i} \,.\, R\boldsymbol{r}'_{l,i}$$ subject to R orthonormal

to 4D quaternions:

$$= \max_q \sum_{i=1}^{n} r'_{r,i} \,.\, (q\, r'_{l,i}\, q^*)$$ subject to q.q=1

Simplify further:

$$= \max_q \sum_{i=1}^{n} (r'_{r,i} q) \,.\, (q\, r'_{l,i})$$ since (ab).c = a.(cb*)

$$= \max_q \sum_{i=1}^{n} (q\, r'_{l,i}) \,.\, (\, r'_{r,i}\, q)$$ commutative

BOSTON UNIVERSITY

# Simplifying the Maximization Problem Further:

$$= \max_{q} \sum_{i=1}^{n} (\bar{\mathbb{R}}_{l,i}\, q) \,.\, (\mathbb{R}_{r,i}\, q)$$

$$= \max_{q} \sum_{i=1}^{n} (q^T\, \bar{\mathbb{R}}_{r,i}^T\, \mathbb{R}_{l,i}\, q)$$

$$= \max_{q}\; q^T \Big( \sum_{i=1}^{n} \bar{\mathbb{R}}_{r,i}^T\, \mathbb{R}_{l,i} \Big)\, q$$

$$= \max_{q}\; q^T\, N\, q$$

$$N = \begin{pmatrix} s_{xx}+s_{yy}+s_{zz} & s_{yz}-s_{zy} & s_{zx}-s_{xz} & s_{xy}-s_{yx} \\ s_{yz}-s_{zy} & s_{xx}-s_{yy}-s_{zz} & s_{xy}+s_{yx} & s_{zx}-s_{xz} \\ s_{zx}-s_{xz} & s_{xy}+s_{yx} & -s_{xx}+s_{yy}-s_{zz} & s_{yz}-s_{zy} \\ s_{xy}-s_{yx} & s_{zx}+s_{xz} & s_{yz}+s_{zy} & -s_{xx}-s_{yy}+s_{zz} \end{pmatrix}$$

where

$$s_{xx} = \sum x'_{l,i} x'_{r,i},$$

$$s_{xy} = \sum x'_{l,i} y'_{r,i},$$

...

$$\boldsymbol{r}'_{l,i} = (x'_{l,i}, y'_{l,i}, z'_{l,i})$$

$$\boldsymbol{r}'_{r,i} = (x'_{r,i}, y'_{r,i}, z'_{r,i})$$

$$N = \begin{pmatrix} s_{xx}+s_{yy}+s_{zz} & s_{yz}-s_{zy} & s_{zx}-s_{xz} & s_{xy}-s_{yx} \\ s_{yz}-s_{zy} & s_{xx}-s_{yy}-s_{zz} & s_{xy}+s_{yx} & s_{zx}-s_{xz} \\ s_{zx}-s_{xz} & s_{xy}+s_{yx} & -s_{xx}+s_{yy}-s_{zz} & s_{yz}-s_{zy} \\ s_{xy}-s_{yx} & s_{zx}+s_{xz} & s_{yz}+s_{zy} & -s_{xx}-s_{yy}+s_{zz} \end{pmatrix}$$

where

$$s_{xx} = \sum x'_{l,i} x'_{r,i},$$

$$s_{xy} = \sum x'_{l,i} y'_{r,i},$$

...

$$\boldsymbol{r}'_{l,i} = (x'_{l,i}, y'_{l,i}, z'_{l,i})$$

$$\boldsymbol{r}'_{r,i} = (x'_{r,i}, y'_{r,i}, z'_{r,i})$$

Matrix N can be computed from the transformed landmark coordinates

Since $N$ is symmetric, 4 eigenvalues $\lambda_1, \ldots, \lambda_4$ are real, where $Ne_i = \lambda_i e_i$. Eigenvectors are orthogonal.

# Solving the Maximization Problem by Taking Advantage of the Symmetry of Matrix N:

$$q = \alpha_1 e_1 + \ldots + \alpha_4 e_4$$

# Solving the Maximization Problem by Taking Advantage of the Symmetry of Matrix M:

Since $N$ is symmetric, 4 eigenvalues $\lambda_1, \ldots, \lambda_4$ are real, where $Ne_i = \lambda_i e_i$. Eigenvectors are orthogonal.

$$q = \alpha_1 e_1 + \ldots + \alpha_4 e_4$$

$$N\,q = \alpha_1\,Ne_1 + \ldots + \alpha_4\,Ne_4 = \alpha_1\lambda_1 e_1 \ldots + \alpha_4\lambda_4 e_4$$

$$q^T N\,q = \alpha_1^2\lambda_1 + \ldots + \alpha_4^2\lambda_4$$

since $e_i.e_i = 1$ and $e_i.e_j = 0$.

# Solving the Maximization Problem by Taking Advantage of the Symmetry of Matrix M:

Since $N$ is symmetric, 4 eigenvalues $\lambda_1, \ldots, \lambda_4$ are real, where $N e_i = \lambda_i e_i$. Eigenvectors are orthogonal.

$$q = \alpha_1 e_1 + \ldots + \alpha_4 e_4$$

$$N\, q = \alpha_1\, N e_1 + \ldots + \alpha_4\, N e_4 = \alpha_1 \lambda_1 e_1 \ldots + \alpha_4 \lambda_4 e_4$$

$$q^T N\, q = \alpha_1^2 \lambda_1 + \ldots + \alpha_4^2 \lambda_4$$

since $e_i.e_i = 1$ and $e_i.e_j = 0$.

Since $q.q = 1$, we have $\alpha_1^2 + \ldots \alpha_4^2 = 1$.

Say $\lambda_1 \geq \ldots \geq \lambda_4$, which means $\lambda_1$ is largest eigenvalue:

$$q\, N\, q \leq \alpha_1^2 \lambda_1 + \ldots + \alpha_4^2 \lambda_1 = (\alpha_1^2 + \ldots + \alpha_4^2) \lambda_1 = \lambda_1$$

# Solving the Maximization Problem by Taking Advantage of the Symmetry of Matrix M:

Since $N$ is symmetric, 4 eigenvalues $\lambda_1, \ldots, \lambda_4$ are real, where $Ne_i = \lambda_i e_i$. Eigenvectors are orthogonal.

$$q = \alpha_1 e_1 + \ldots + \alpha_4 e_4$$

$$N\, q = \alpha_1\, Ne_1 + \ldots + \alpha_4\, Ne_4 = \alpha_1 \lambda_1 e_1 \ldots + \alpha_4 \lambda_4 e_4$$

$$q^T N\, q = \alpha_1^2 \lambda_1 + \ldots + \alpha_4^2 \lambda_4$$

since $e_i.e_i = 1$ and $e_i.e_j = 0$.

Since $q.q = 1$, we have $\alpha_1^2 + \ldots \alpha_4^2 = 1$.

Say $\lambda_1 \geq \ldots \geq \lambda_4$, which means $\lambda_1$ is largest eigenvalue:

$$q\, N\, q \leq \alpha_1^2 \lambda_1 + \ldots + \alpha_4^2 \lambda_1 = (\alpha_1^2 + \ldots + \alpha_4^2) \lambda_1 = \lambda_1$$

Maximum is attained if $\alpha_1 = 1$ and $\alpha_2 = \alpha_3 = \alpha_4 = 0$ and $q = e_1$.

Unknown quaternion $q$ is the eigenvector $e$ that corresponds to the most positive eigenvalue of $N$

How do we compute rotation matrix R from quaternion q?

A rotation specified by a quaternion q transforms a 3D vector $\boldsymbol{x}$ into a 3D vector $\boldsymbol{x}_{rot}$ according to:

$$r_{rot} = (0 + \boldsymbol{x}_{rot}) = q(0 + \boldsymbol{x})q^* = qrq^* = (\mathbb{Q}\, r)q^* = \bar{\mathbb{Q}}^T \mathbb{Q}\, r$$

$$\bar{\mathbb{Q}}^T \mathbb{Q} = \begin{pmatrix} q_0 & q_x & q_y & q_z \\ -q_x & q_0 & q_z & -q_y \\ -q_y & -q_z & q_0 & q_x \\ -q_z & q_y & -q_x & q_0 \end{pmatrix} \begin{pmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & q_z & -q_y \\ q_y & -q_z & q_0 & q_x \\ q_z & q_y & -q_x & q_0 \end{pmatrix}$$

$$= \begin{pmatrix} q.q & 0 & 0 & 0 \\ 0 & q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 0 & 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 0 & 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}$$

$$R = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}$$

# Algorithm for Absolution Orientation in 3D (Horn 1987)

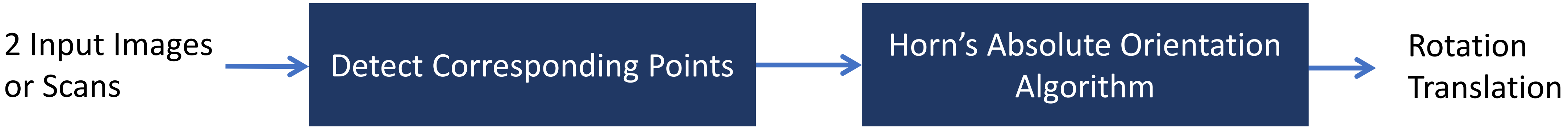
Input: Landmarks in two 3D scans

1. Determine corresponding point pairs.
2. Compute centroids of landmarks in each image $(x_r, y_r, z_r)^T$ and $(x_l, y_l, z_l)^T$
3. Transform point coordinates into coordinate systems with centroids serving as the origins.
4. Compute quaternion & rotation matrix (previous slide)
5. Compute translation: $\boldsymbol{r}_0 = \bar{\boldsymbol{r}}_r - R\,\bar{\boldsymbol{r}}_l$

