

*Spring 2019*

# CSCI 621: **Digital Geometry Processing**

## 7.1 **Shape Matching**



Hao Li

<http://cs621.hao-li.com>

# Acknowledgement

## Images and Slides are courtesy of

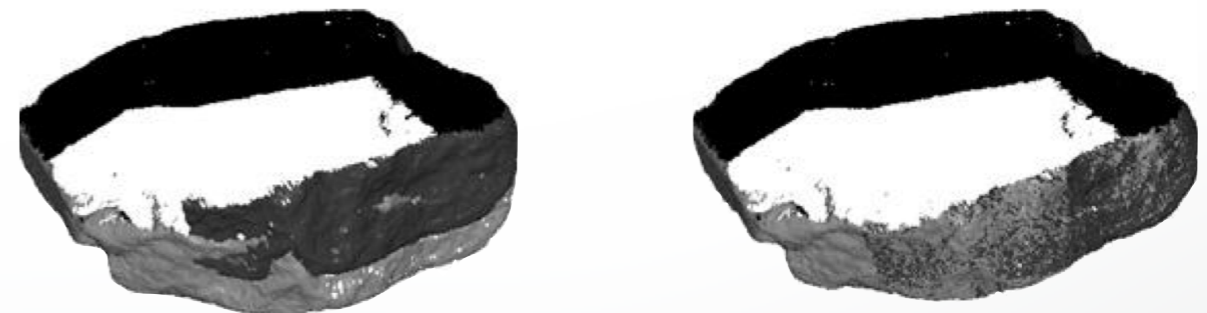
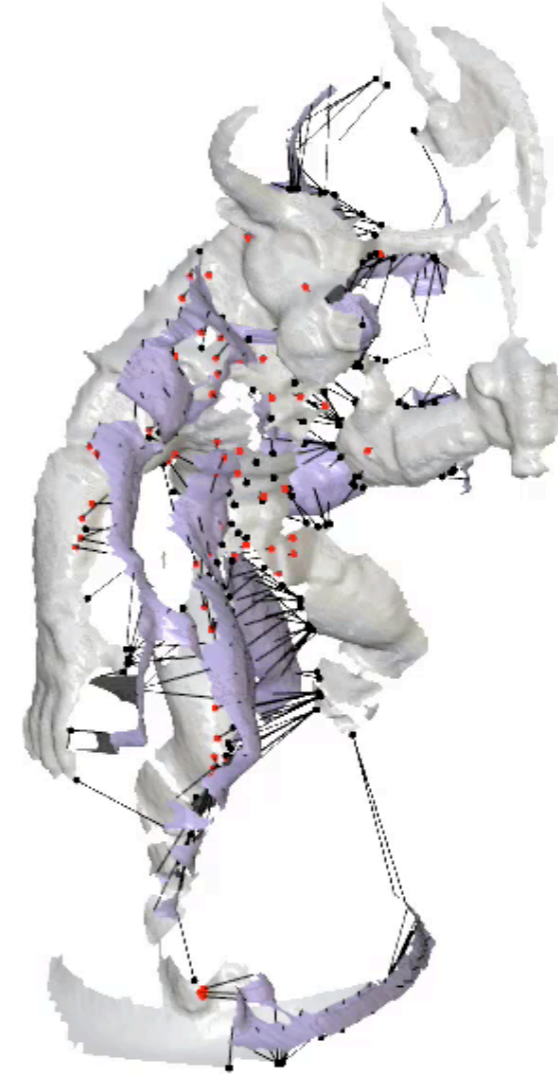
- Prof. Michael Kazhdan, Johns Hopkins University
- ICCV Course 2005: [http://gfx.cs.princeton.edu/proj/iccv05\\_course/](http://gfx.cs.princeton.edu/proj/iccv05_course/)



# Last Time

## Surface Registration

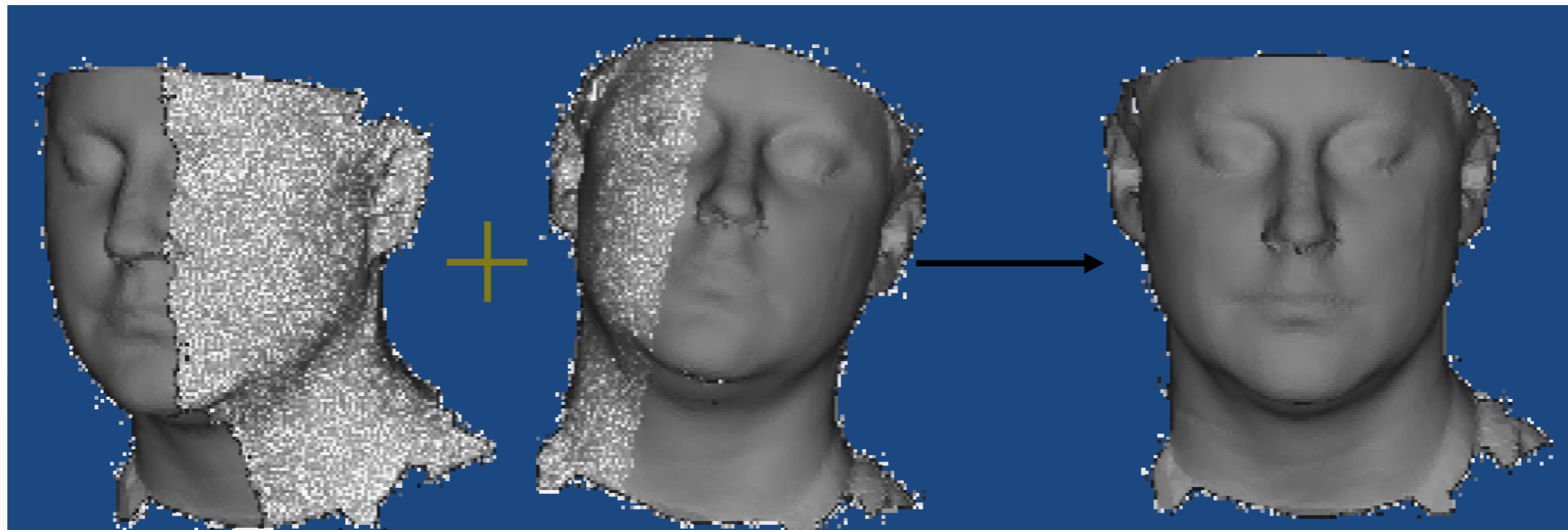
- Pairwise ICP & Variants
- Point-to-point/plane metric
- BSP closes point search
- Stability Analysis
- Global Registration



# Shape Matching for Model Alignment

## Goal

- Given two partially overlapping scans, compute transformation that aligns the two.
- **No assumption** about rough initial alignment



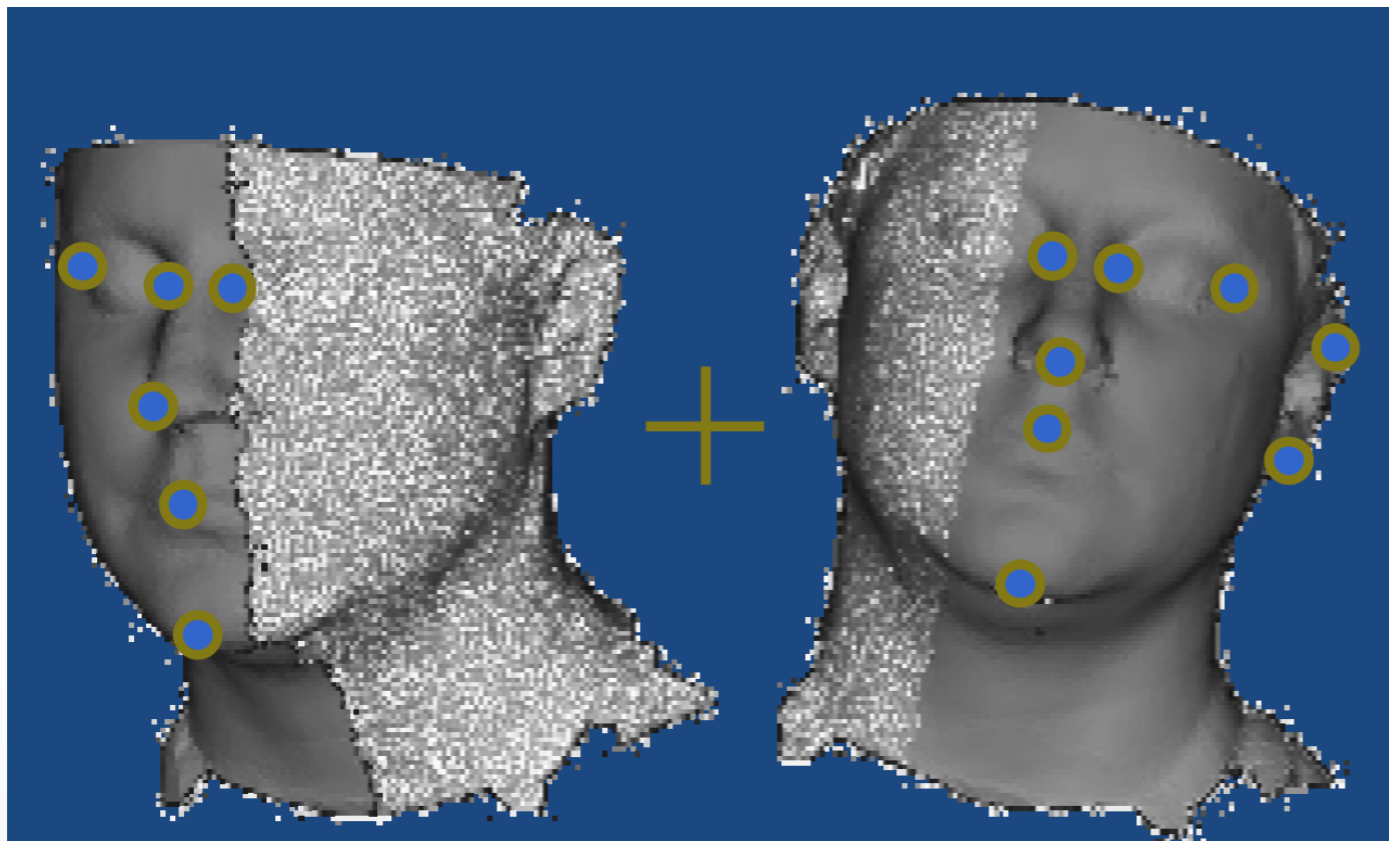
Partially Overlapping Scans

Aligned Scans

# Shape Matching for Model Alignment

## Approach

- Find feature points on the two scans

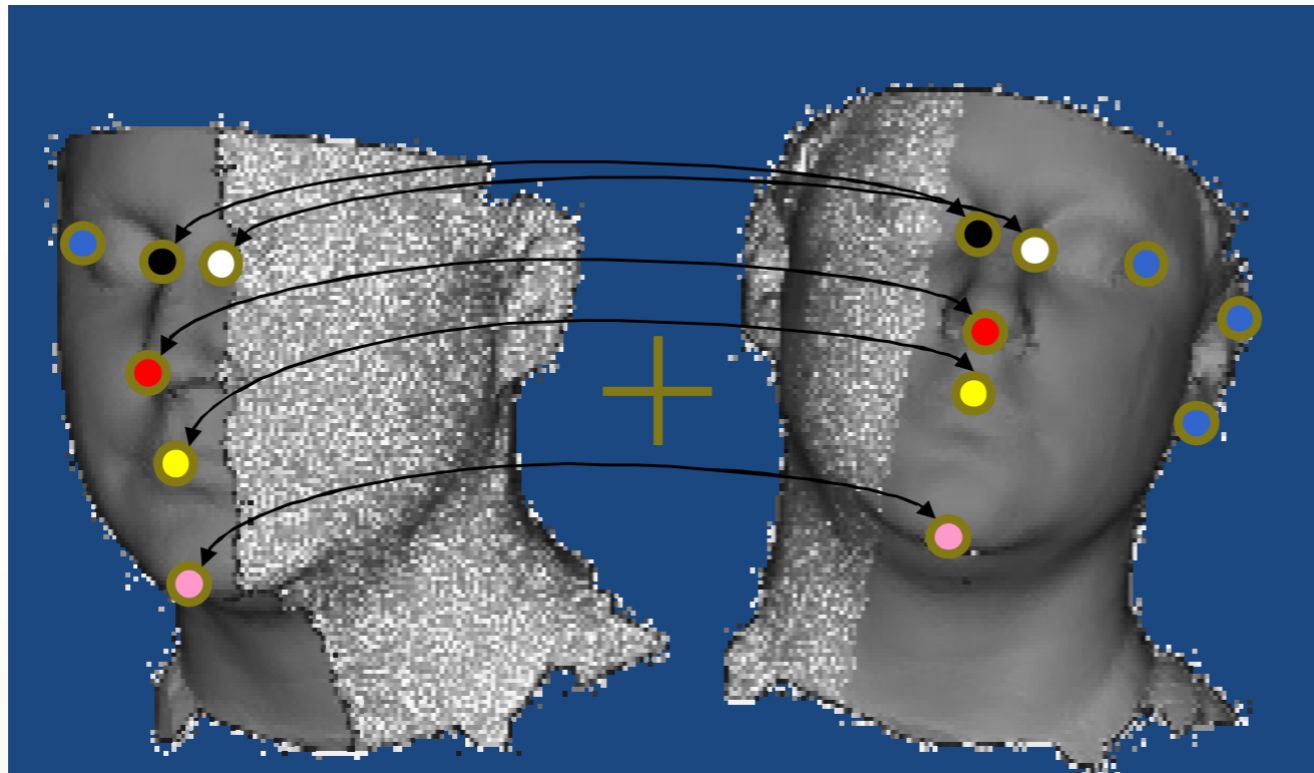


Partially Overlapping Scans

# Shape Matching for Model Alignment

## Approach

- Find feature points on the two scans
- Establish correspondences

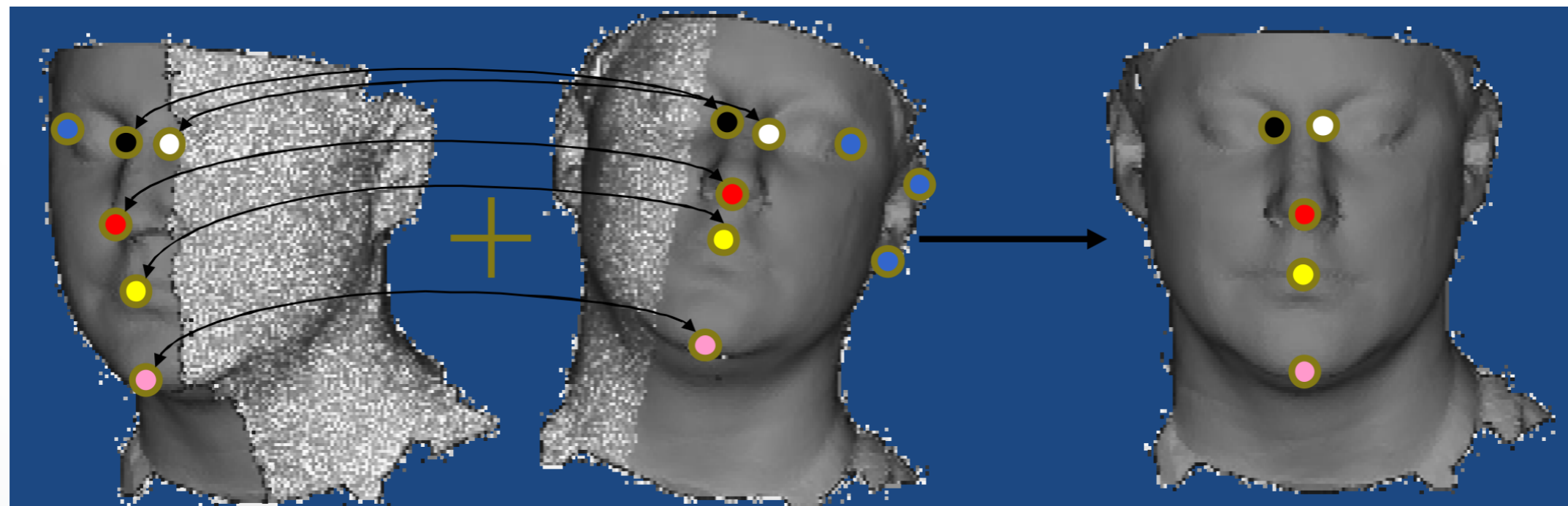


Partially Overlapping Scans

# Shape Matching for Model Alignment

## Approach

- Find feature points on the two scans
- Establish correspondences
- Compute the alignment