Output: Rotation matrix R and translation vector $(x_o, y_o, z_o)^T$

# A New, More Difficult Problem:

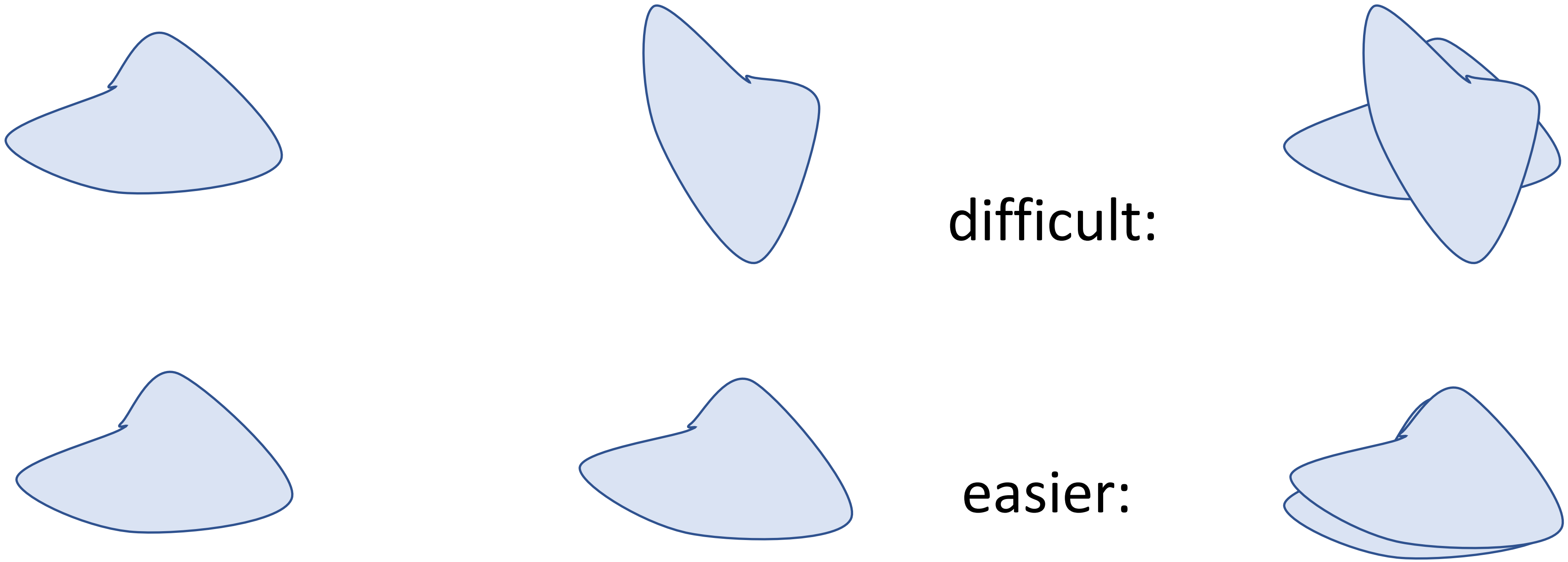
- Horn's Algorithm assumes that corresponding points are given in the two datasets.




- What if you need to find both the correspondence and the rigid body alignment (rotation & translation) together?

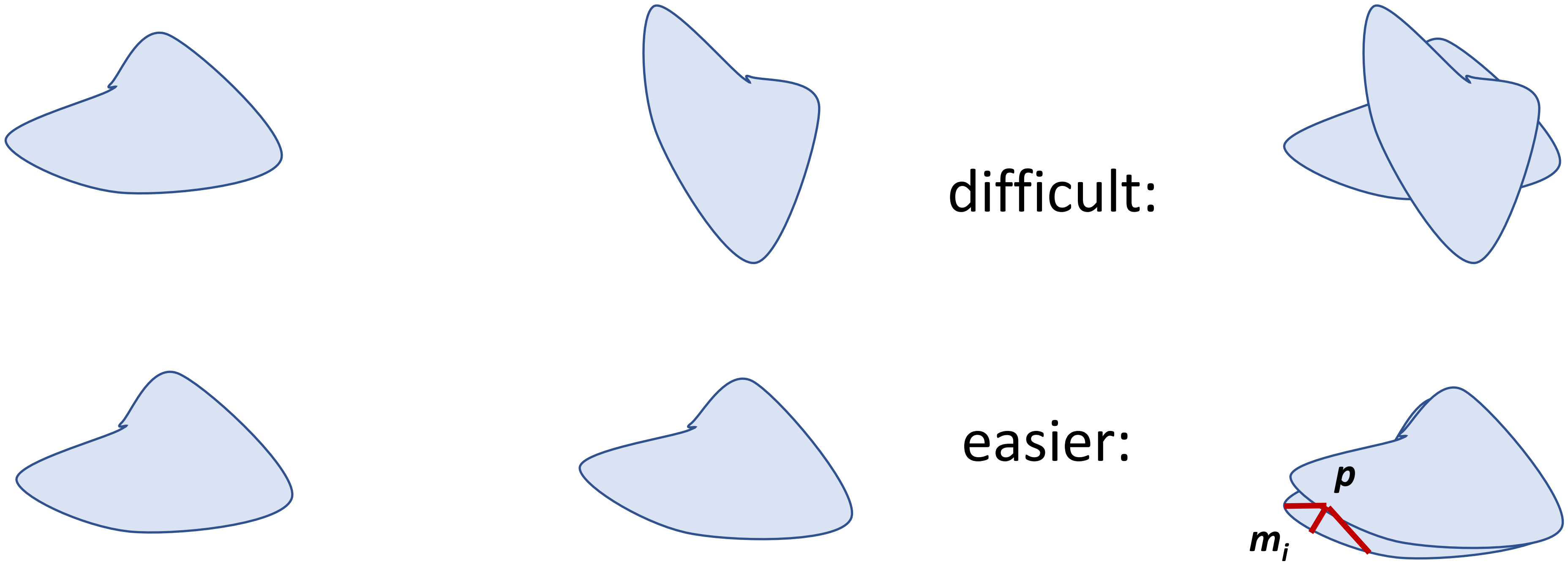
# Finding Corresponding Points on Curves or Surfaces

Curve in Scan 1 | Curve in Scan 2



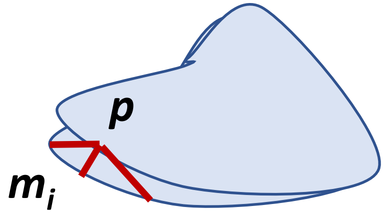
# Finding Corresponding Points on Curves or Surfaces

Curve $C_1$ in Scan 1 — Curve $C_2$ in Scan 2




Compute minimum Euclidean distance $d(\boldsymbol{p}, C_1) = \min_{\boldsymbol{m}_i \in C_2} \|\boldsymbol{p} - \boldsymbol{m}_i\|$

# Correspondence based on shortest Euclidean distance




$$d(\boldsymbol{p}, C_1) = \min_{\boldsymbol{m}_i \in C_2} ||\boldsymbol{p} - \boldsymbol{m}_i||$$

Finding Correspondence and rotation/translation together is a "chicken and egg problem:"

1. If correspondence given: Use Horn'87 to solve for rotation/translation
2. If rotation/translation given: Use shortest Euclidean distance (as in picture above) to obtain correspondence

Solution: Iterative Scheme

# Iterative Closest Point Algorithm: [Besl, McKay 1992]

- Input: Curves $C_1$ and $C_2$

1. Initialize R=I, $r_0$ =0 (no rotation & translation), k=1
2. Compute closest points for $C_1$ and $C_2$
3. Compute absolute orientation and update R and $r_0$ (Use Horn 87)
4. Transform $C_2$ with R and $r_0$
5. Compute squared error in kth iteration
6. If error(k-1) – error(k) is less than a threshold: terminate
7. Otherwise go back to step 2

Betke et al., 2003



Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

ELSEVIER

Medical Image Analysis 0 (2003) 000–000

MEDICAL IMAGE ANALYSIS

www.elsevier.com/locate/media

# Landmark detection in the chest and registration of lung surfaces with an application to nodule registration

Margrit Betke[a,*], Harrison Hong[a], Deborah Thomas[a], Chekema Prince[a], Jane P. Ko[b]

[a] *Computer Science Department, Boston University, Boston, MA 02215, USA*
[b] *Department of Radiology, New York University Medical School, New York, NY 10016, USA*

Received 13 February 2002; received in revised form 24 November 2002; accepted 17 January 2003

**Abstract**

We developed an automated system for registering computed tomography (CT) images of the chest temporally. Our system detects anatomical landmarks, in particular, the trachea, sternum and spine, using an attenuation-based template matching approach. It computes the optimal rigid-body transformation that aligns the corresponding landmarks in two CT scans of the same patient. This transformation then provides an initial registration of the lung surfaces segmented from the two scans. The initial surface alignment is refined step by step in an iterative closest-point (ICP) process. To establish the correspondence of lung surface points, Elias' nearest neighbor algorithm was adopted. Our method improves the processing time of the original ICP algorithm from $O(kn \log n)$ to $O(kn)$, where $k$ is the number of iterations and $n$ the number of surface points. The surface transformation is applied to align nodules in the initial CT scan with nodules in the follow-up scan. For 56 out of 58 nodules in the initial CT scans of 10 patients, nodule correspondences in the follow-up scans are established correctly. Our methods can therefore potentially facilitate the radiologist's evaluation of pulmonary nodules on chest CT for interval growth.

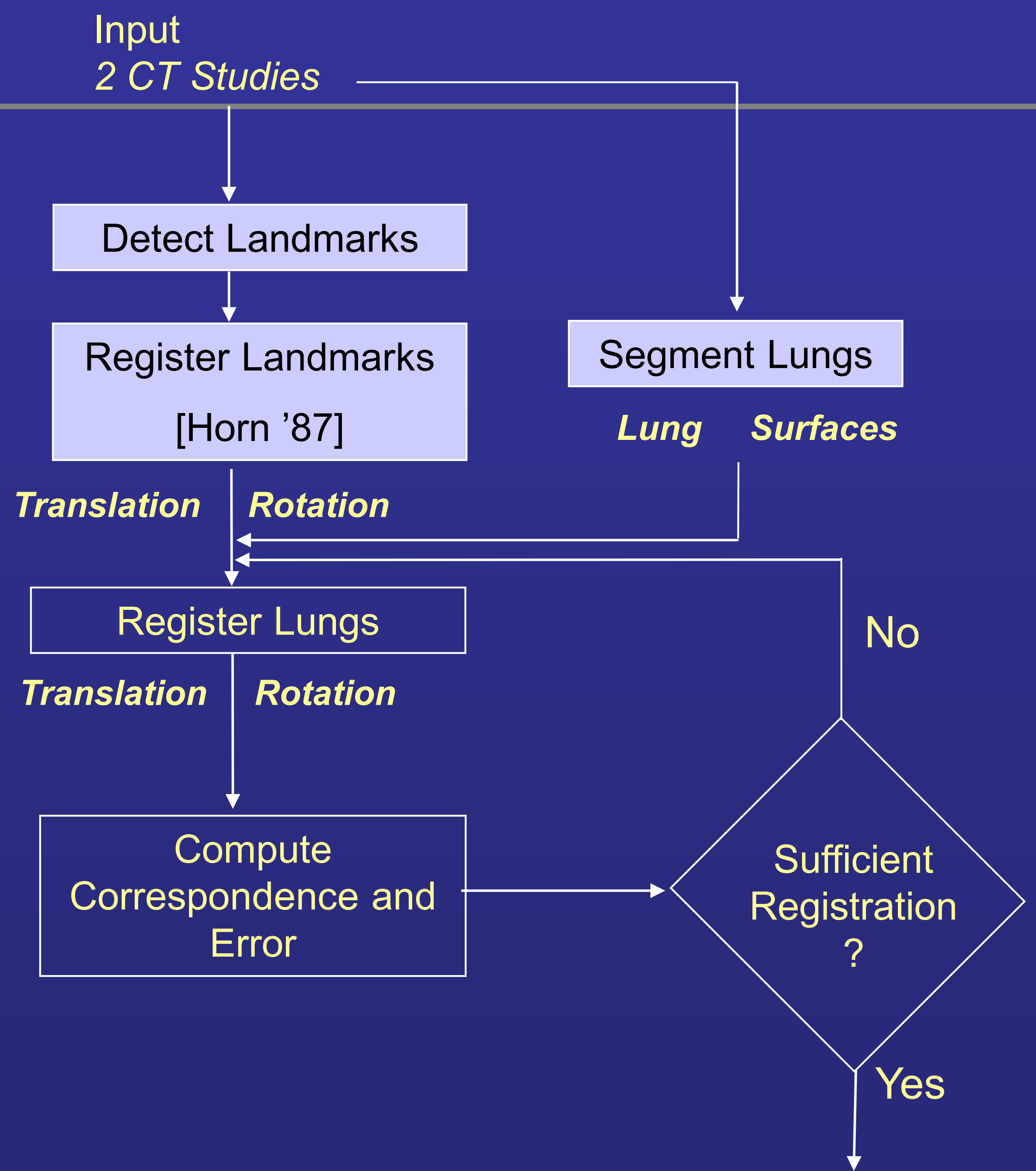

*Keywords:* Computed tomography; Chest; Lung surfaces; Nodule registration

## 1. Introduction

Chest computed tomography (CT) has become a well-established means of diagnosing pulmonary metastases of oncology patients and evaluating response to treatment regimens. Since diagnosis and prognosis of cancer general- include functional lung imaging to evaluate asthma and emphysema and detection of primary lung cancer. Lung cancer remains the leading cause of cancer death in the United States, killing 160,000 people a year. The overall 5-year survival rate is only 15% (Landis et al., 1999), but early detection and resection of pulmonary nodules in