Partially Overlapping Scans

Aligned Scans

# Outline

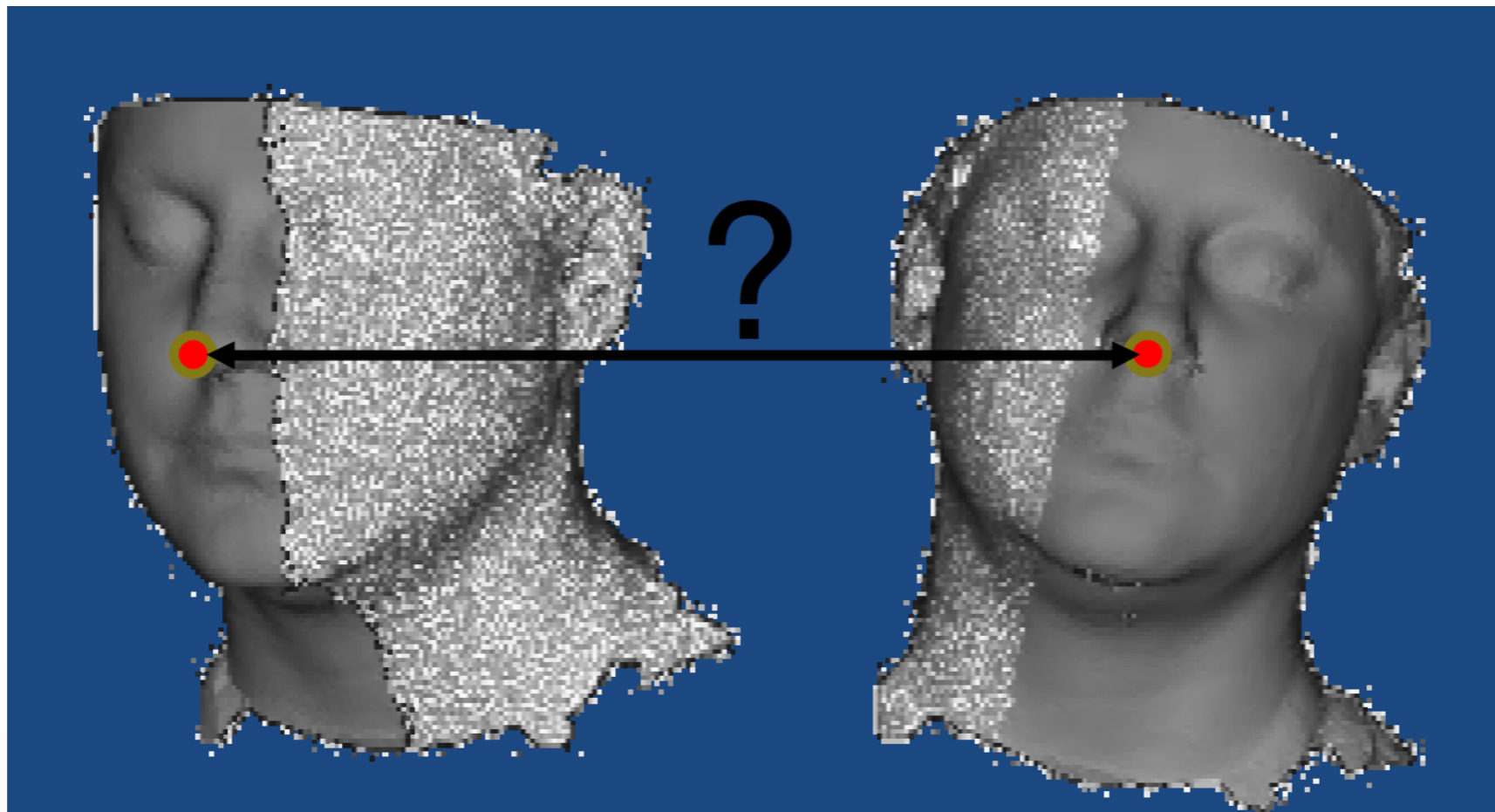
- **Global Shape Correspondence**
  - Shape Descriptors
  - Alignment
- **Partial Shape Correspondence**
  - From Global to Local
  - Pose Normalization
  - Partial Shape Descriptors
- **Registration**
  - Closed Form Solutions
  - Branch & Bound
  - Random Sample Consensus (RANSAC)



# Correspondence

## Goal

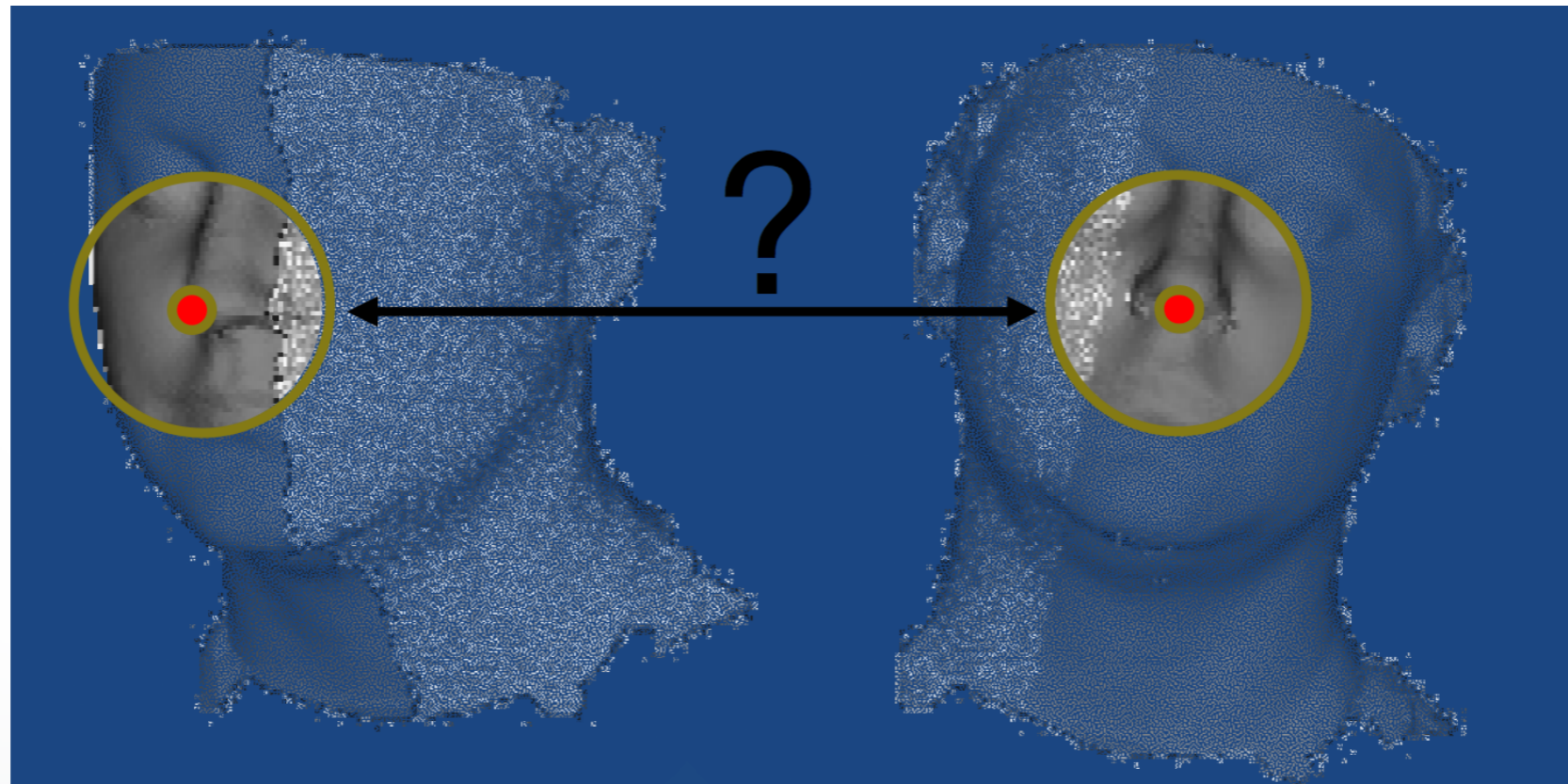
- Identify when two points on different scans represent the same feature



# Local Correspondence

## Goal

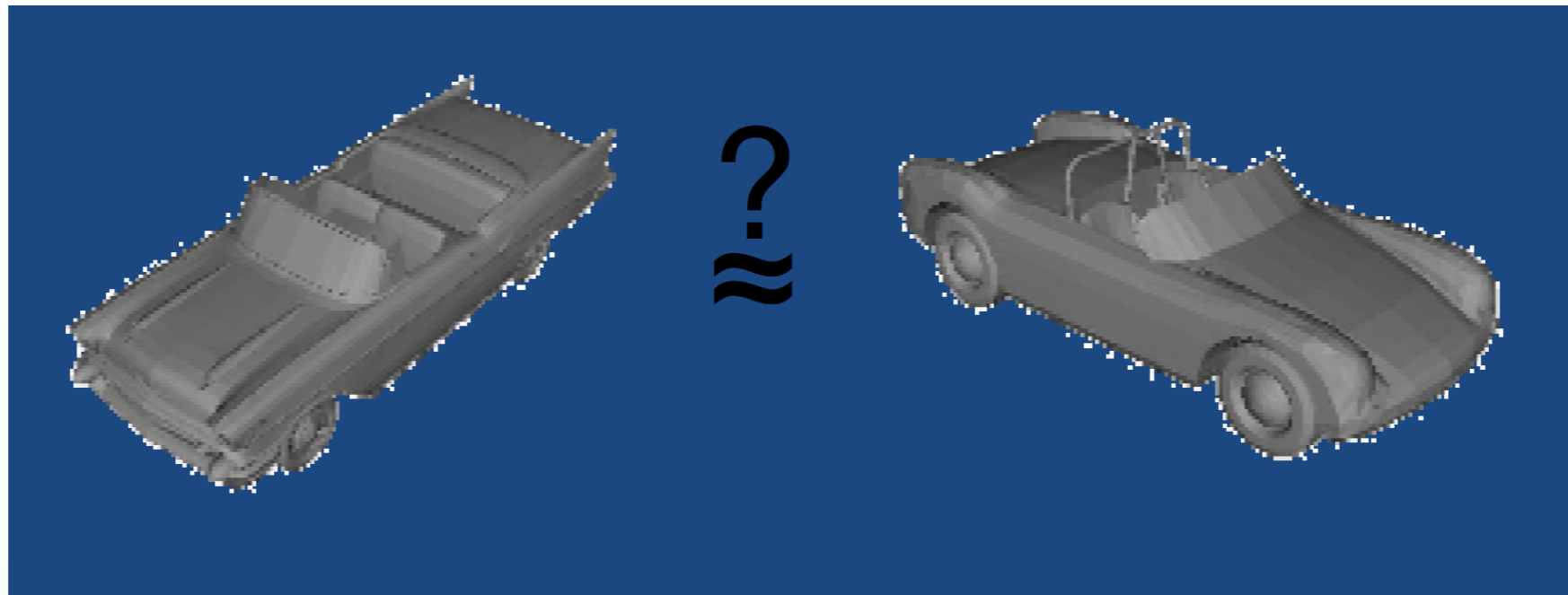
- Identify when two points on different scans represent the same feature
  - Are the surrounding regions similar?



# Global Correspondence

## More Generally:

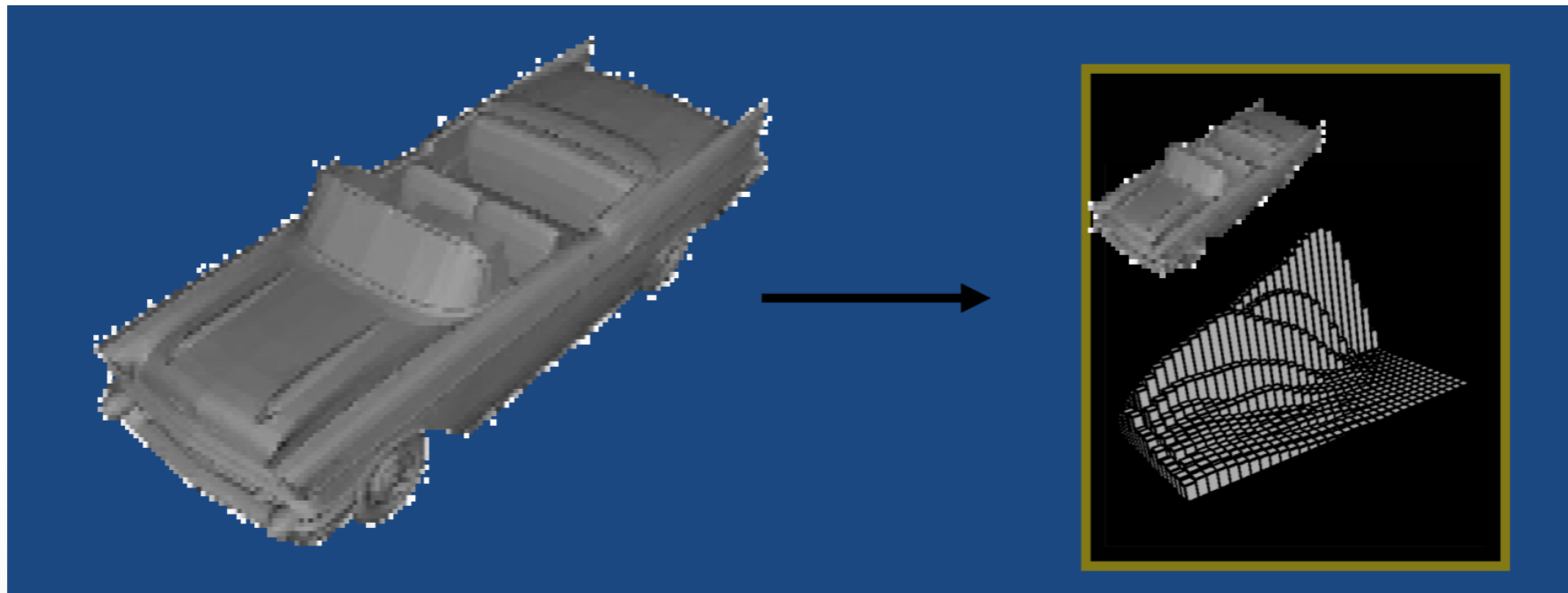
- Given two models, determine if they represent the same/similar shapes
- models can have different representations, tessellations, topologies, etc.