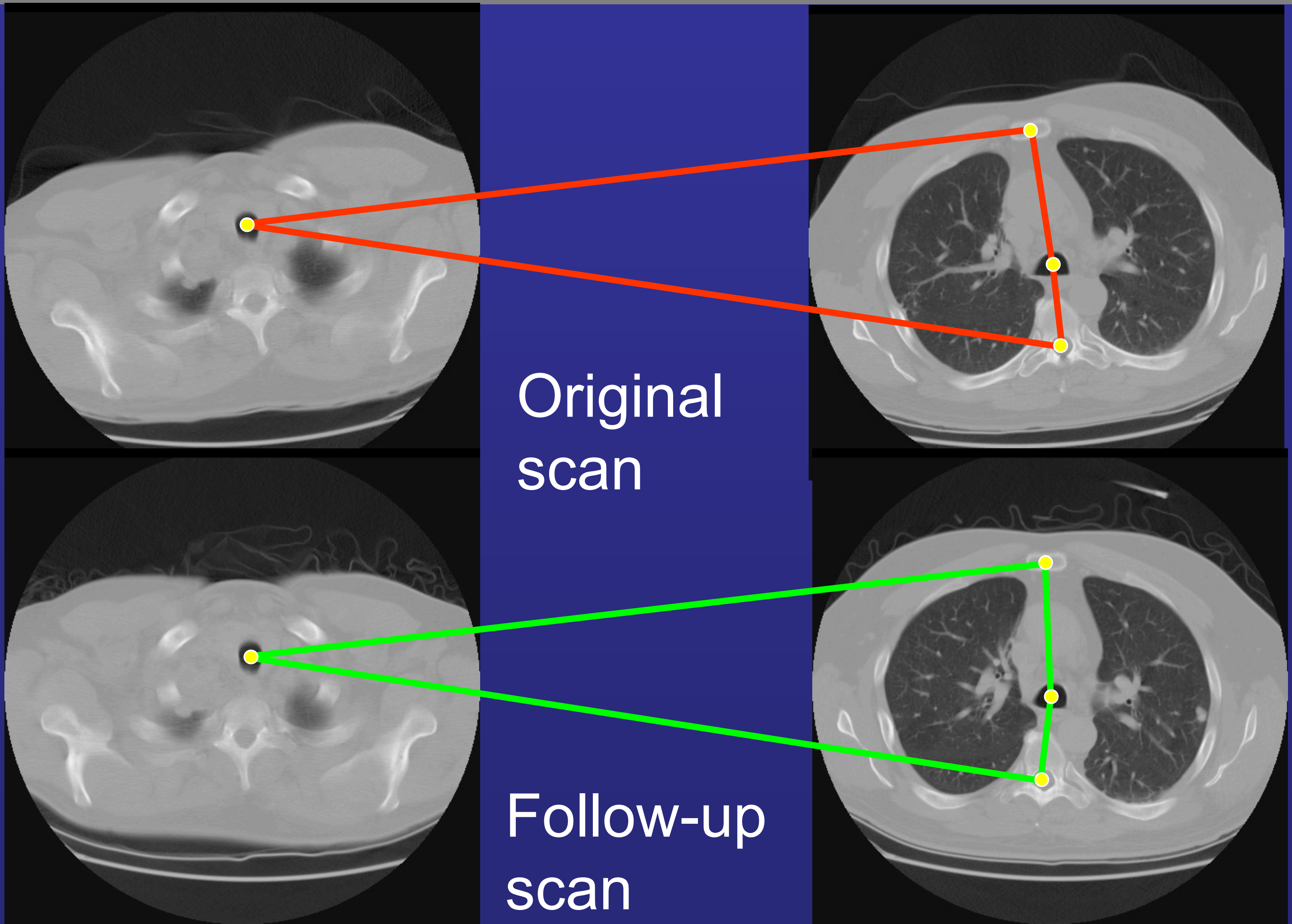
BOSTON UNIVERSITY

# Registration Method: Phase 1

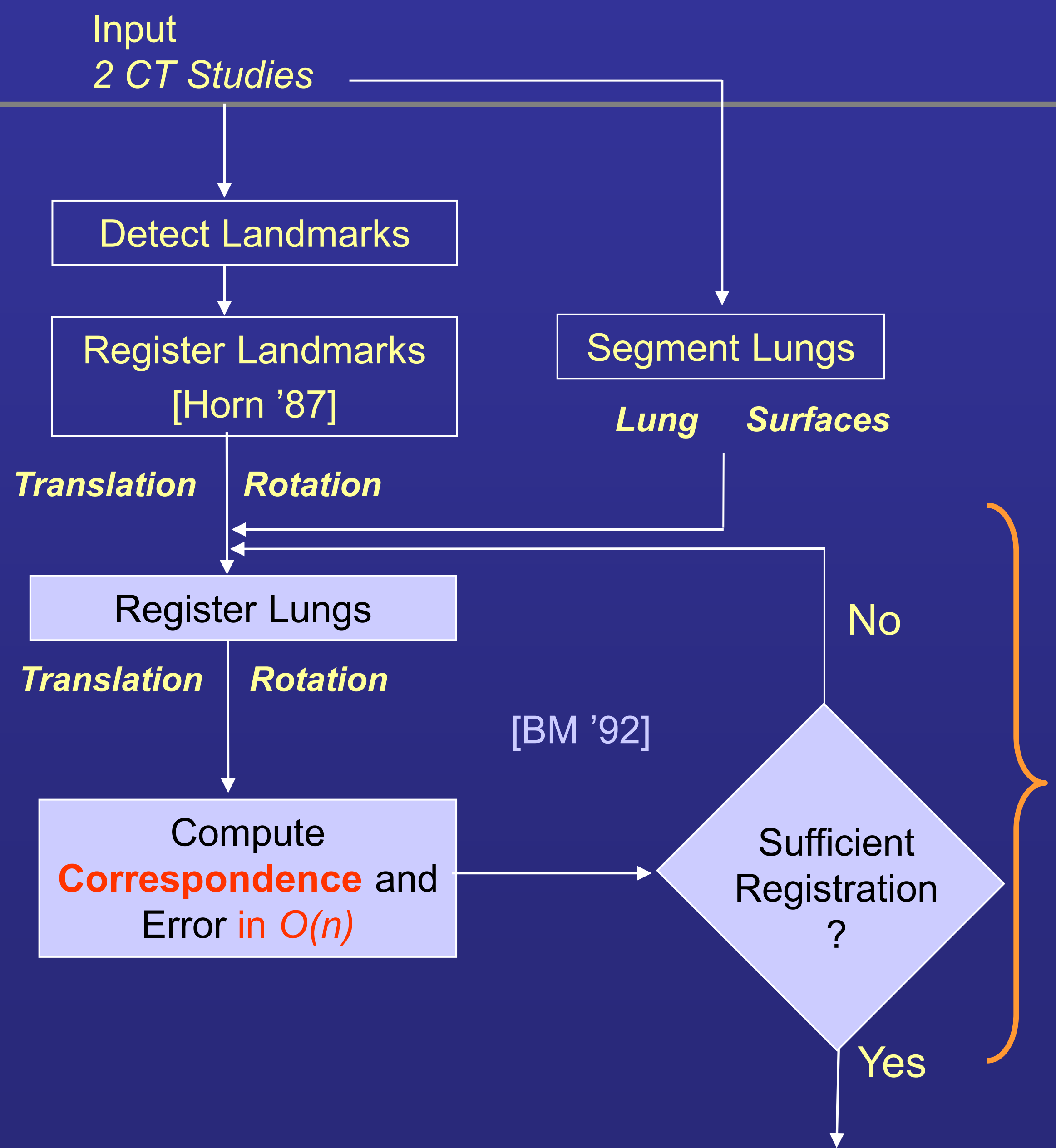


# Register Landmarks

Computer Science



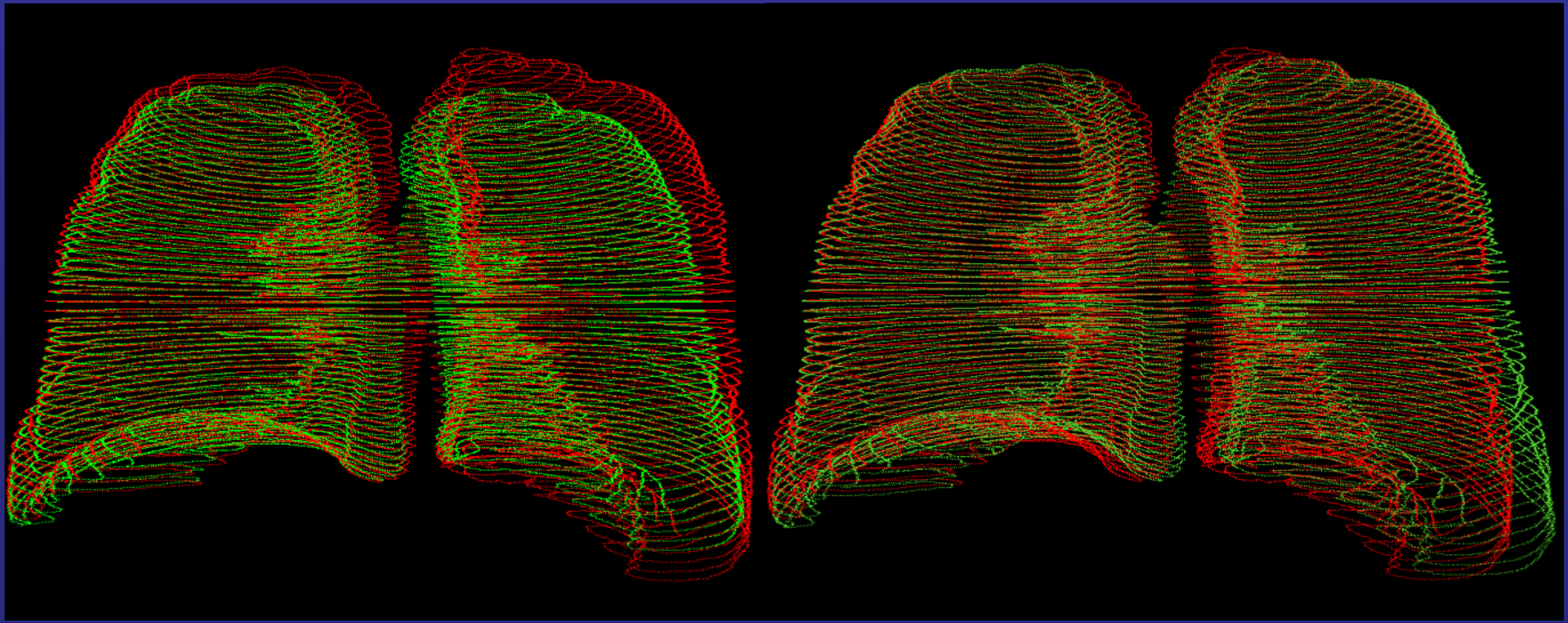
# Registration Method: Phase 2




**Multi Level Approach**

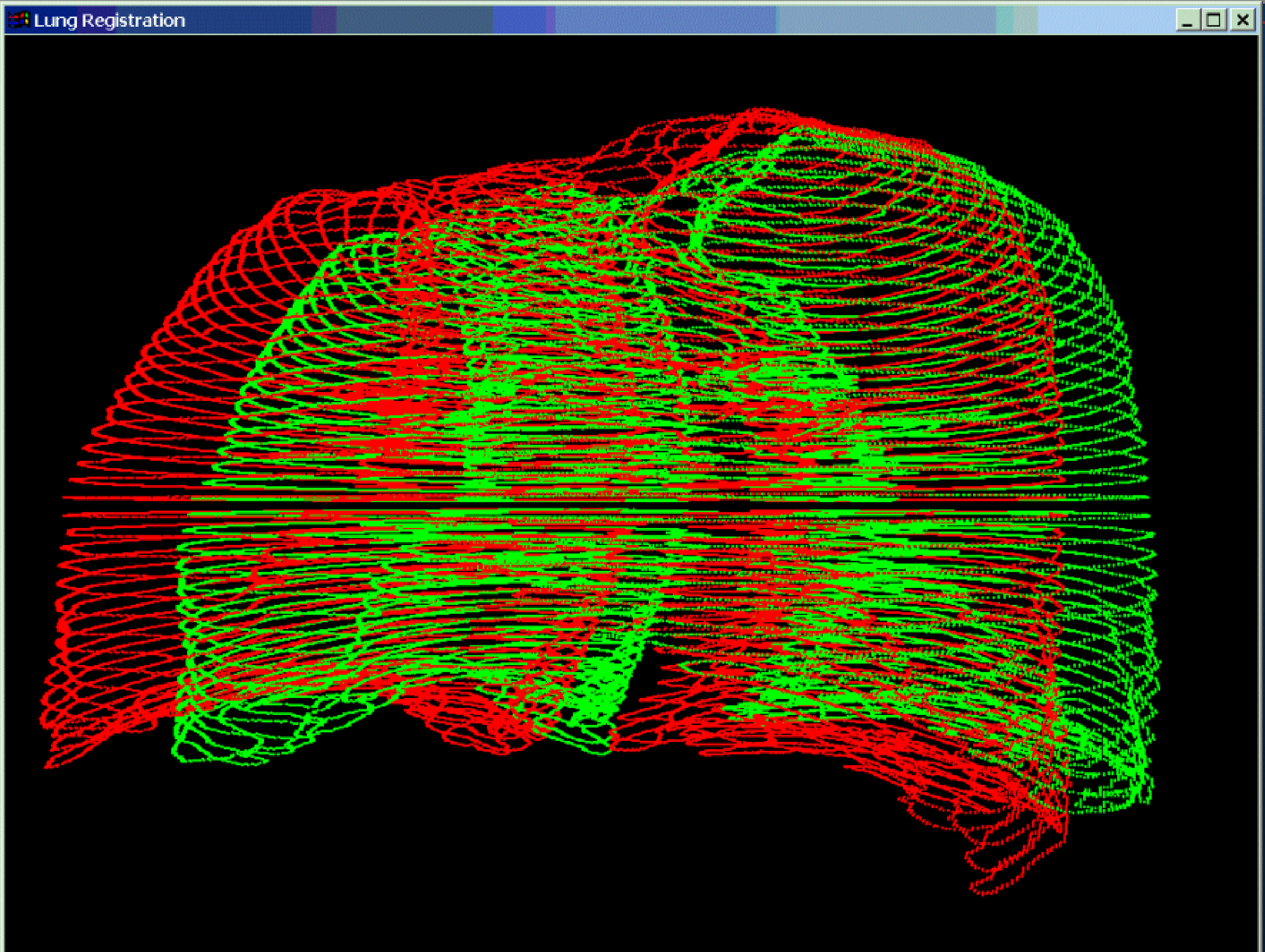
# Initial Landmark Registration

Before After initial registration

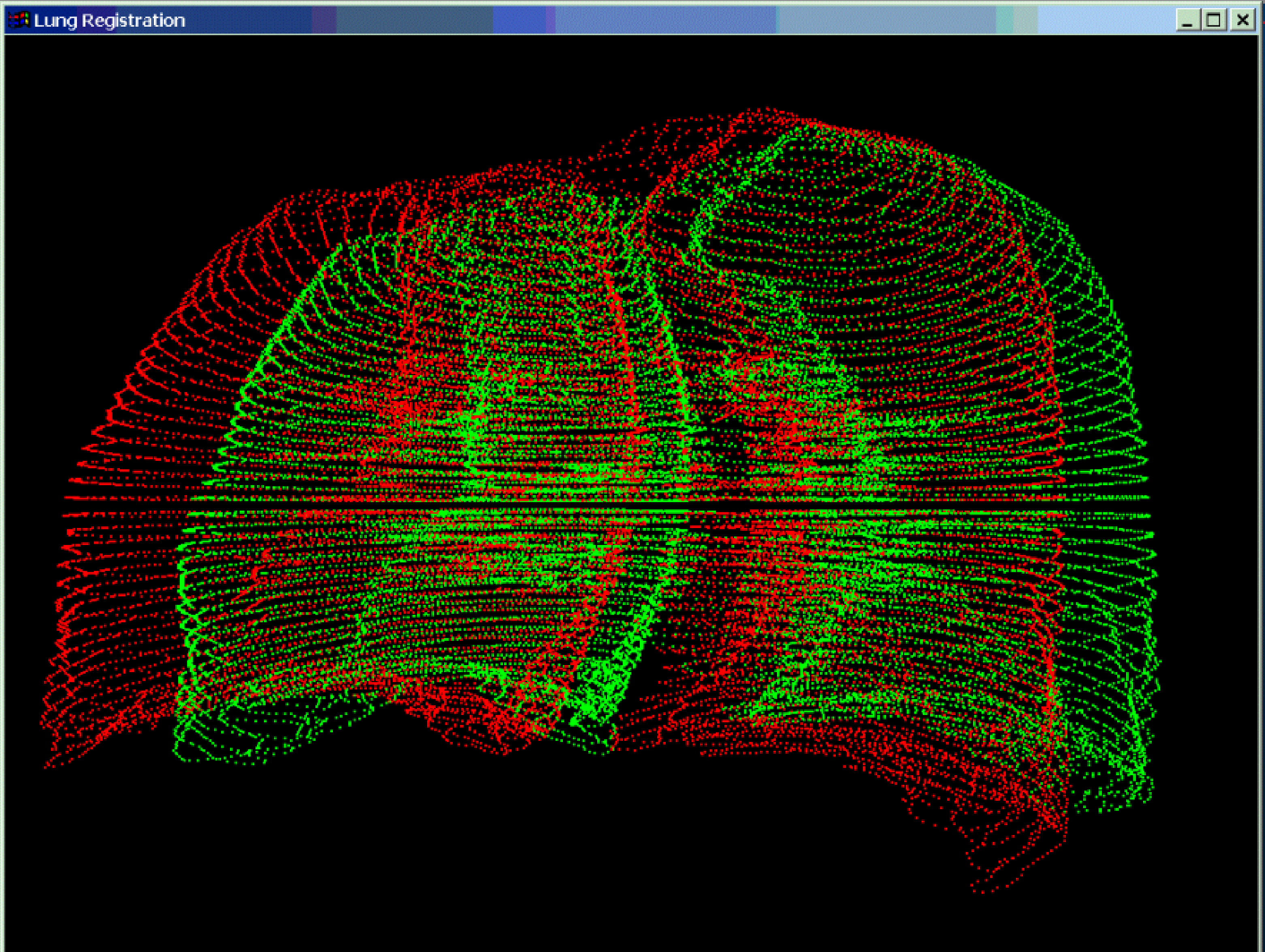


Rigid-body transformation: $p' = T(p) = R\,p + t$

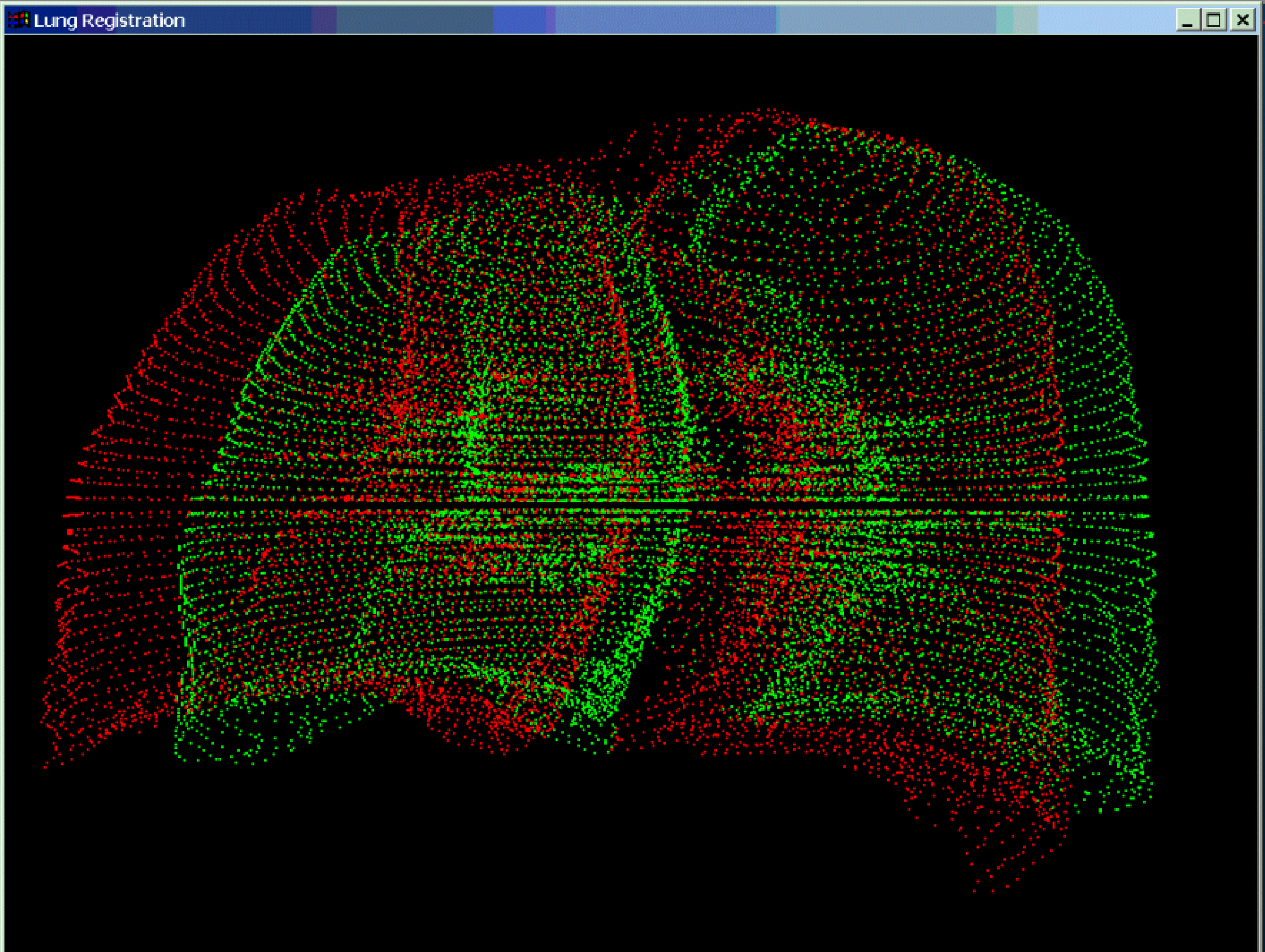
# PHASE III
# Multilevel Step - Video



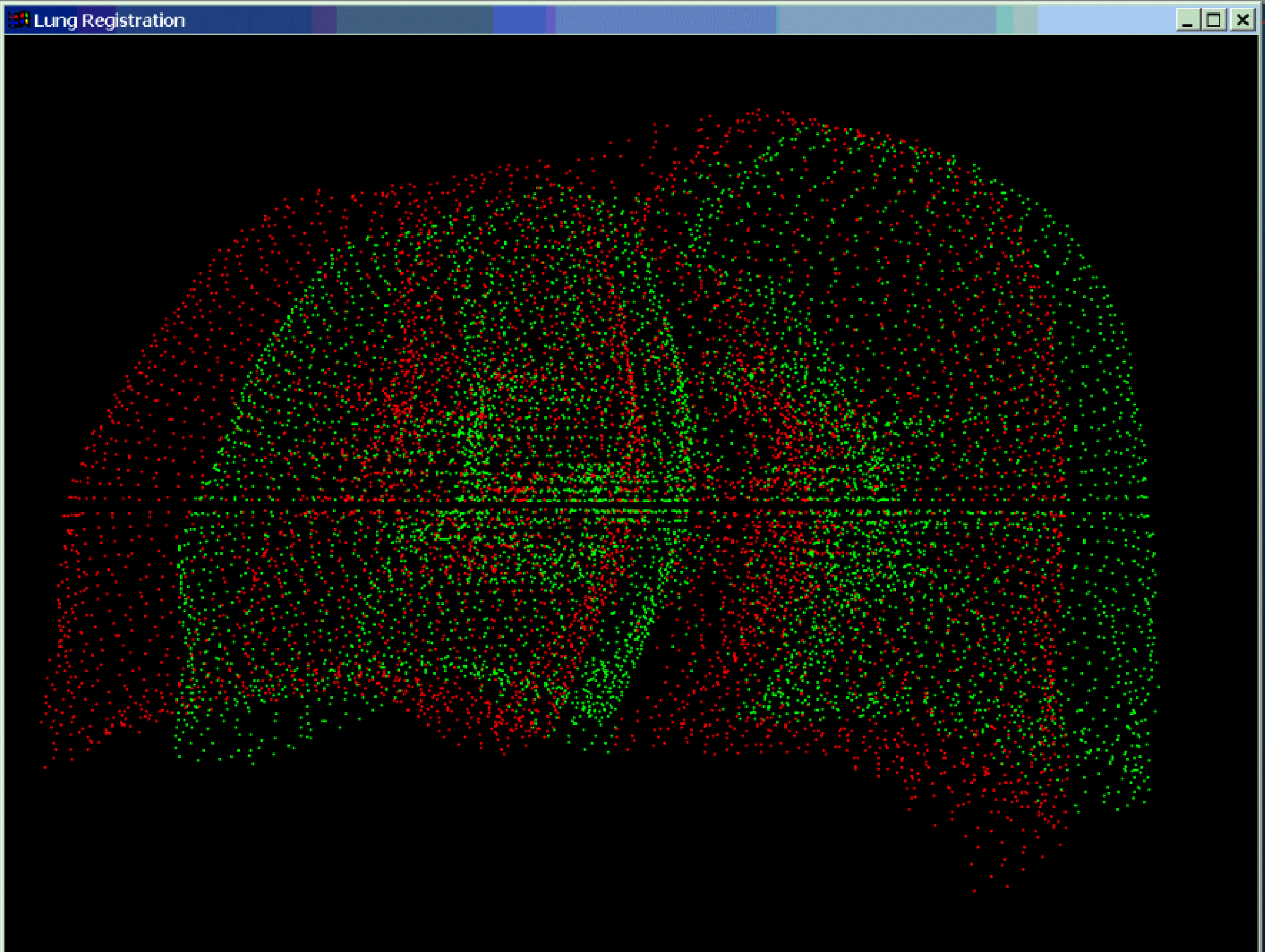
# PHASE III
# Multilevel Step - Video



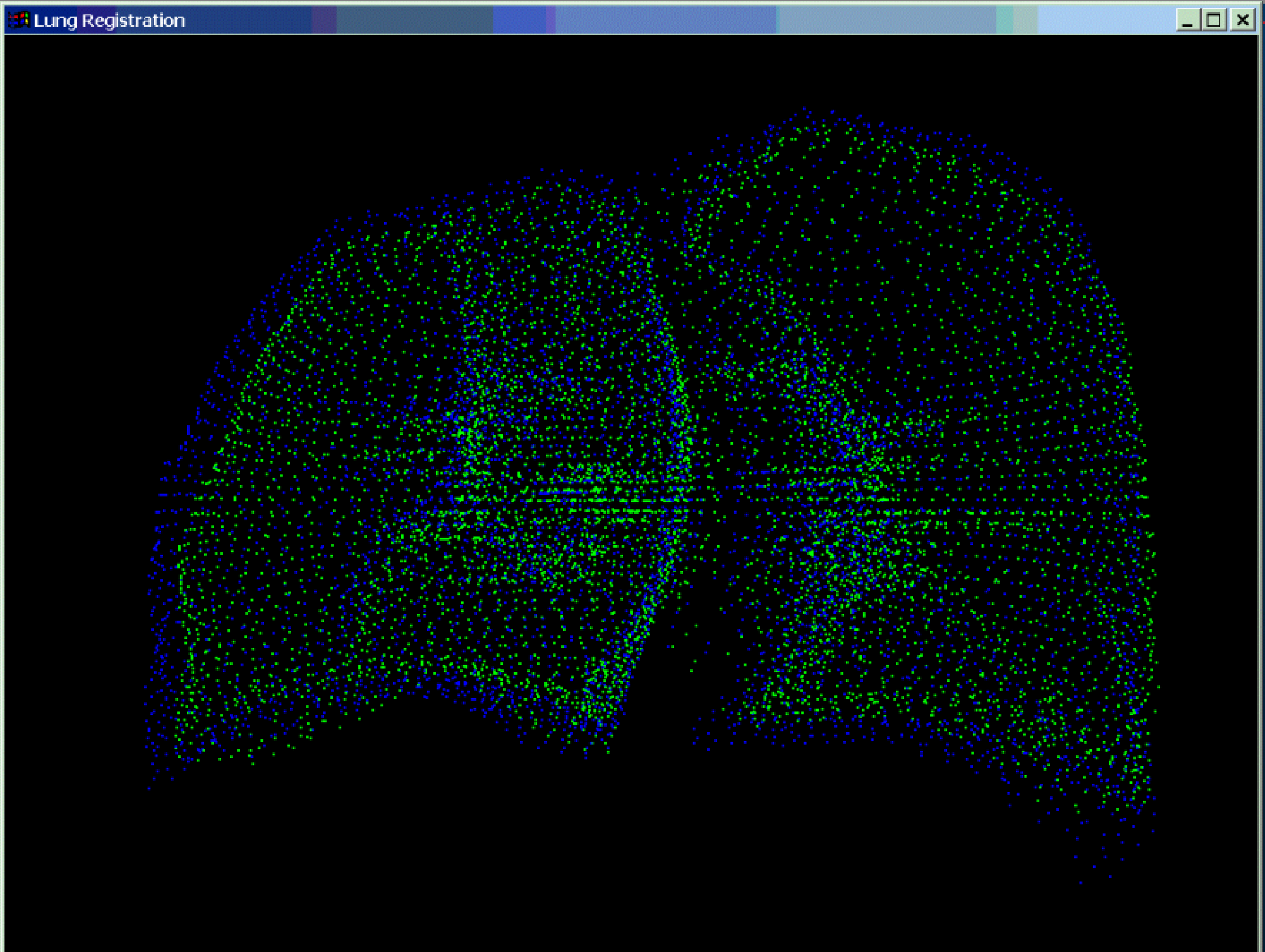