# Global Correspondence

## Approach:

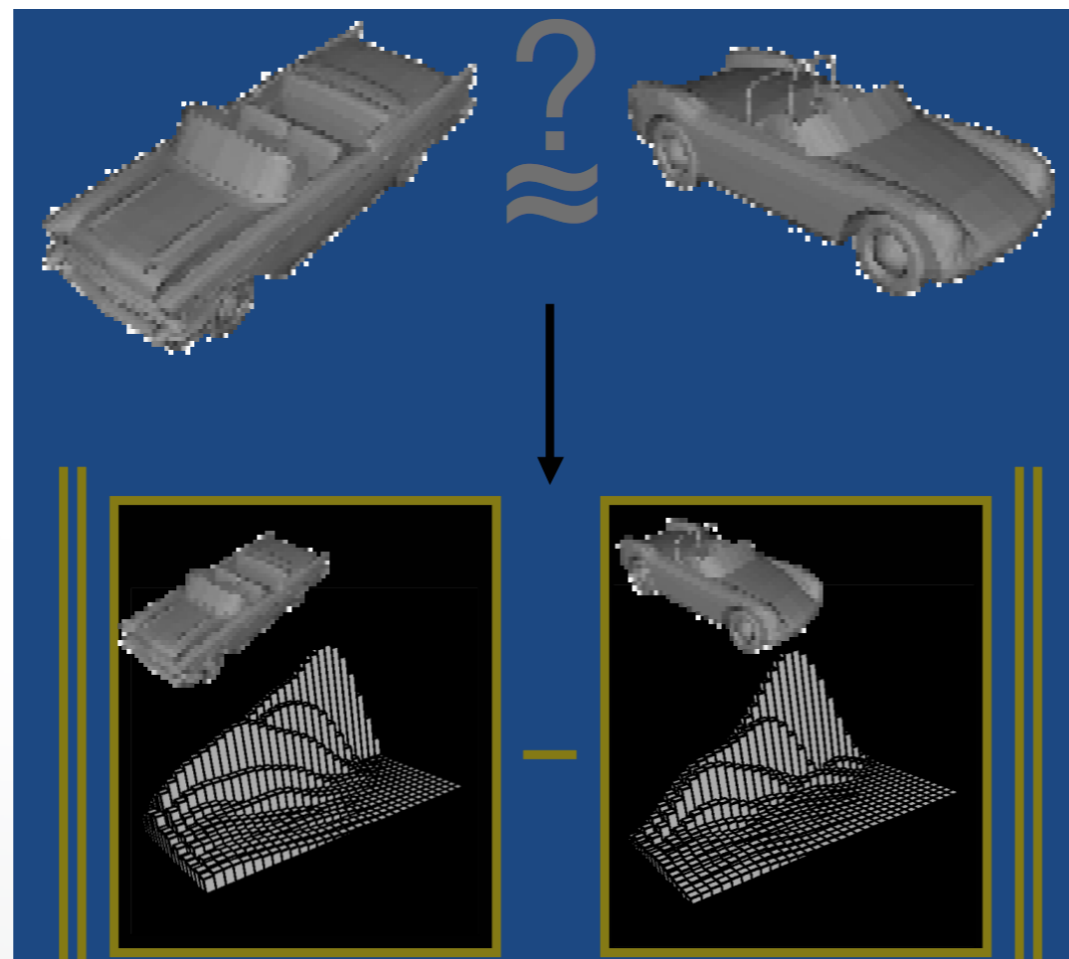
- Represent each model by a shape descriptor:
  - A **structured** abstraction of a 3D model
  - that captures **salient** shape information



# Global Correspondence

## Approach:

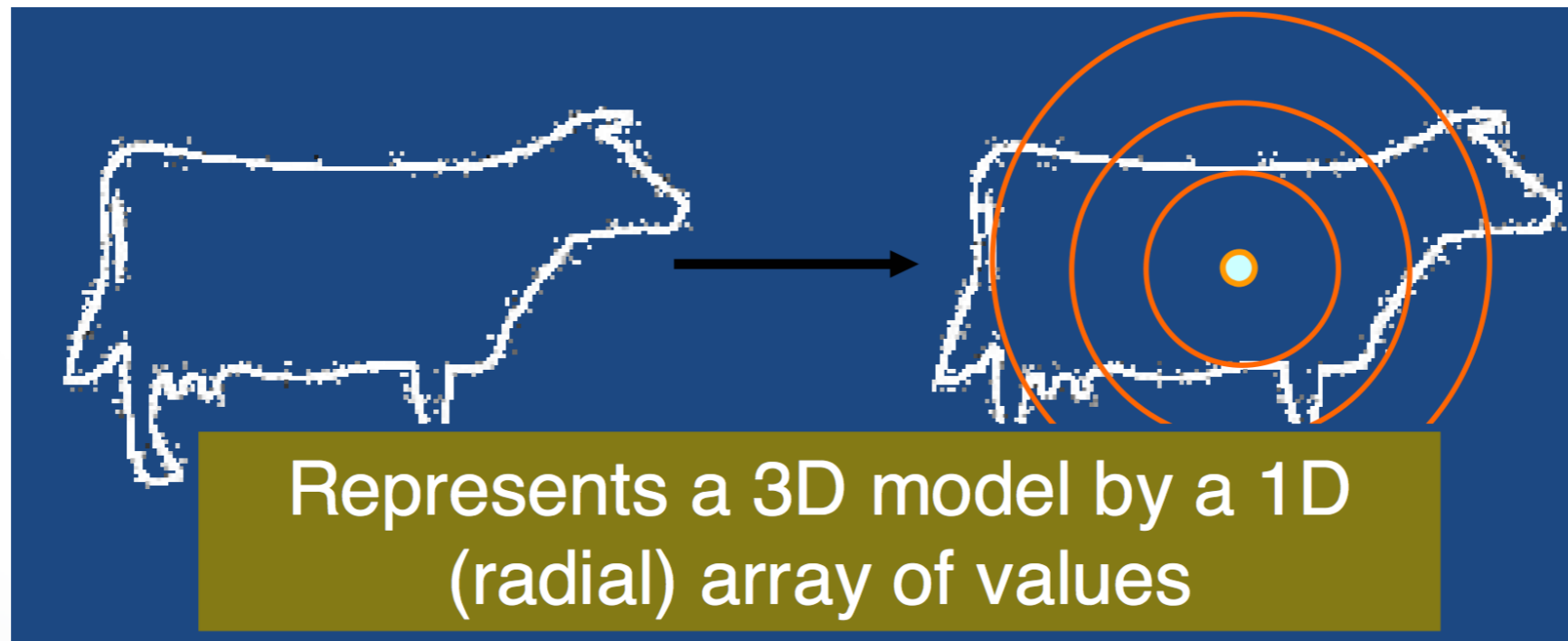
- Represent each model by a shape descriptor:
- Compare shapes by comparing their shape descriptors



# Shape Descriptors: Examples

## Shape Histograms

- Shape descriptor stores a histogram of how much surface **area** resides within different **concentric shells** in space

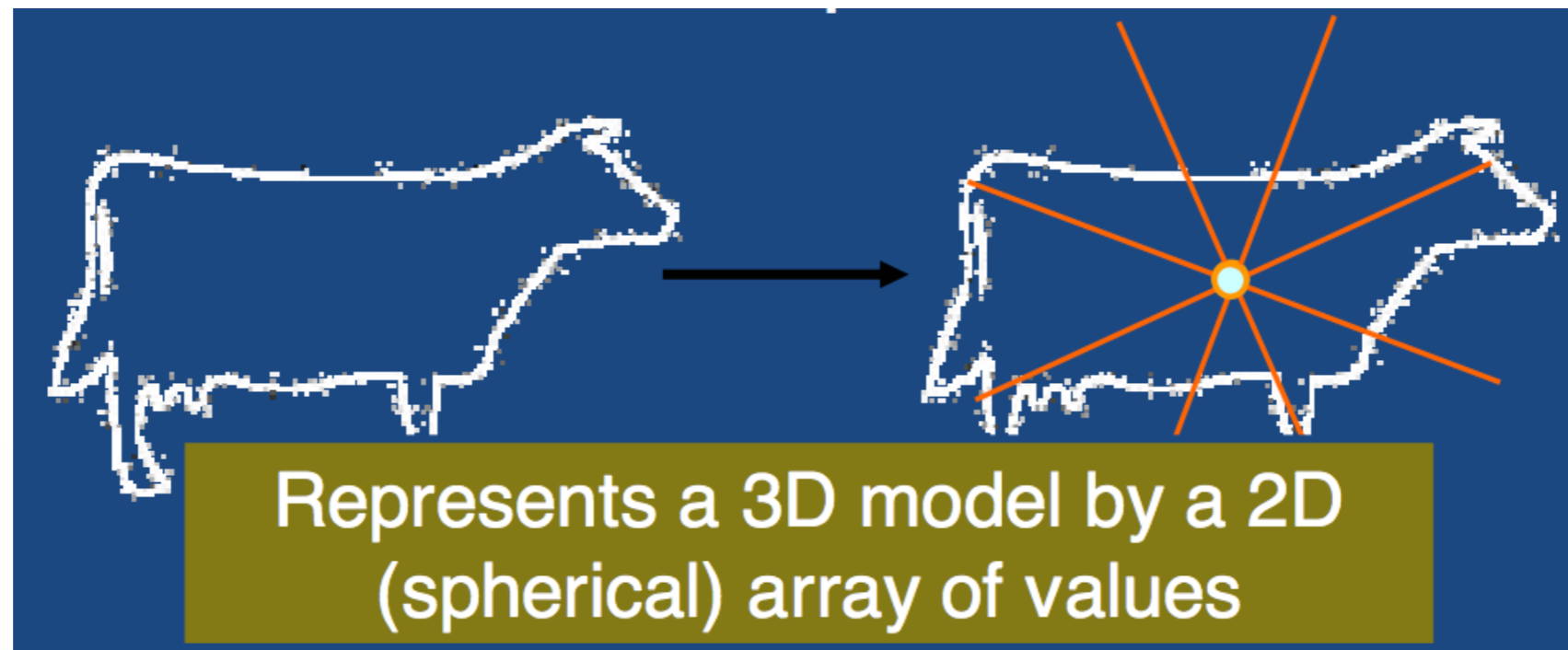


[Ankerst et al. 1999]

# Shape Descriptors: Examples

## Shape Histograms

- Shape descriptor stores a histogram of how much surface **area** resides within different **sectors** in space

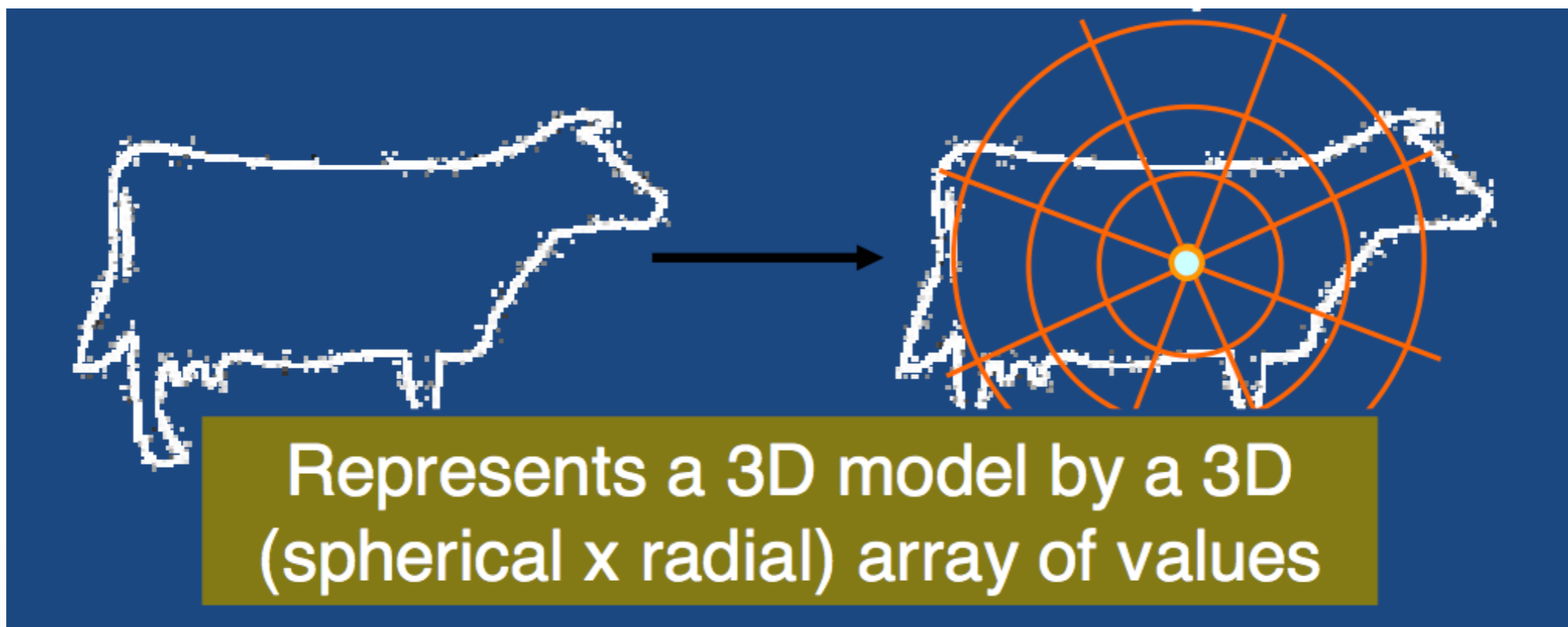


[Ankerst et al. 1999]

# Shape Descriptors: Examples

## Shape Histograms

- Shape descriptor stores a histogram of how much surface **area** resides within different **shells** and **sectors** in space

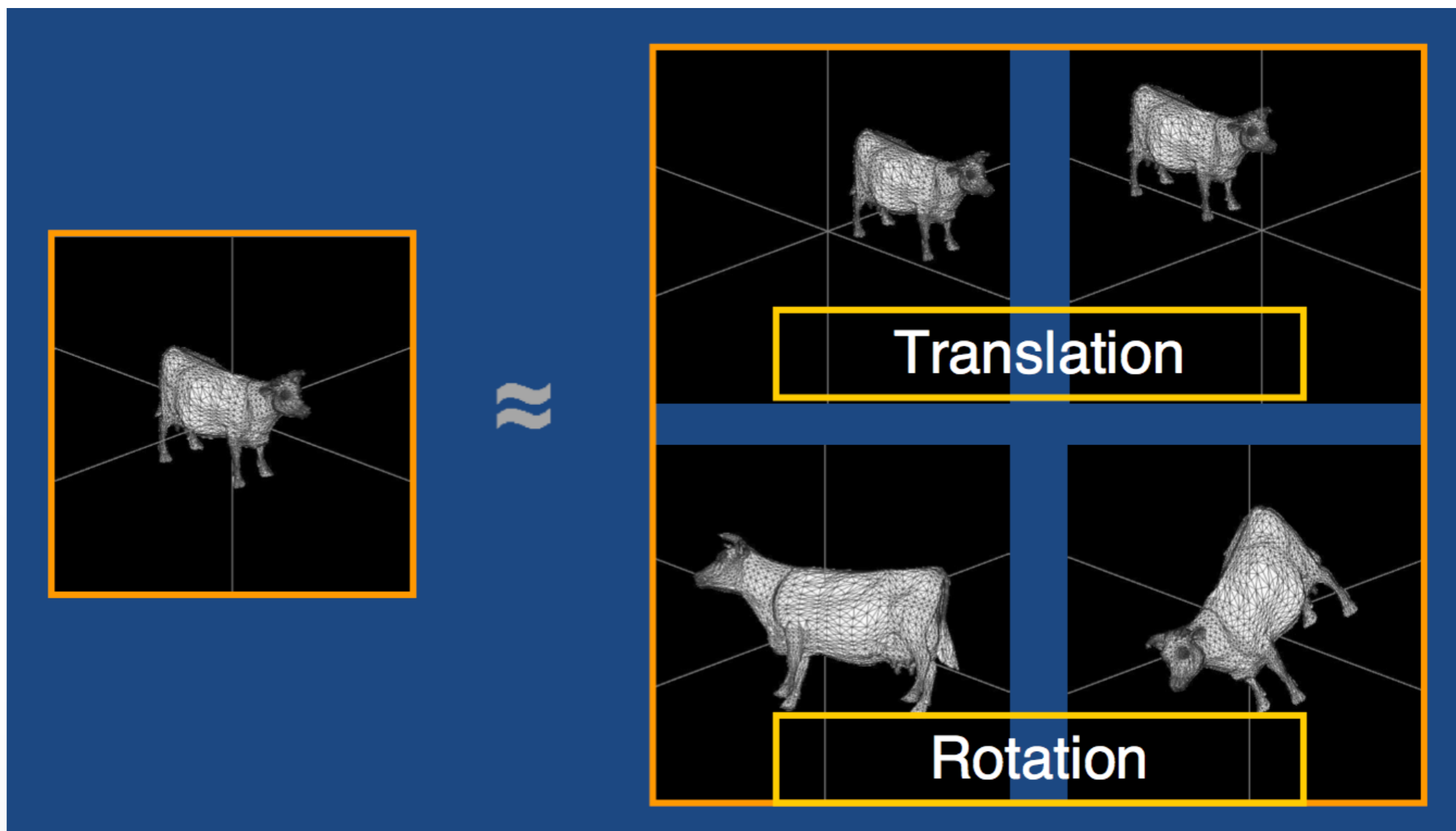


[Ankerst et al. 1999]



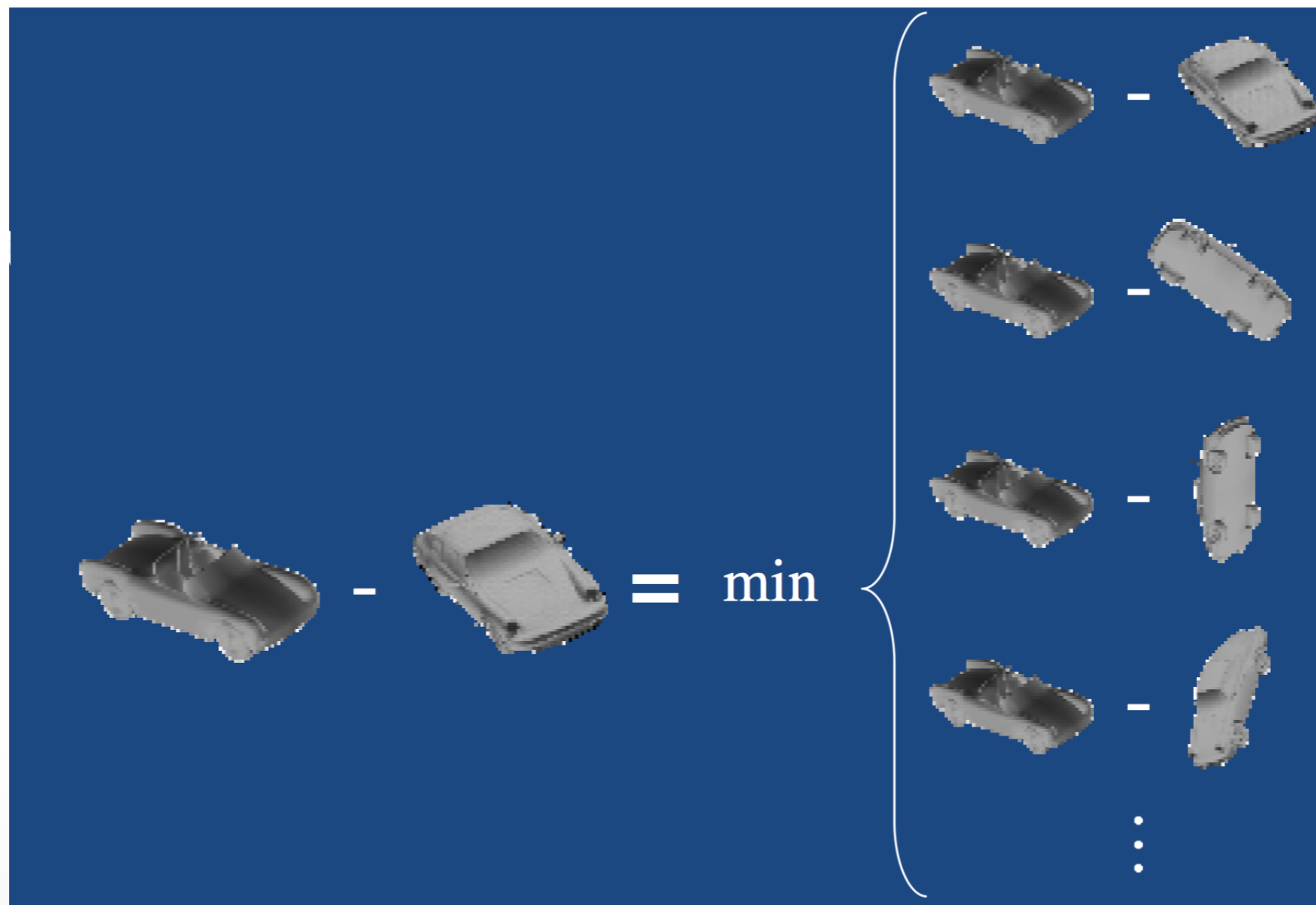
# Shape Descriptors: Challenge

- The **shape** of a model does not change when a rigid body transformation is applied to the model.



# Shape Descriptors: Challenge

- To compare two models, we need them at their optimal alignment



# Shape Descriptors: Alignment

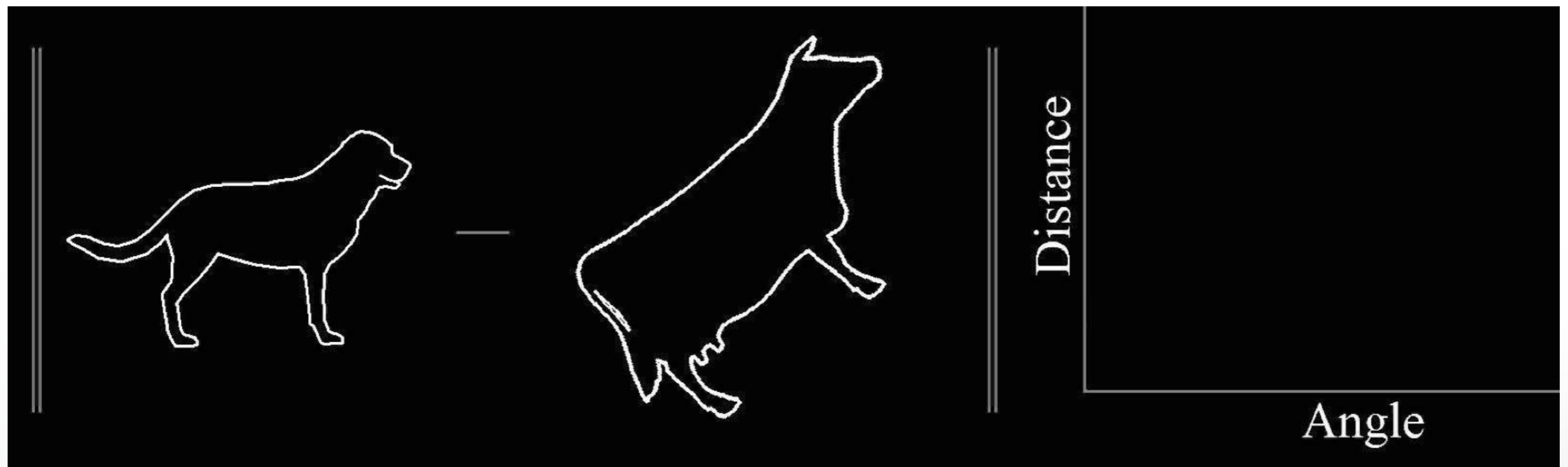
## Three general methods:

- Exhaustive Search
- Normalization
- Invariance

# Shape Descriptors: Alignment

## Exhaustive Search:

- Compare at all alignments

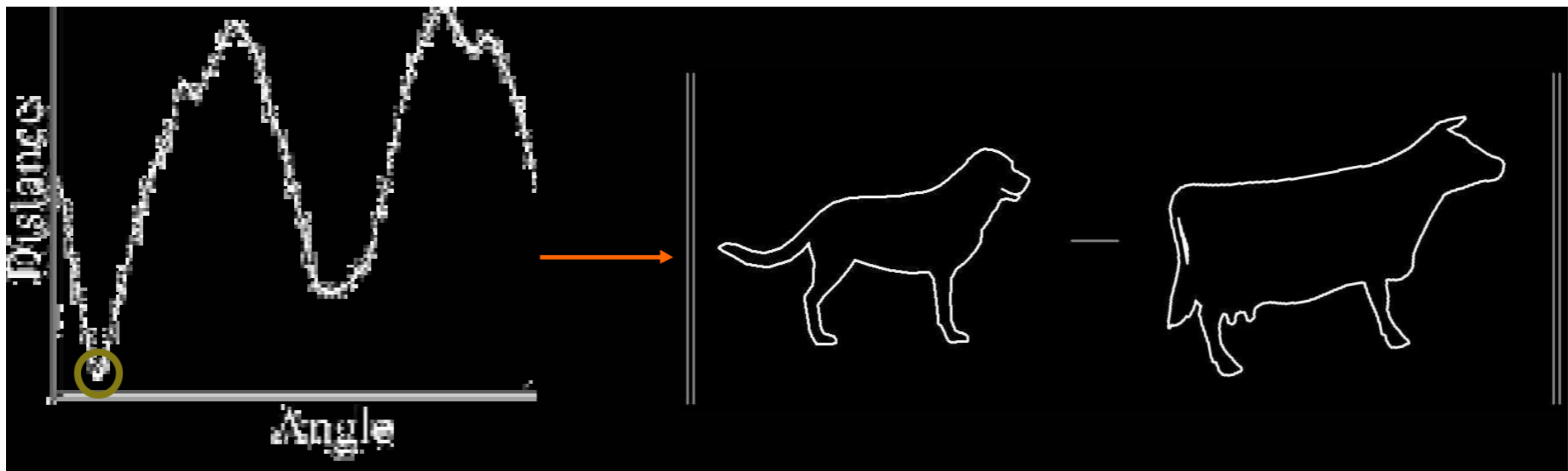


Exhaustive search for optimal rotation

# Shape Descriptors: Alignment

## Exhaustive Search:

- Compare at all alignments
- Correspondence is determined by the alignment at which the models are closest



Exhaustive search for optimal rotation

# Shape Descriptors: Alignment

## Exhaustive Search:

- Compare at all alignments
- Correspondence is determined by the alignment at which the models are closest

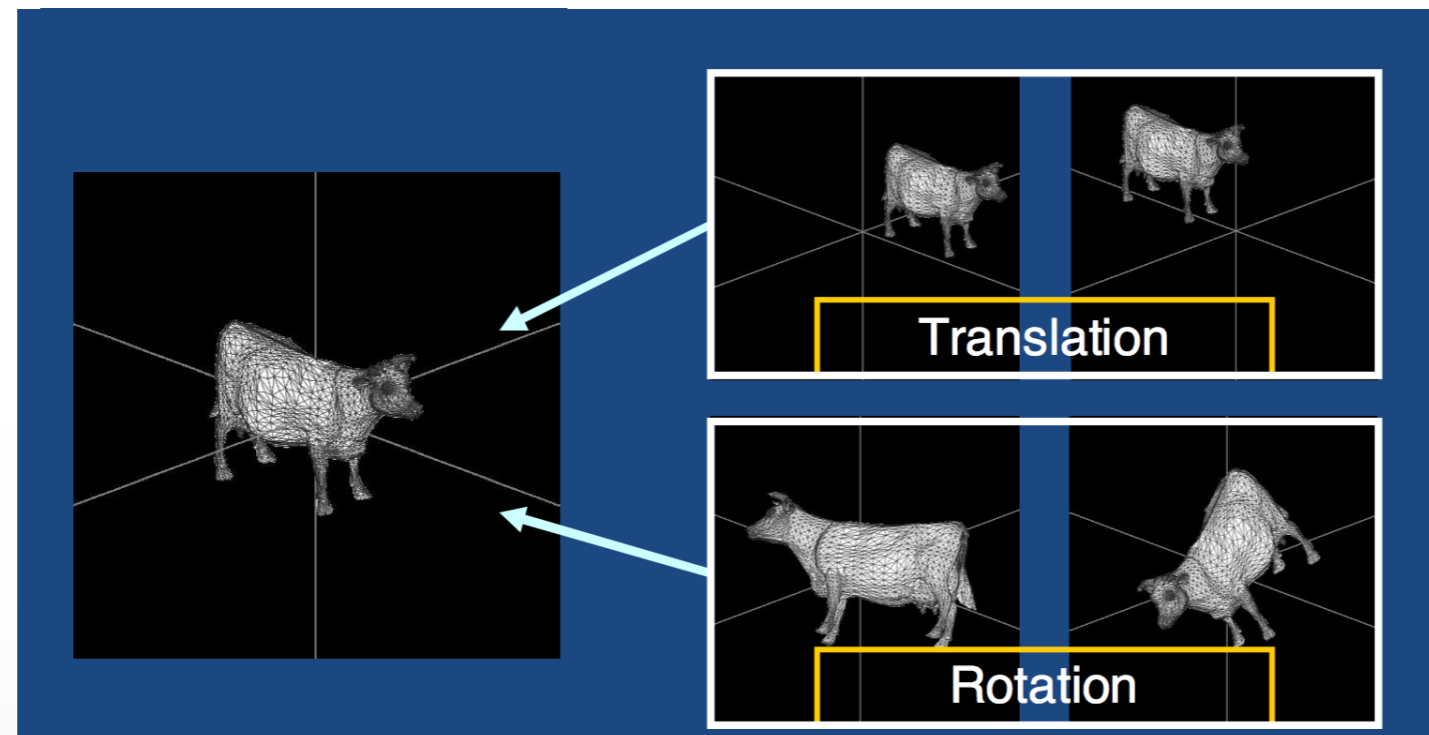
## Properties:

- Gives the correct answer (w.r.t. the metric)
- While slow on a single processor, it can be parallelized (Clusters? Multi-Threading? GPU?)

# Shape Descriptors: Alignment

## Normalization:

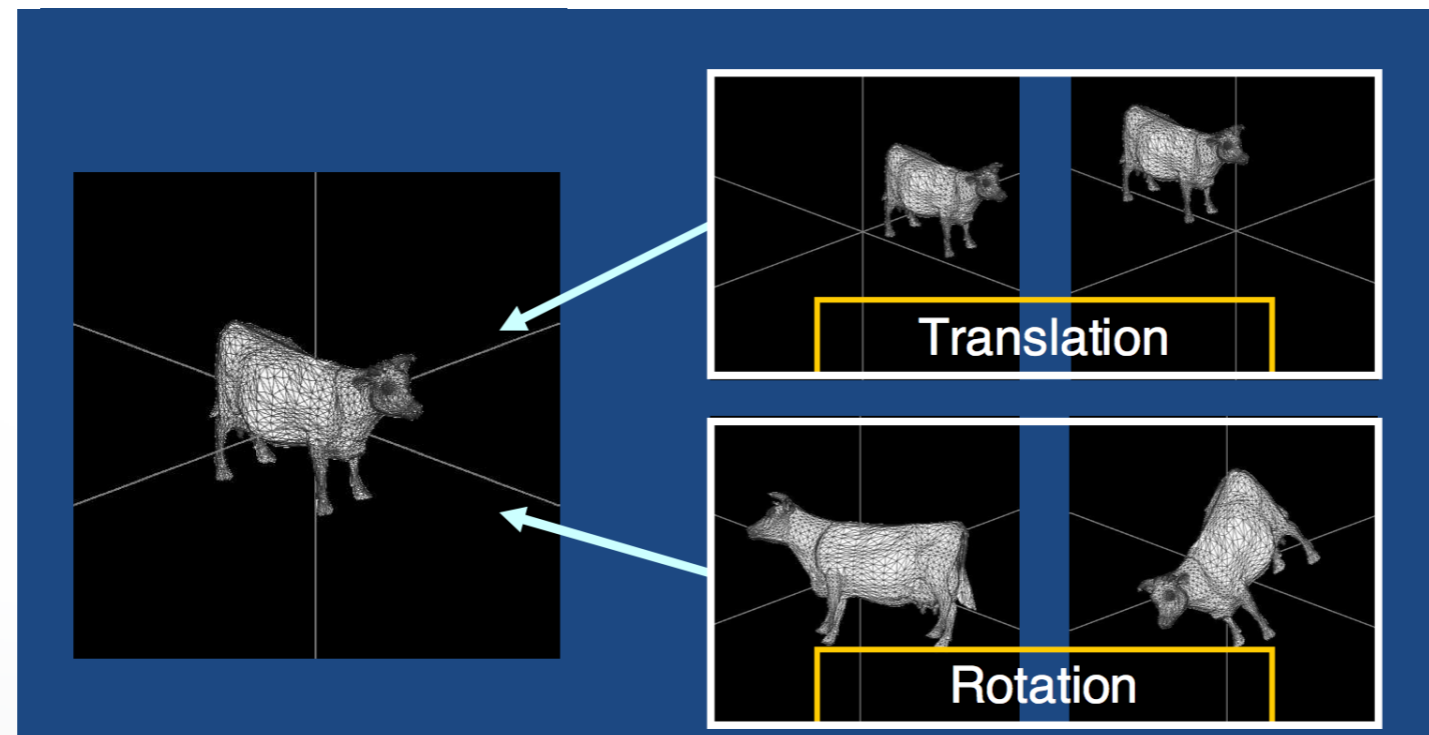
- Put each model into a canonical frame:
  - Translation
  - Rotation



# Shape Descriptors: Alignment

## Normalization:

- Put each model into a canonical frame:
  - Translation: Center of Mass
  - Rotation