# PHASE III
# Multilevel Step - Video



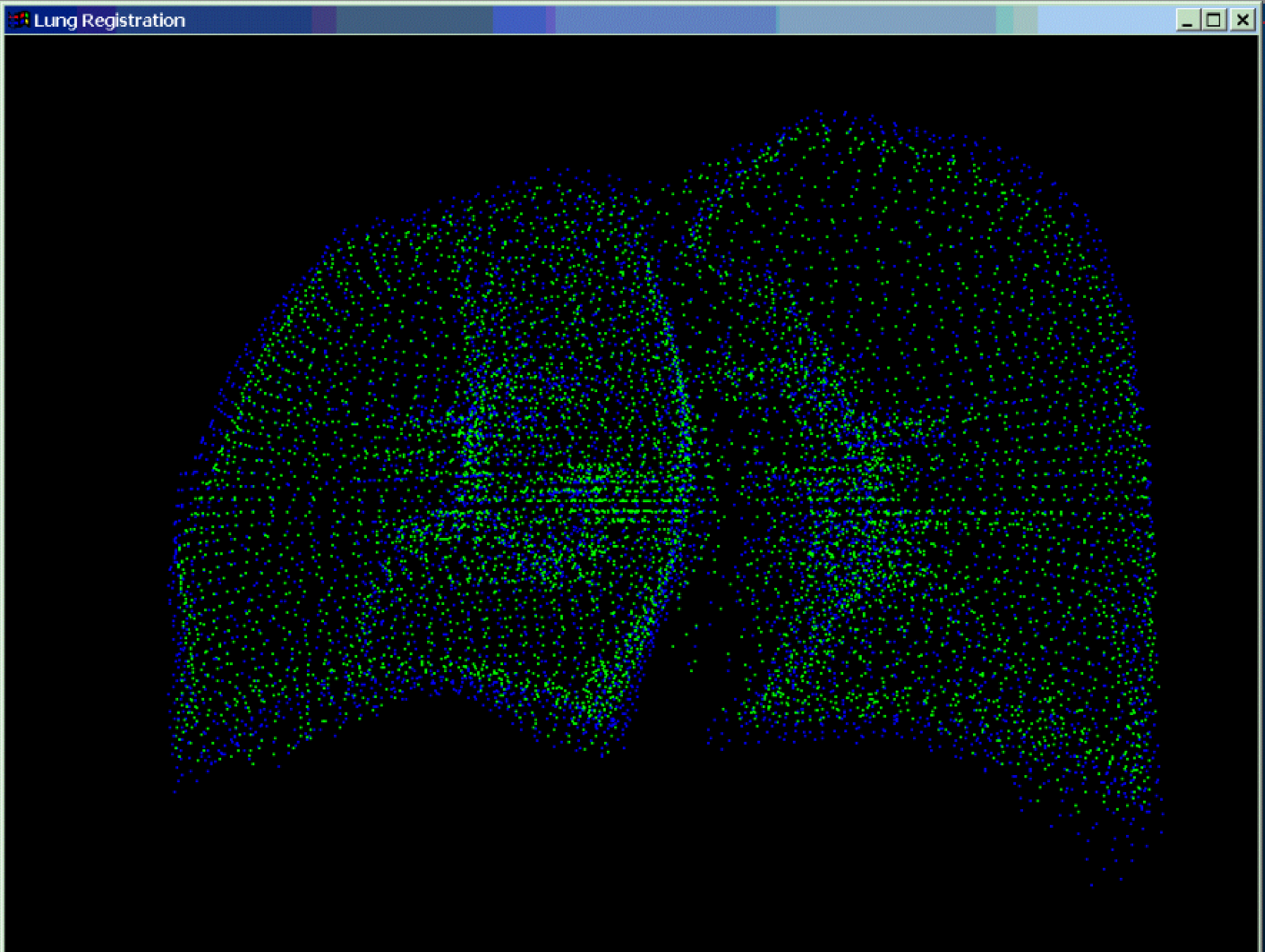
# PHASE III
# Multilevel Step - Video



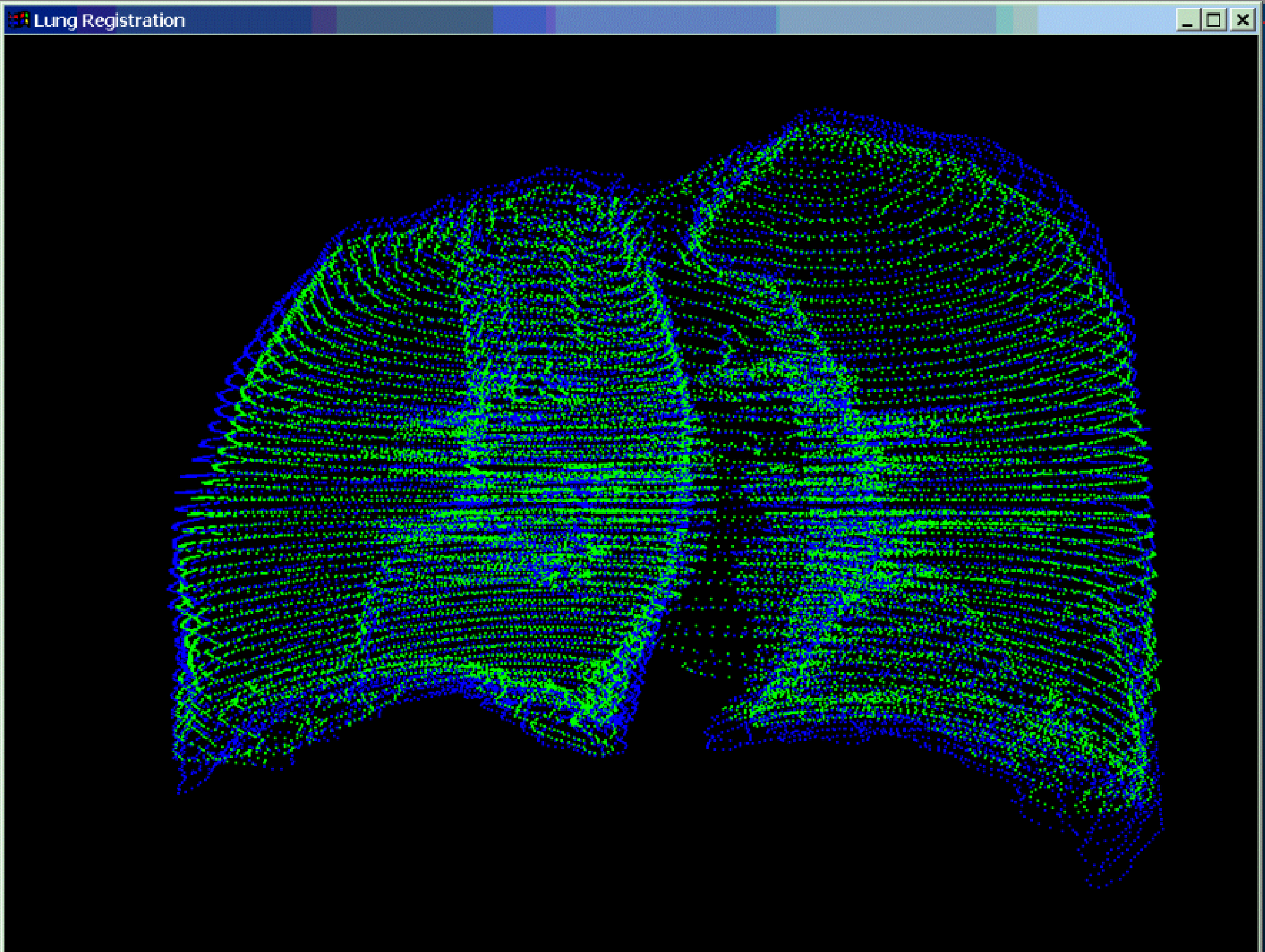
# PHASE III
# Multilevel Step - Video



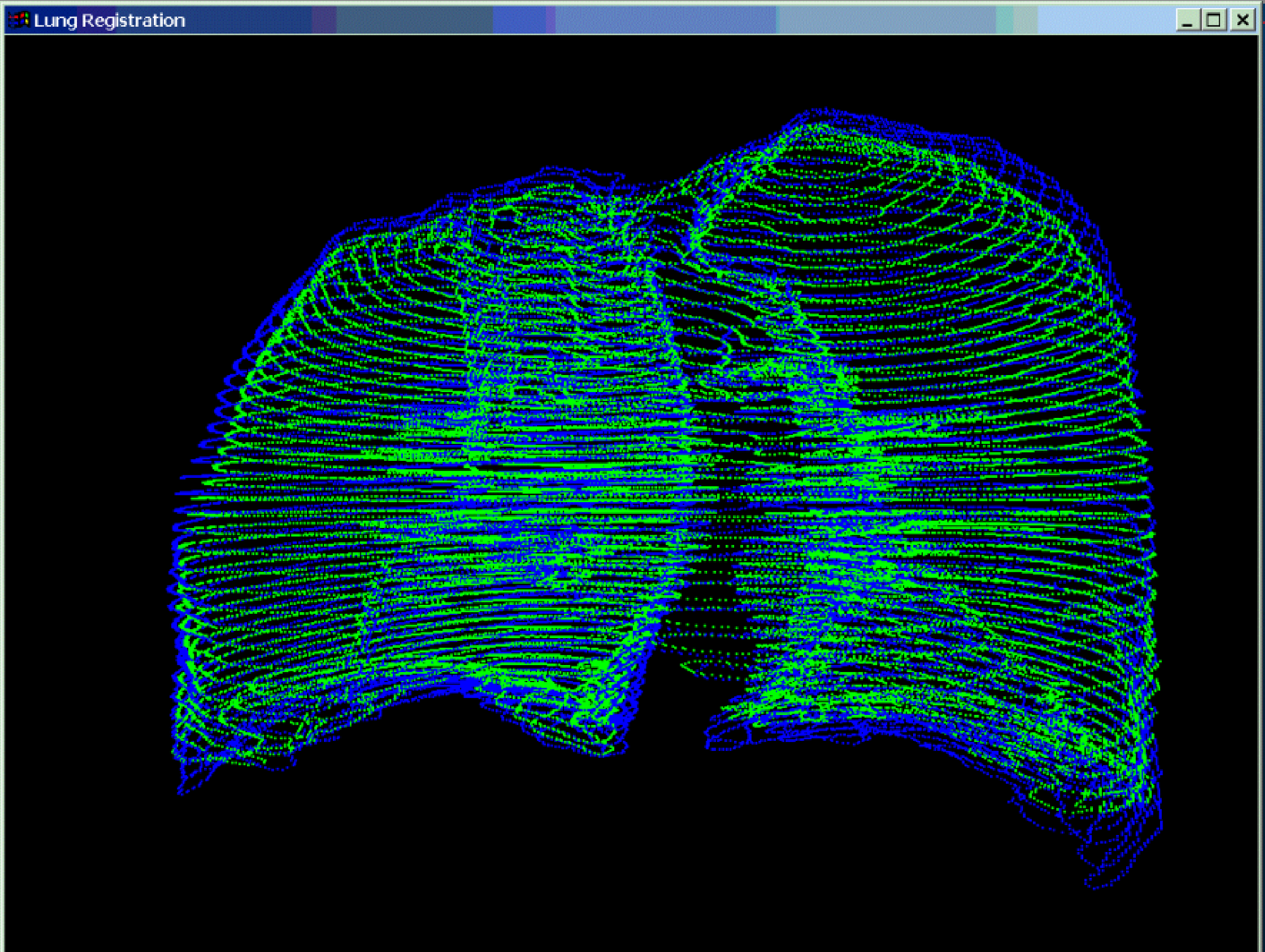
# PHASE III
# Multilevel Step - Video



# PHASE III
# Multilevel Step - Video

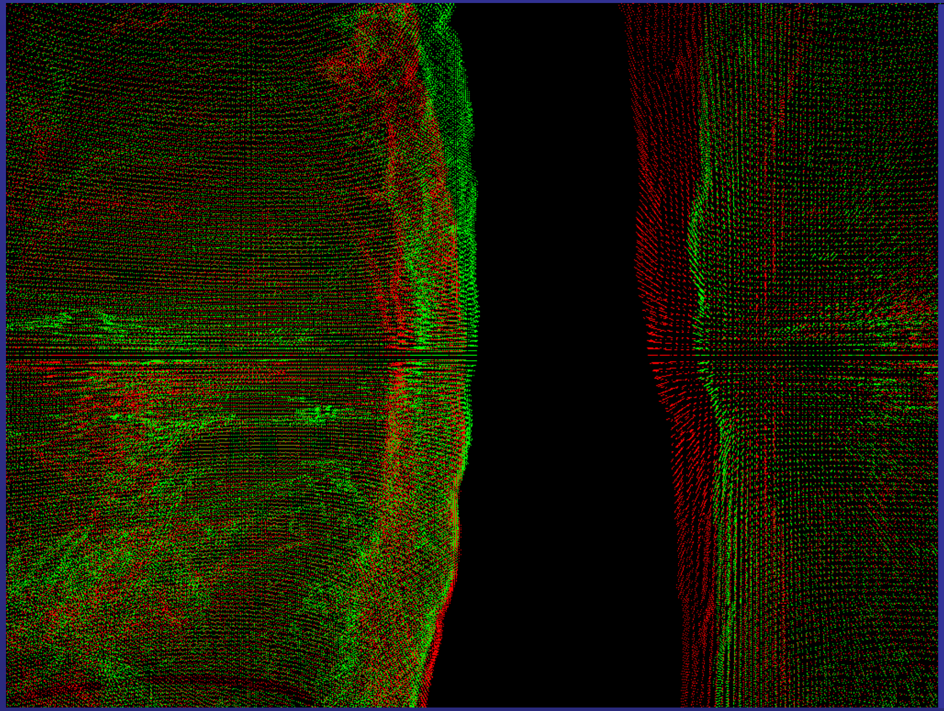


# PHASE III
# Multilevel Step - Video

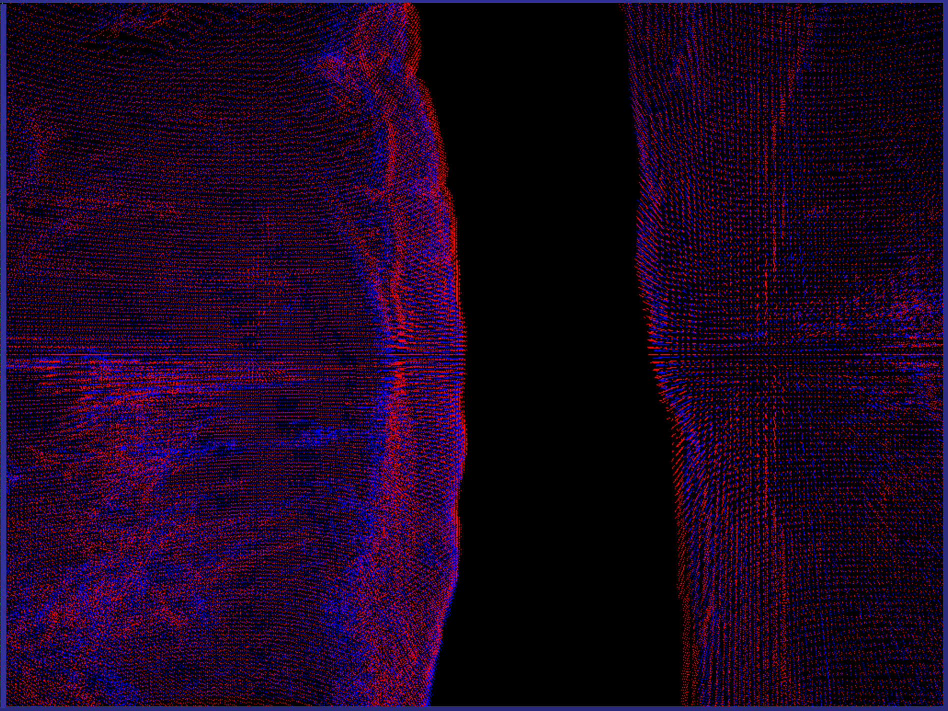


# Results on 1 mm data
## (Close-up view)

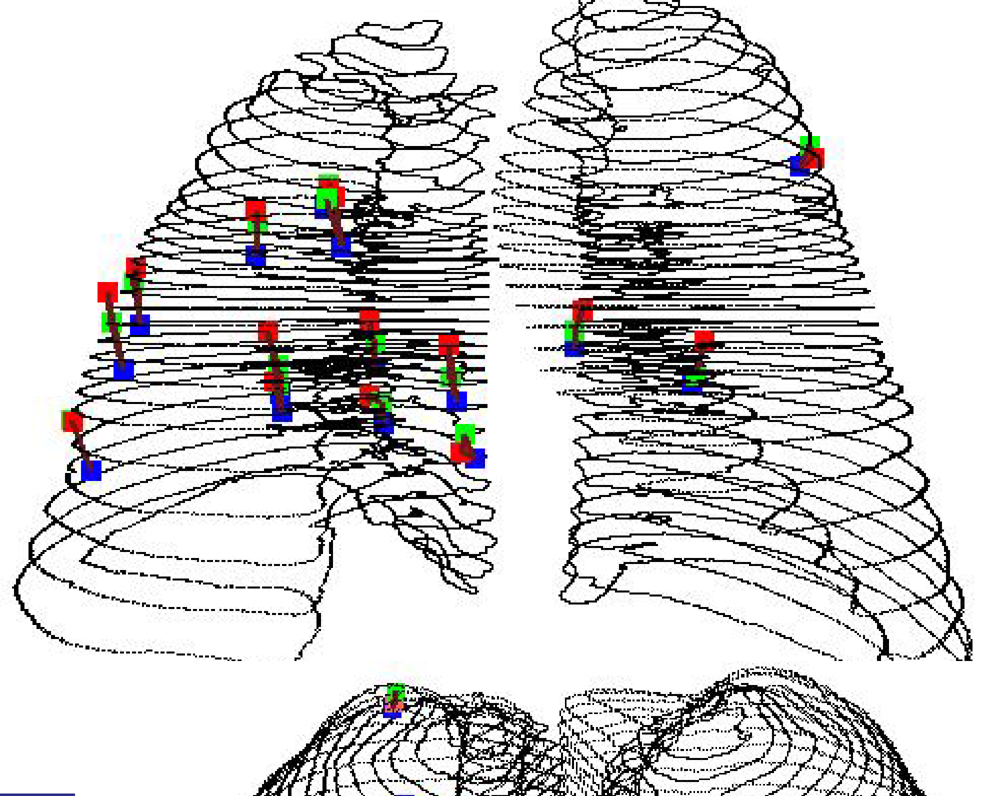
Before

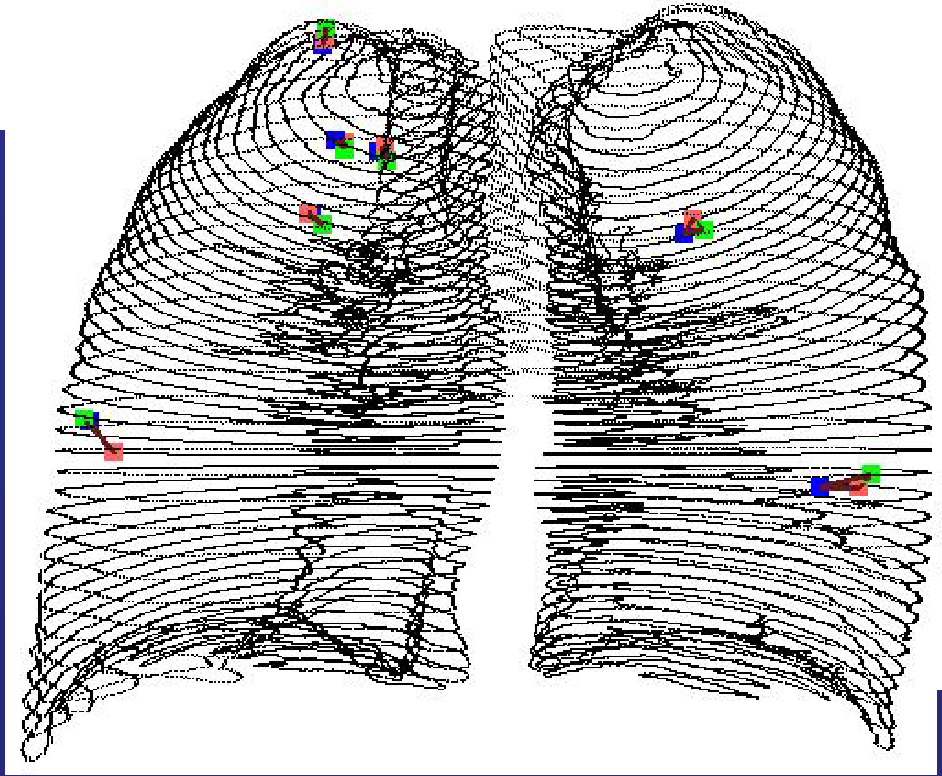


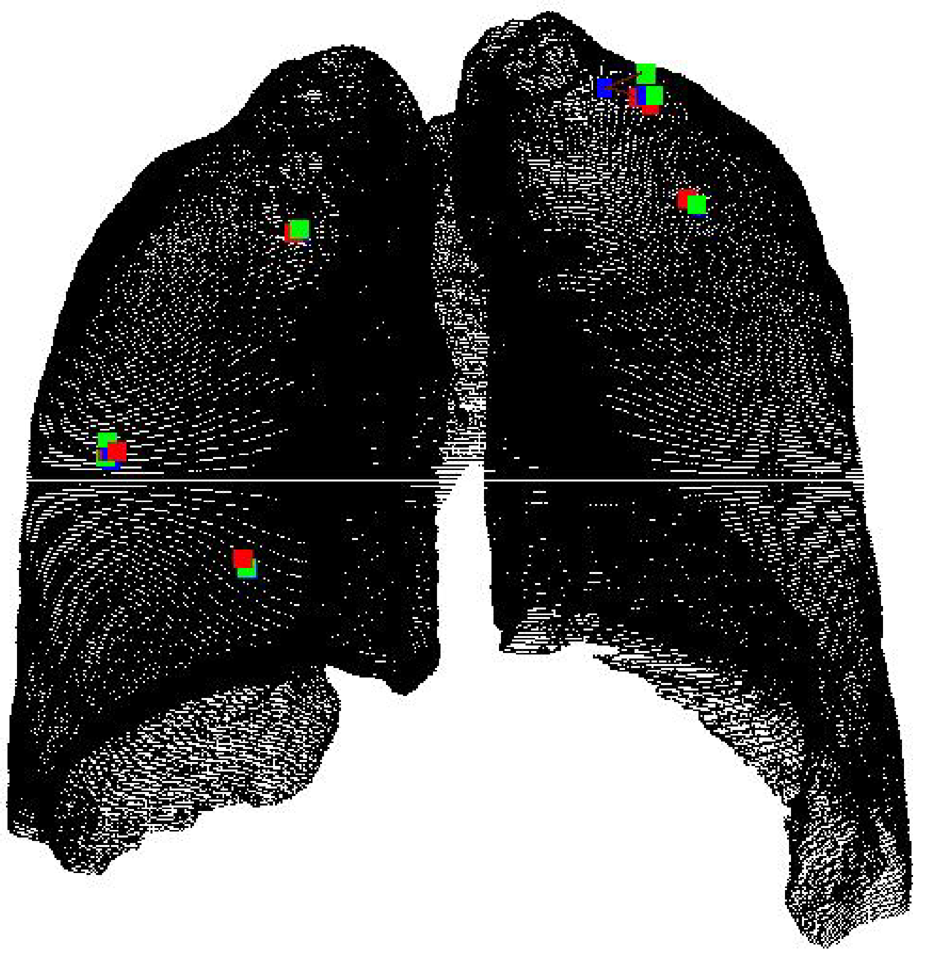
After 25th Iteration



# Nodule Registration







# Learning Outcomes

1. Can manipulate quaternions (add, multiply, convert to reals, convert rotation quaternion to rotation matrix)
2. Know Horn ‘87 algorithm for 2D & 3D registration
3. Know Besl, McKay ‘91 algorithm for curve & surface registration
4. Know how to add multi-level analysis to registration algorithm