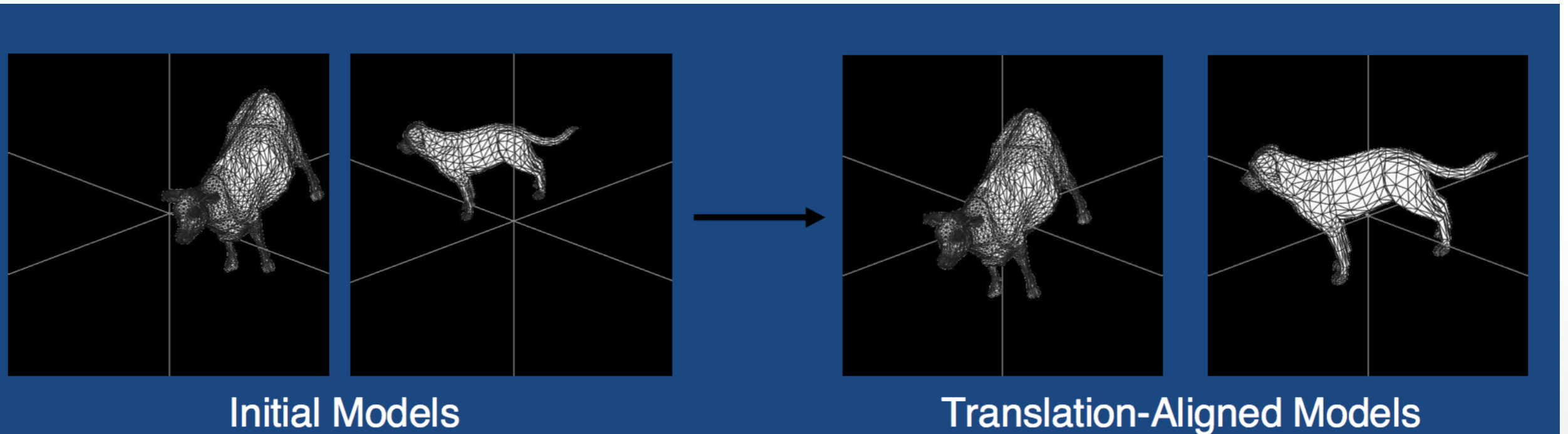




# Shape Descriptors: Alignment

## Normalization:

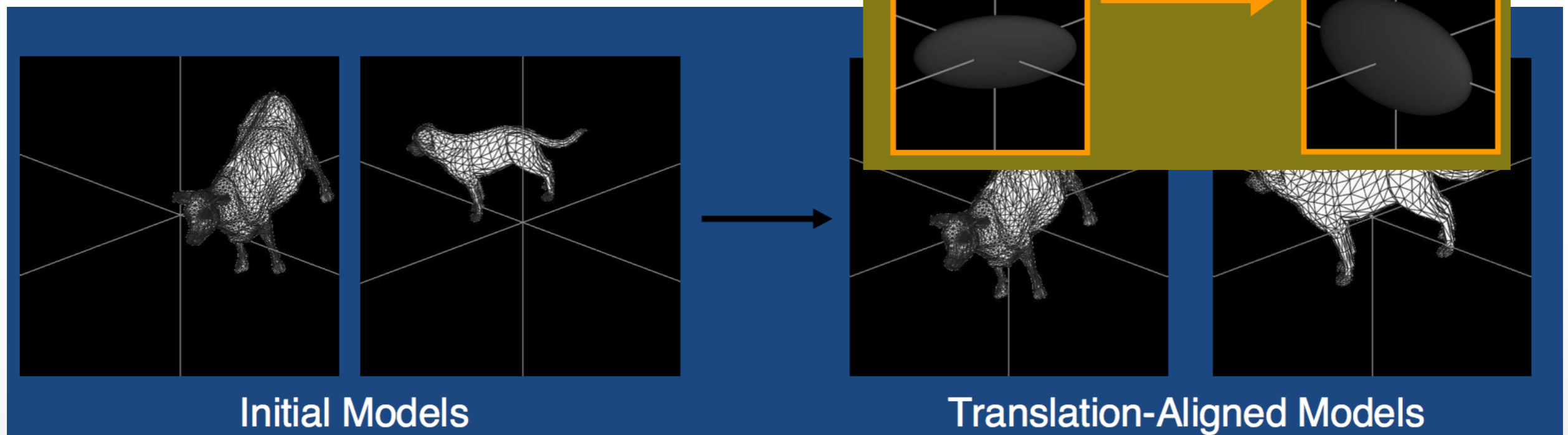
- Put each model into a canonical frame:
  - Translation: Center of Mass
  - Rotation



# Shape Descriptors: Alignment

## Normalization:

- Put each model into a canonical frame:
  - Translation: Center of Mass
  - Rotation: PCA alignment



# Shape Descriptors: Alignment

## Normalization:

- Put each model into a canonical frame:
  - Translation: Center of Mass
  - Rotation: PCA alignment

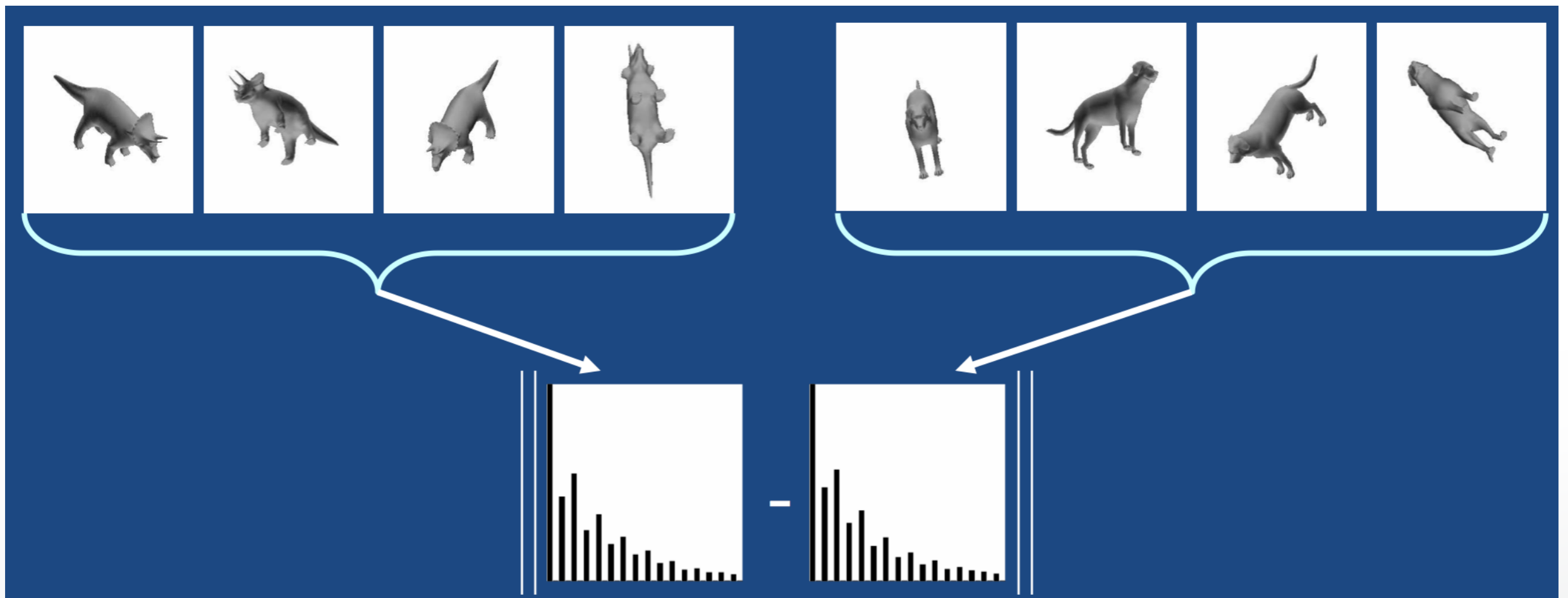
## Properties:

- Efficient
- Not always robust
- Not suitable for local feature matching

# Shape Descriptors: Alignment

## Invariance:

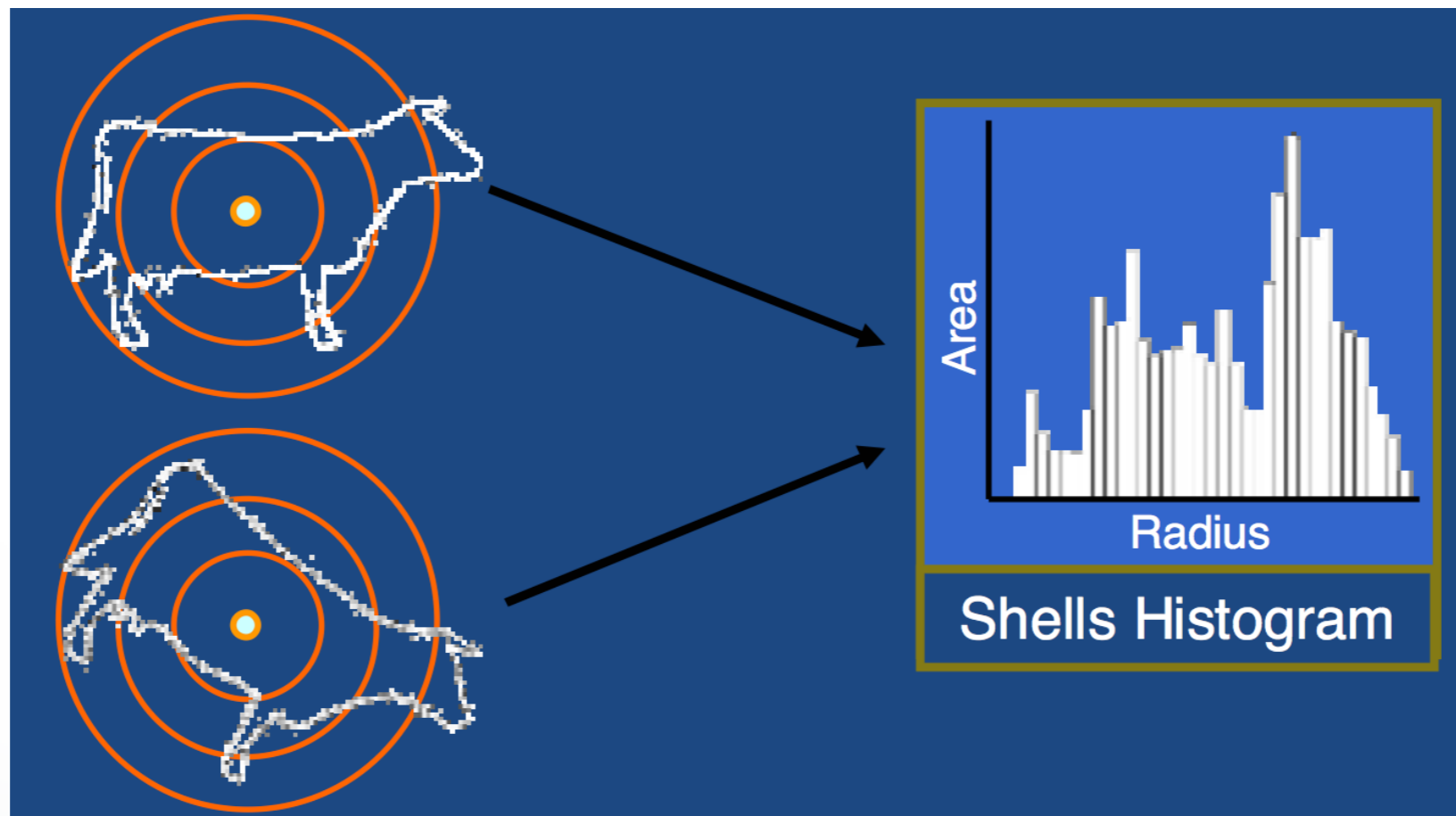
- Represent a model by a shape descriptor that is independent of the pose.



# Shape Descriptors: Alignment

## Example: Ankerst's Shells

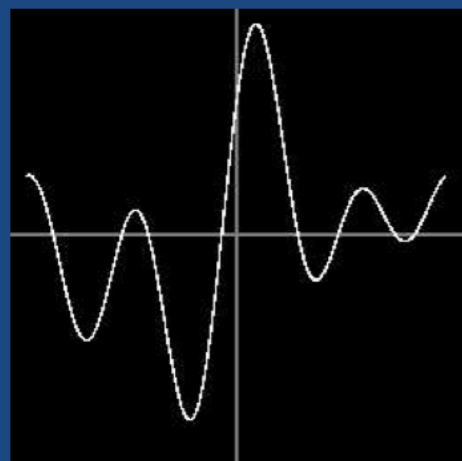
- A histogram of the radial distribution of surface area



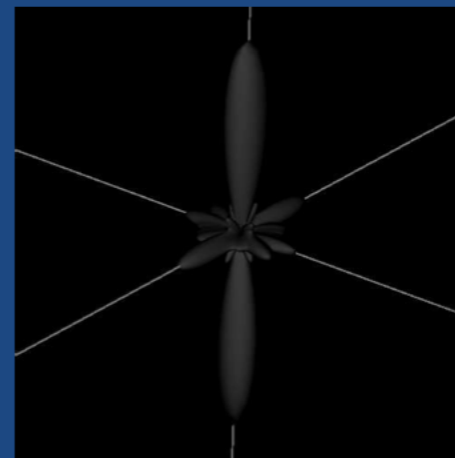
# Shape Descriptors: Alignment

## Invariance

- Power spectrum representation
  - Fourier transform for translations
  - Spherical harmonic transform for rotations

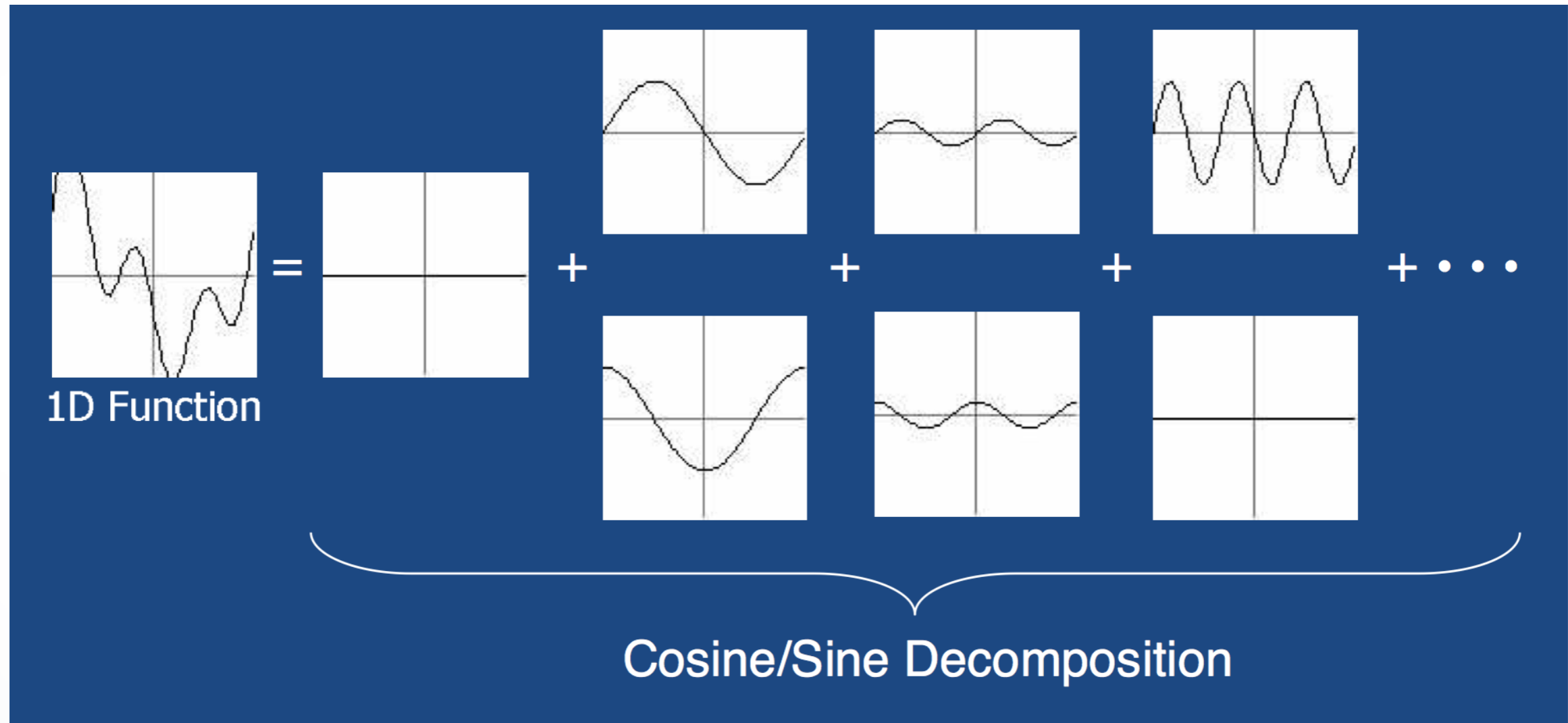


Circular Power Spectrum

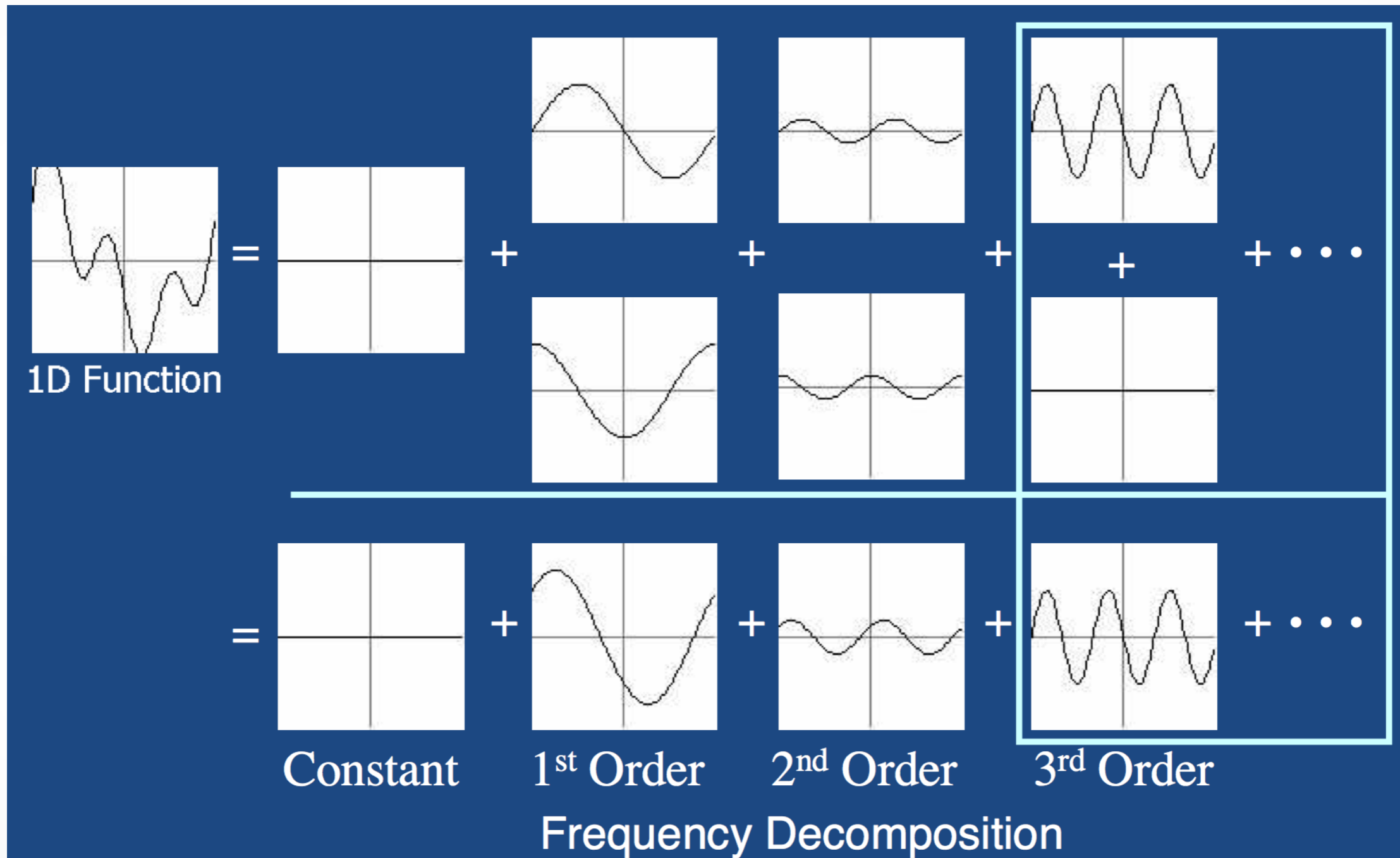


Spherical Power Spectrum

# Translation Invariance



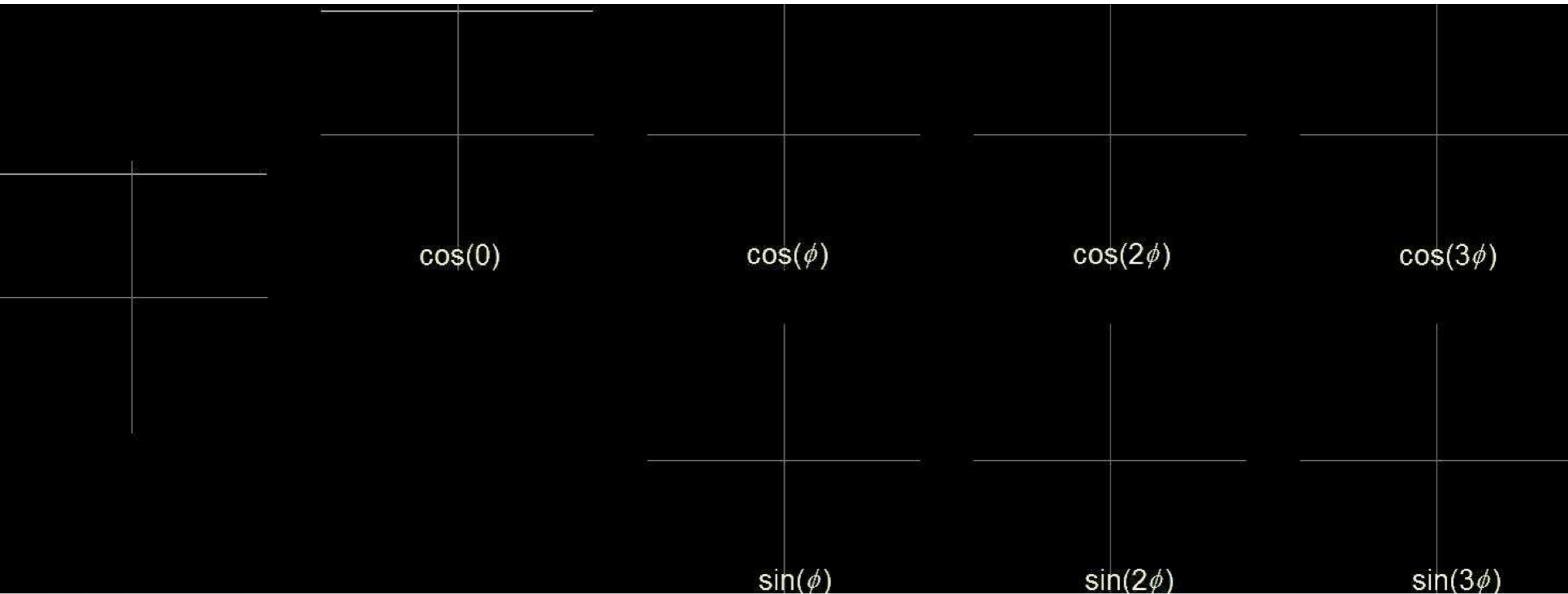
# Translation Invariance





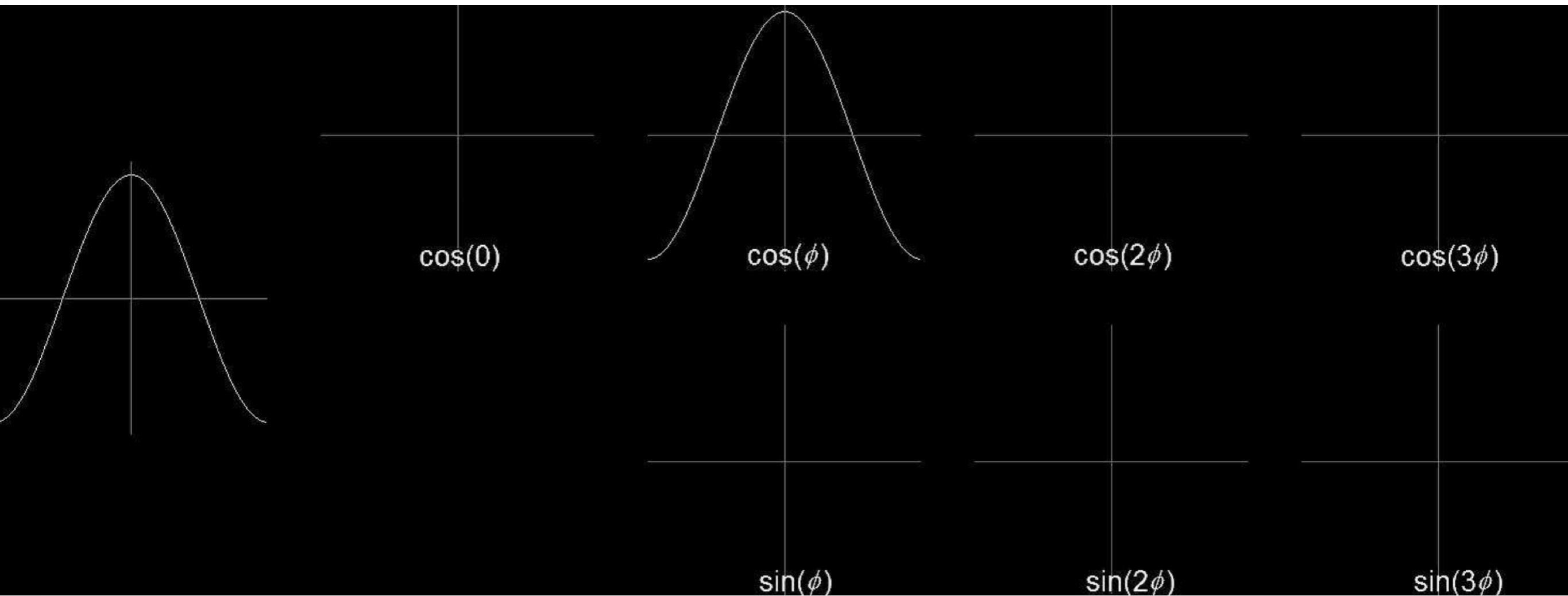
# Translation Invariance

Frequency subspaces are fixed by rotations:



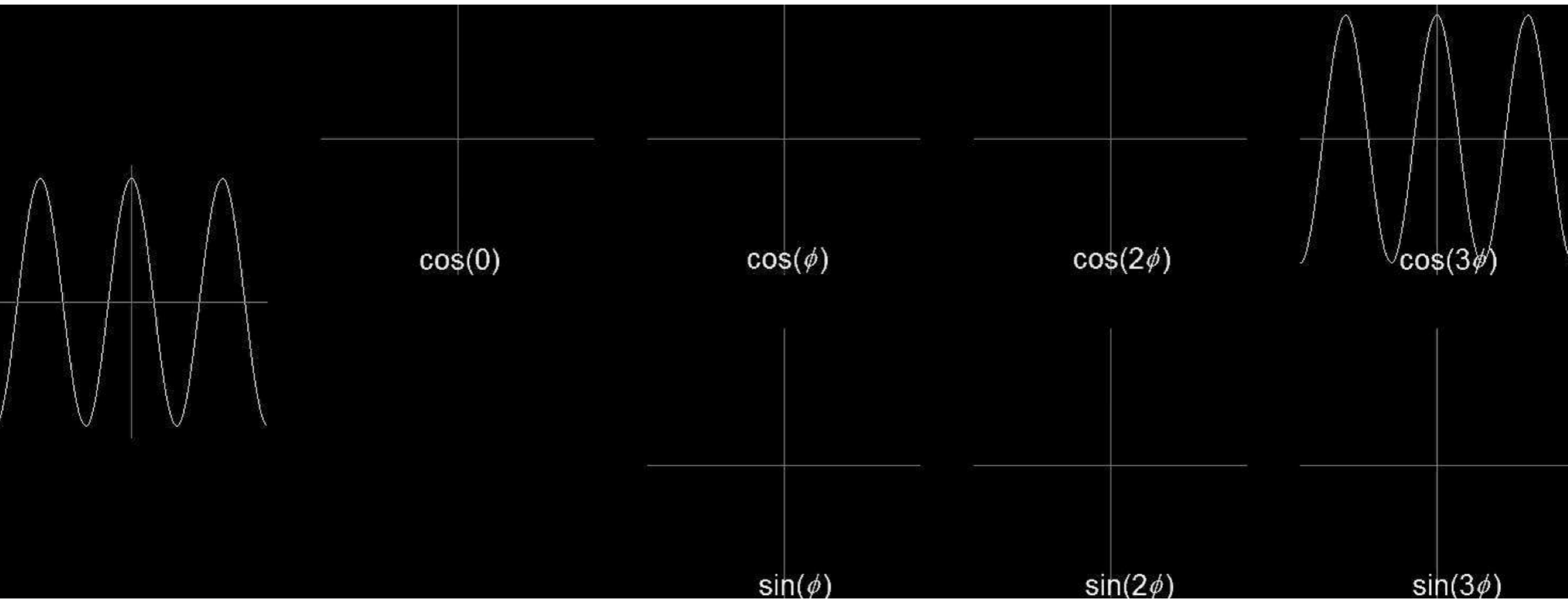
# Translation Invariance

Frequency subspaces are fixed by rotations:

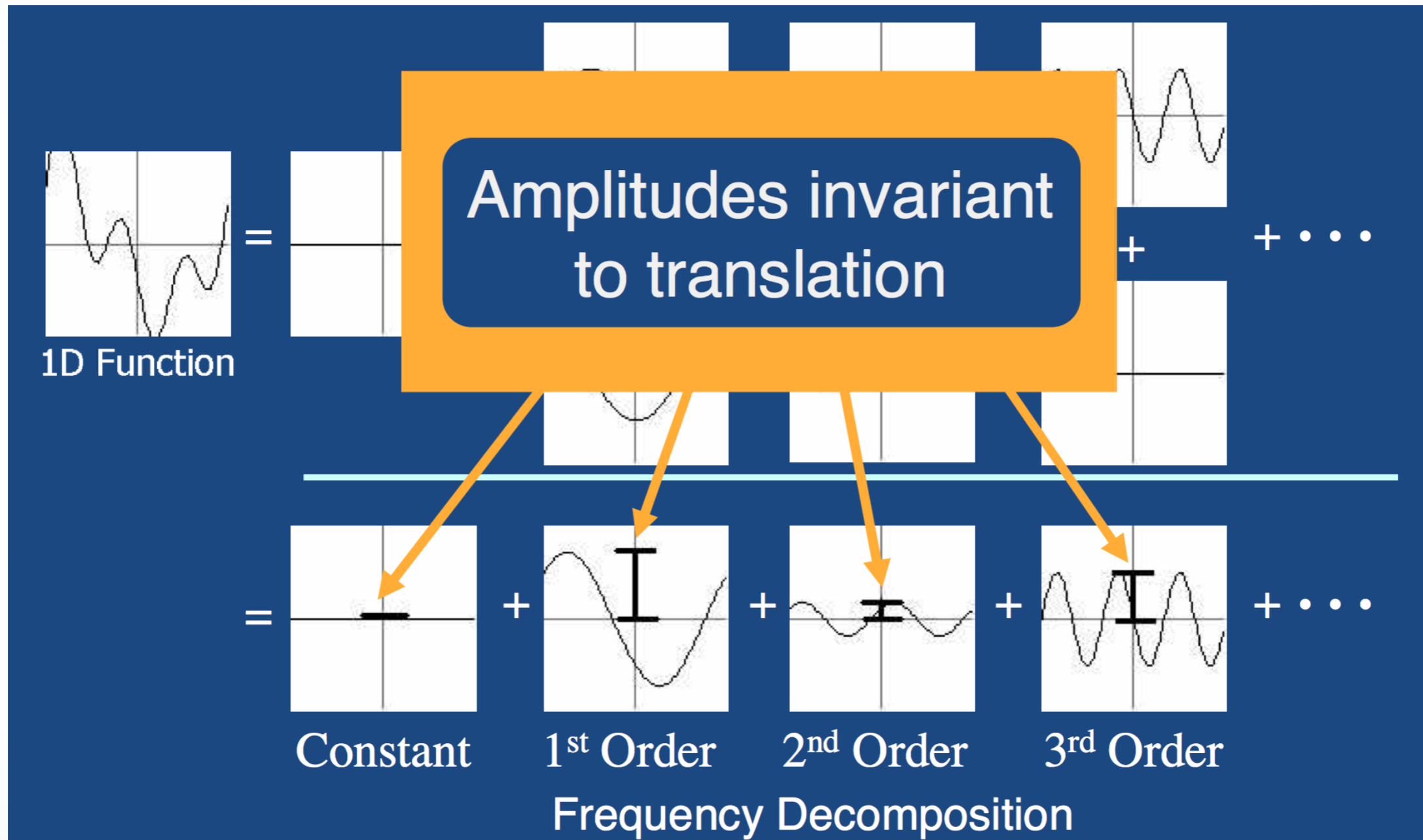


# Translation Invariance

Frequency subspaces are fixed by translations:

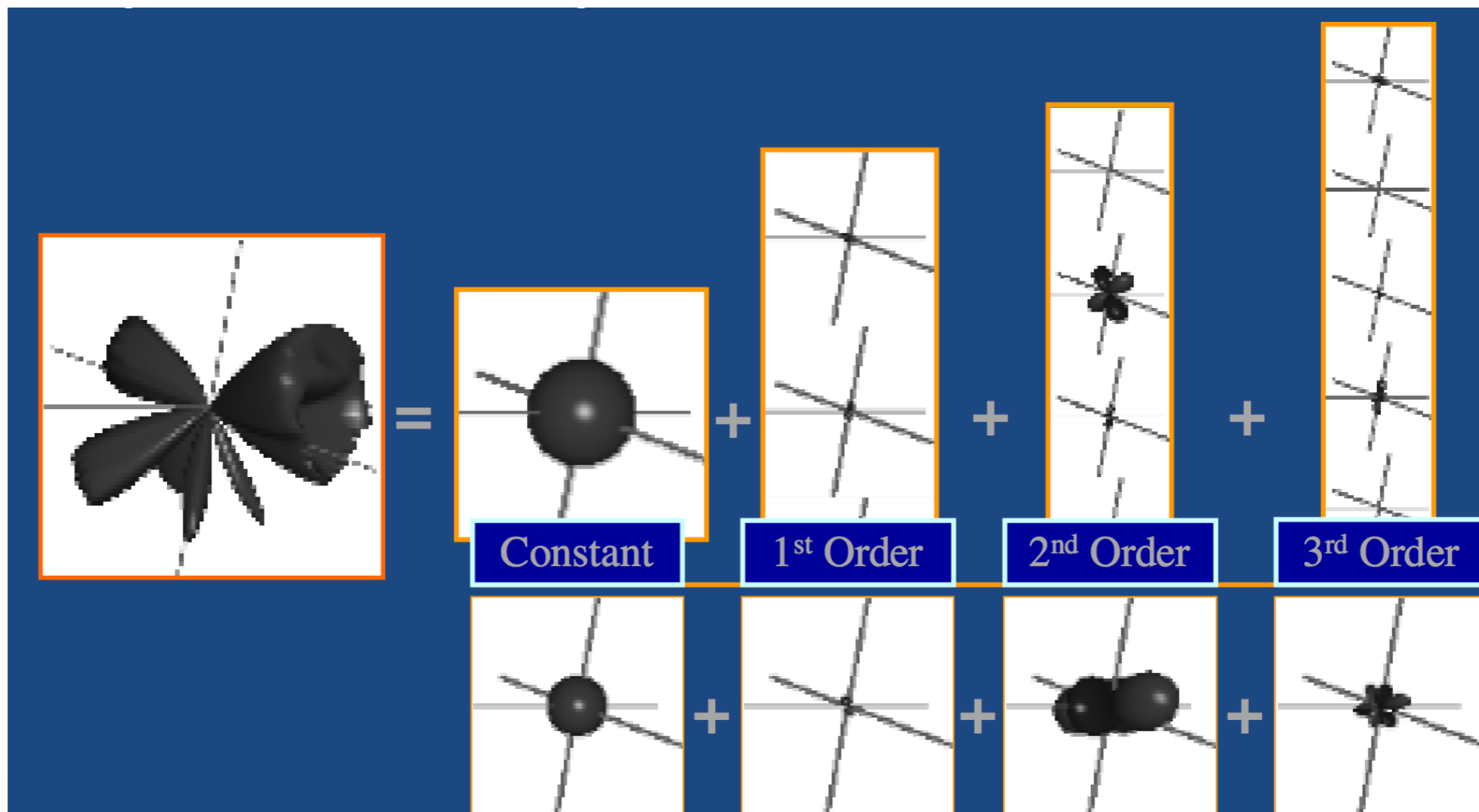


# Translation Invariance



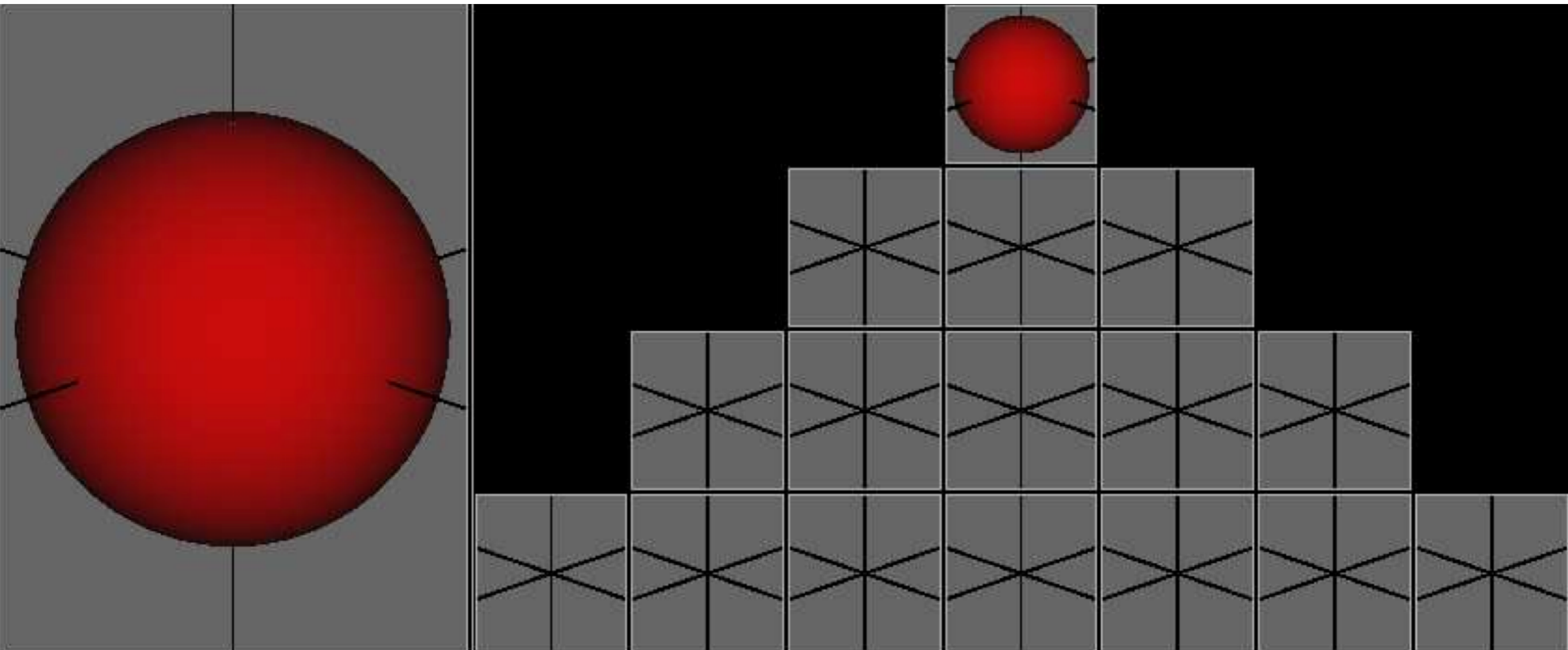
# Rotation Invariance

Represent each spherical function as a sum of harmonic frequencies (orders)



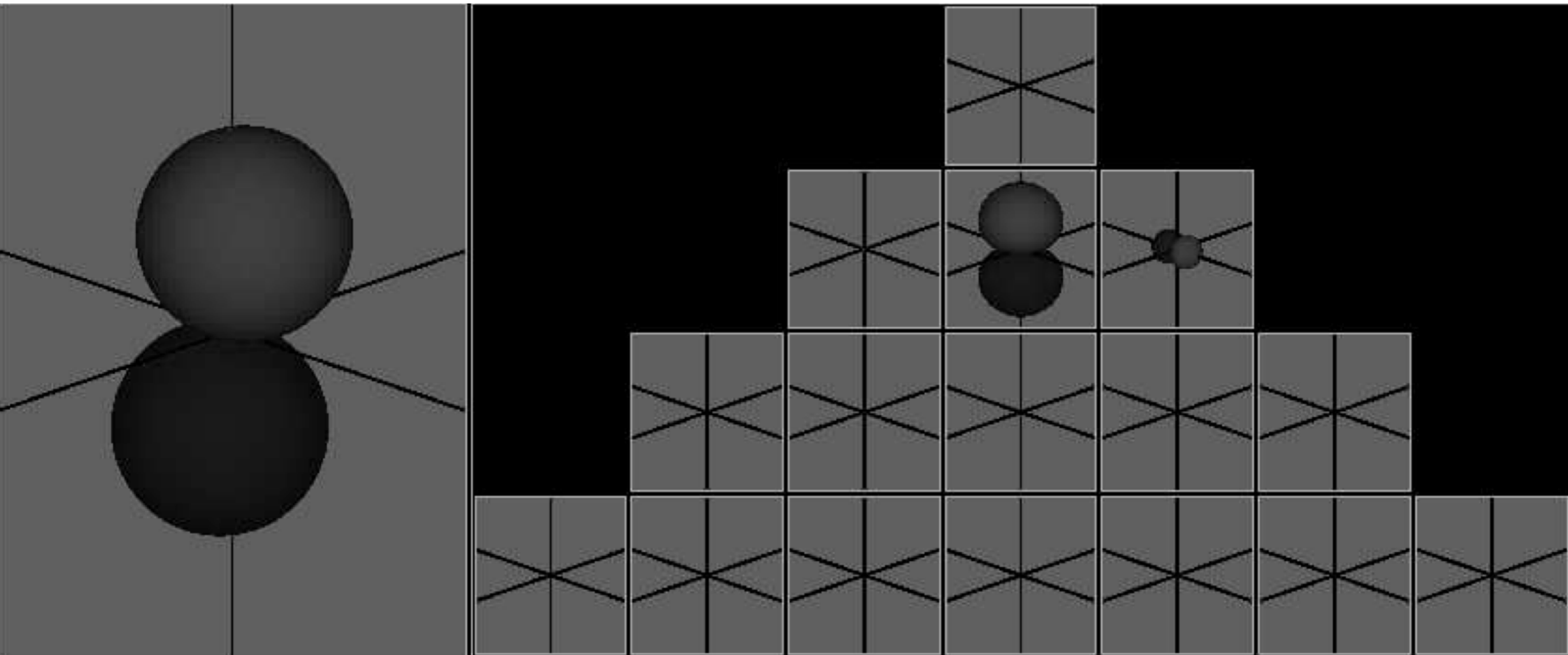
# Rotation Invariance

Frequency subspaces are fixed by rotations



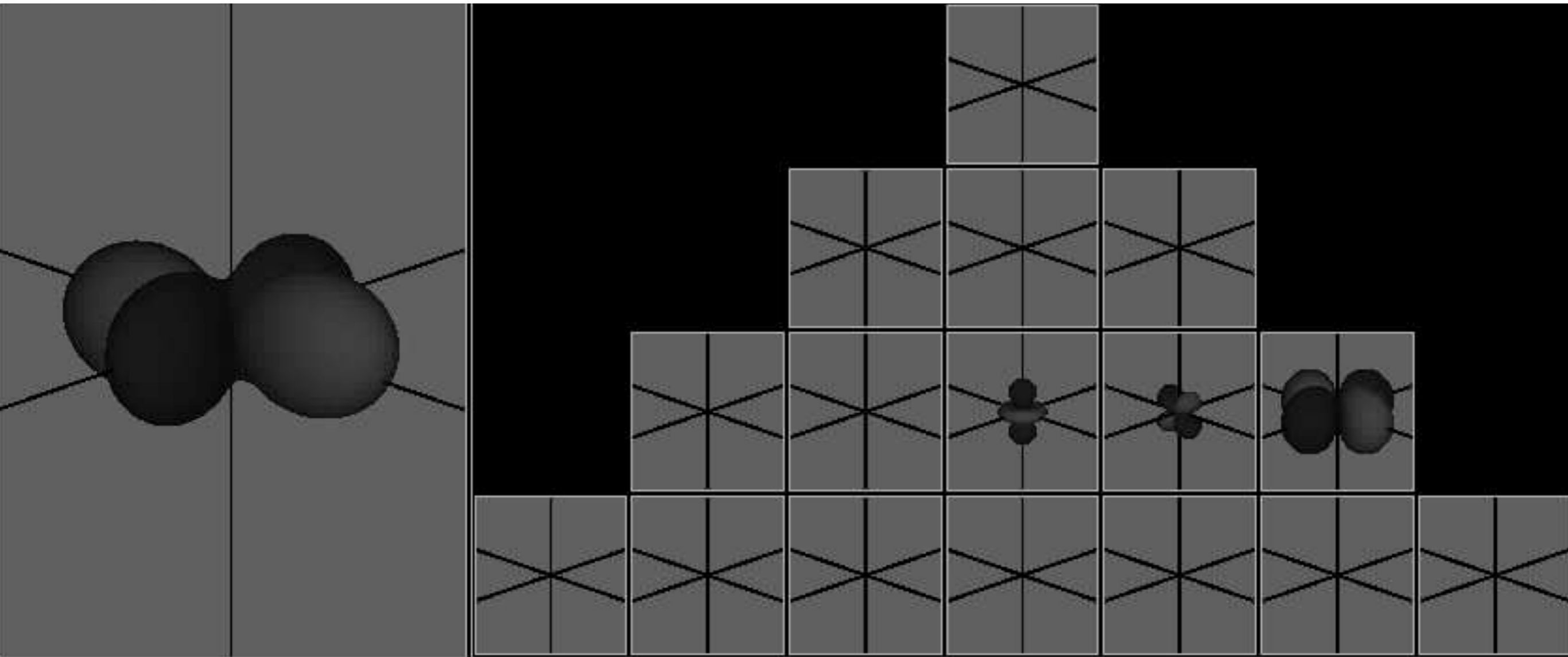
# Rotation Invariance

Frequency subspaces are fixed by rotations



# Rotation Invariance

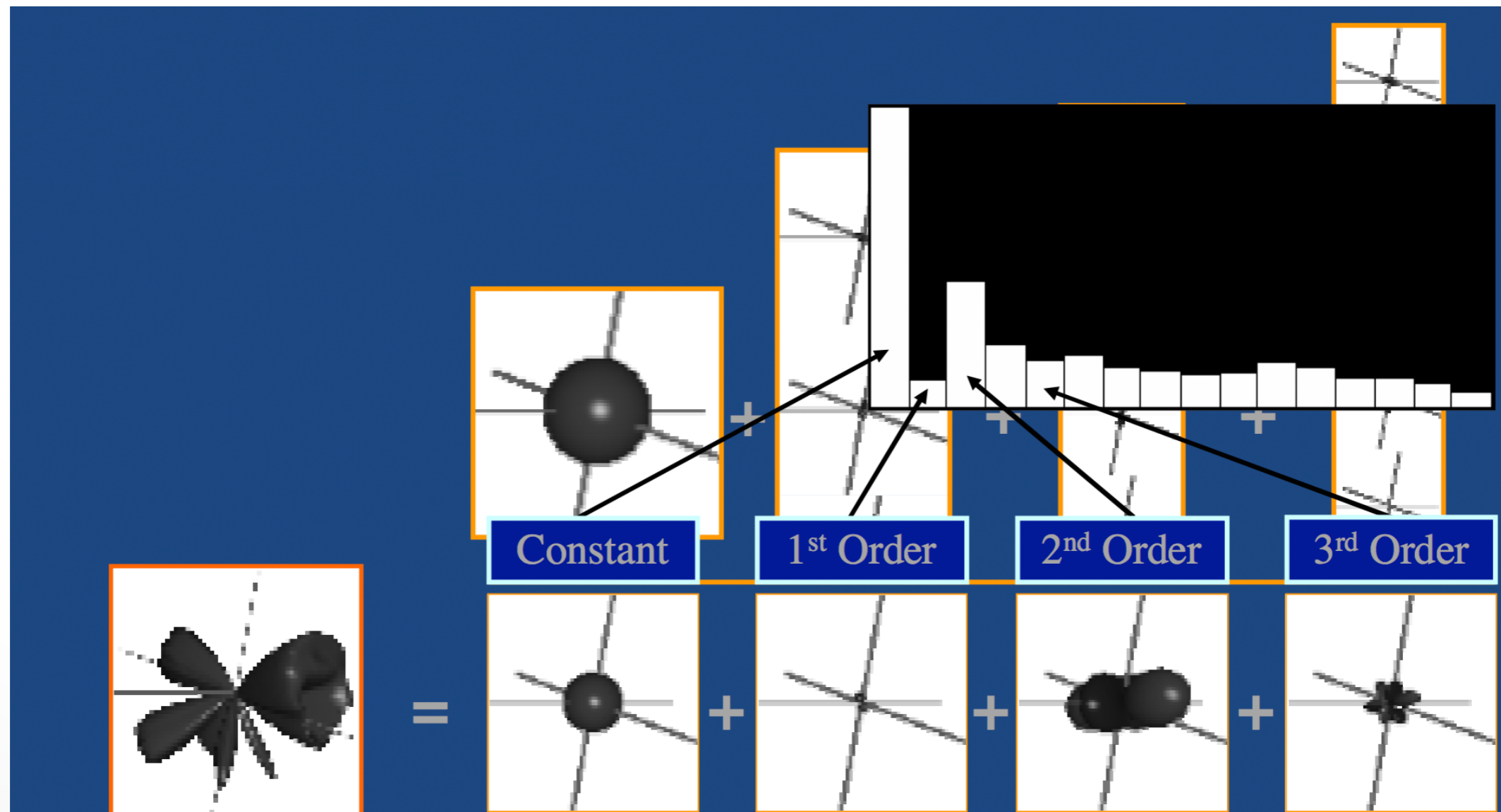
Frequency subspaces are fixed by rotations





# Rotation Invariance

Store “how much” (L2-norm) of the shape resides in each frequency to get a rotation invariant representation



# Shape Descriptors: Alignment

## Invariance:

- Represent a model by a shape descriptor that is independent of the pose

## Properties:

- Compact representation
- Not always discriminating

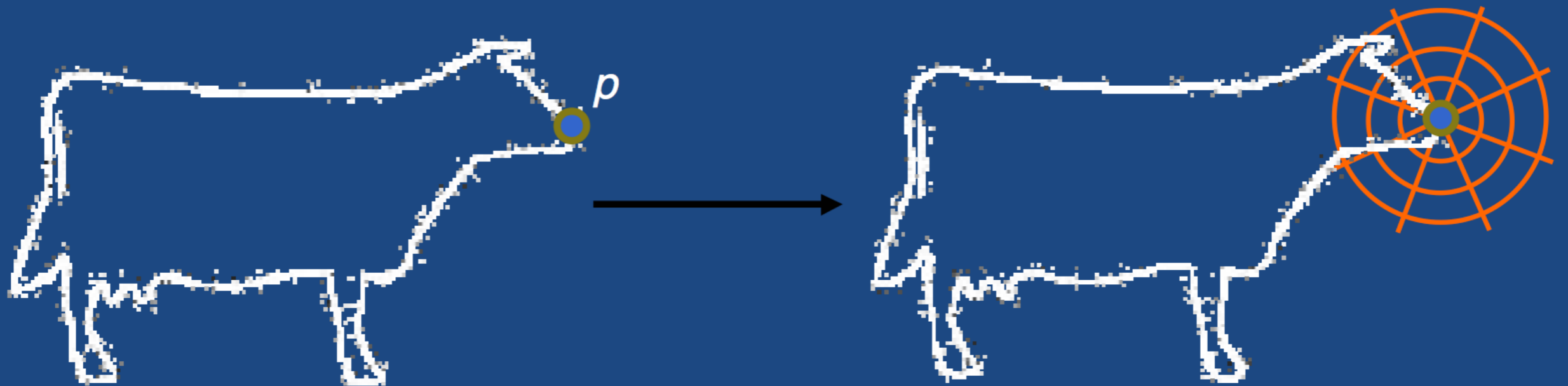
# Outline

- **Global Shape Correspondence**
  - Shape Descriptors
  - Alignment
- **Partial Shape Correspondence**
  - From Global to Local
  - Pose Normalization
  - Partial Shape Descriptors
- **Registration**
  - Closed Form Solutions
  - Branch & Bound
  - Random Sample Consensus (RANSAC)

# From Global to Local

To characterize the surface about a point  $p$ , take global descriptor and:

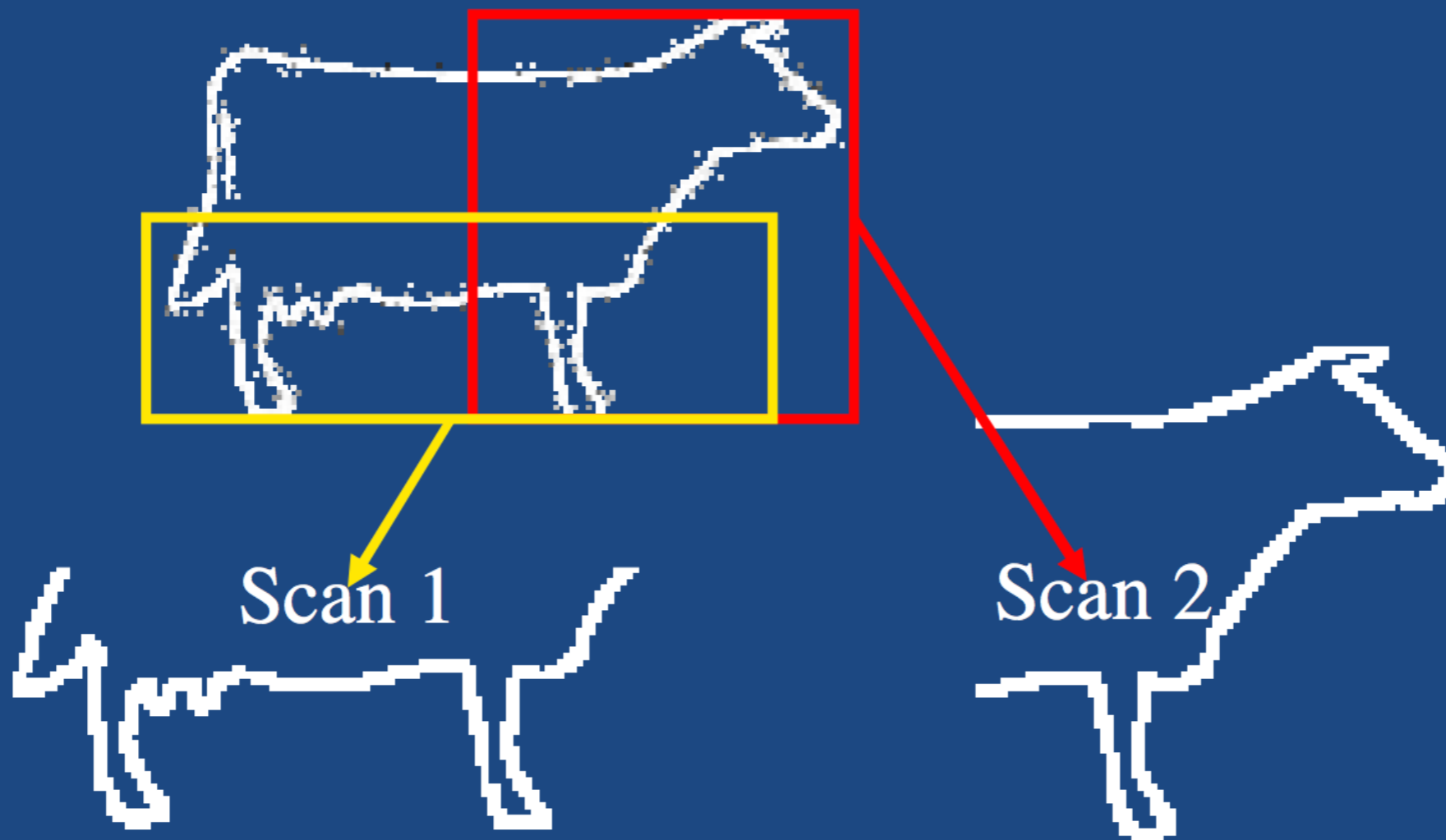
- center it about  $p$  (instead of center of mass), and
- restrict the extent to a small region about  $p$



Shape histograms as local shape descriptors

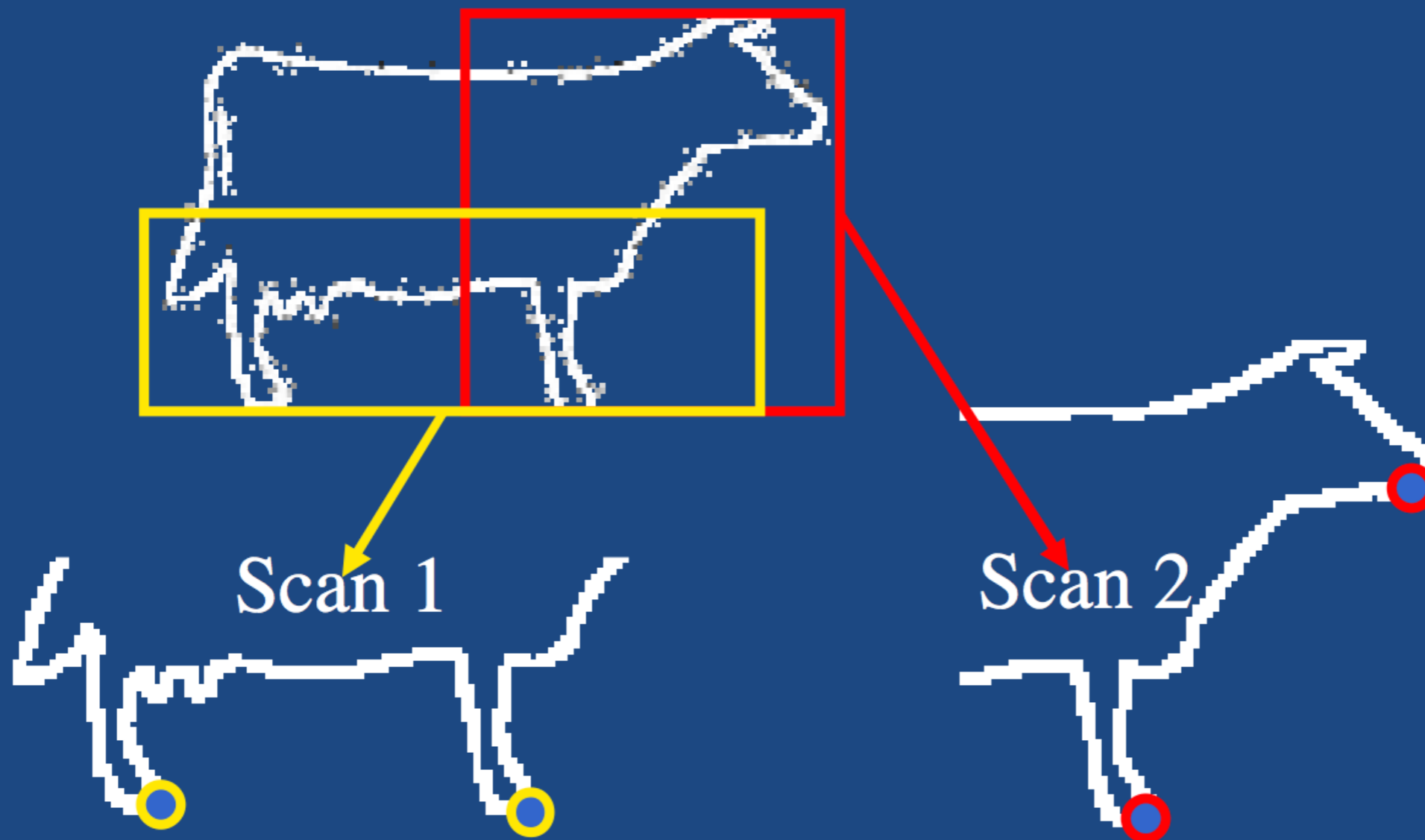
# From Global to Local

Given scans of a model:



# From Global to Local

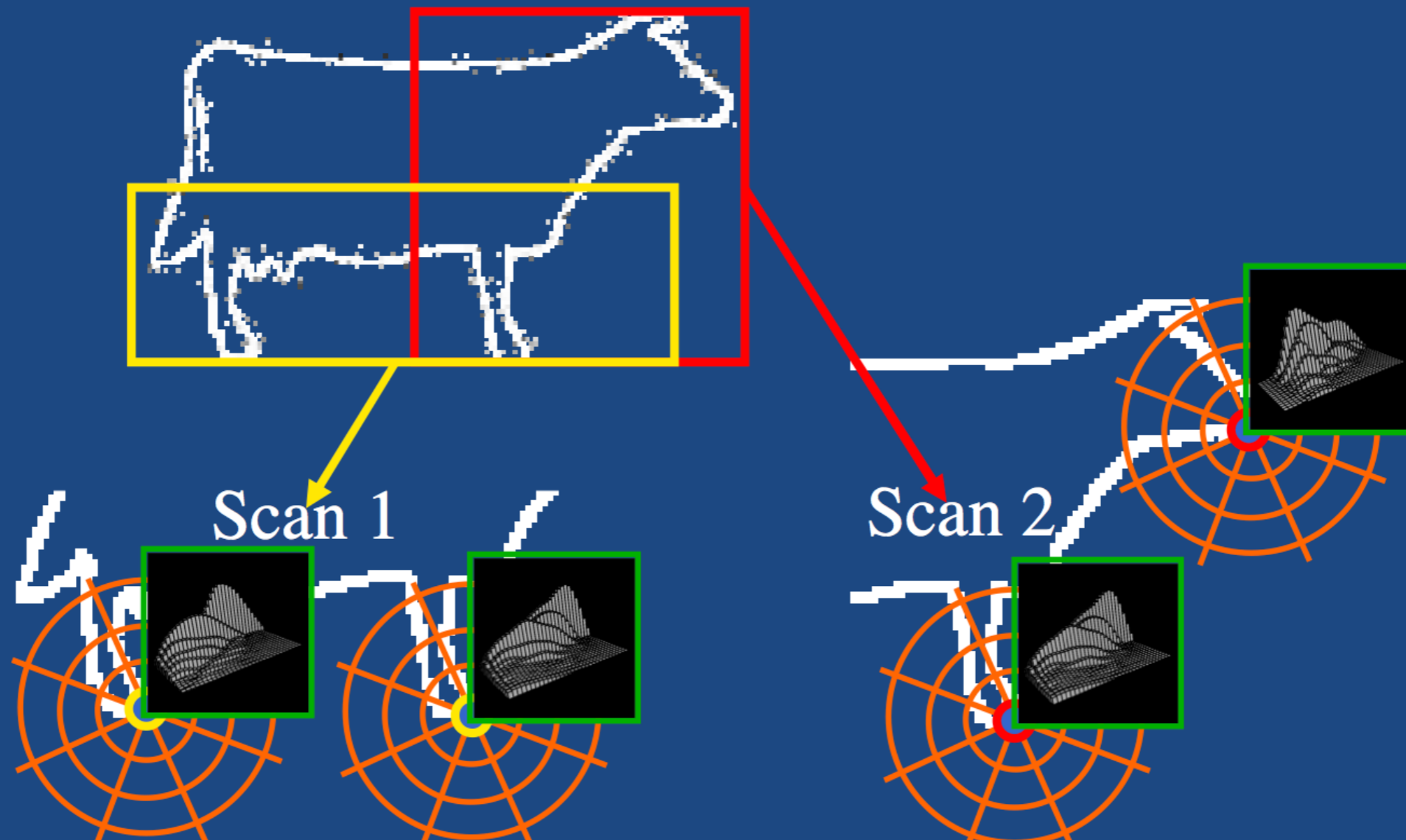
Identify the features



# From Global to Local

Identify the features

Computer a local descriptor for each feature

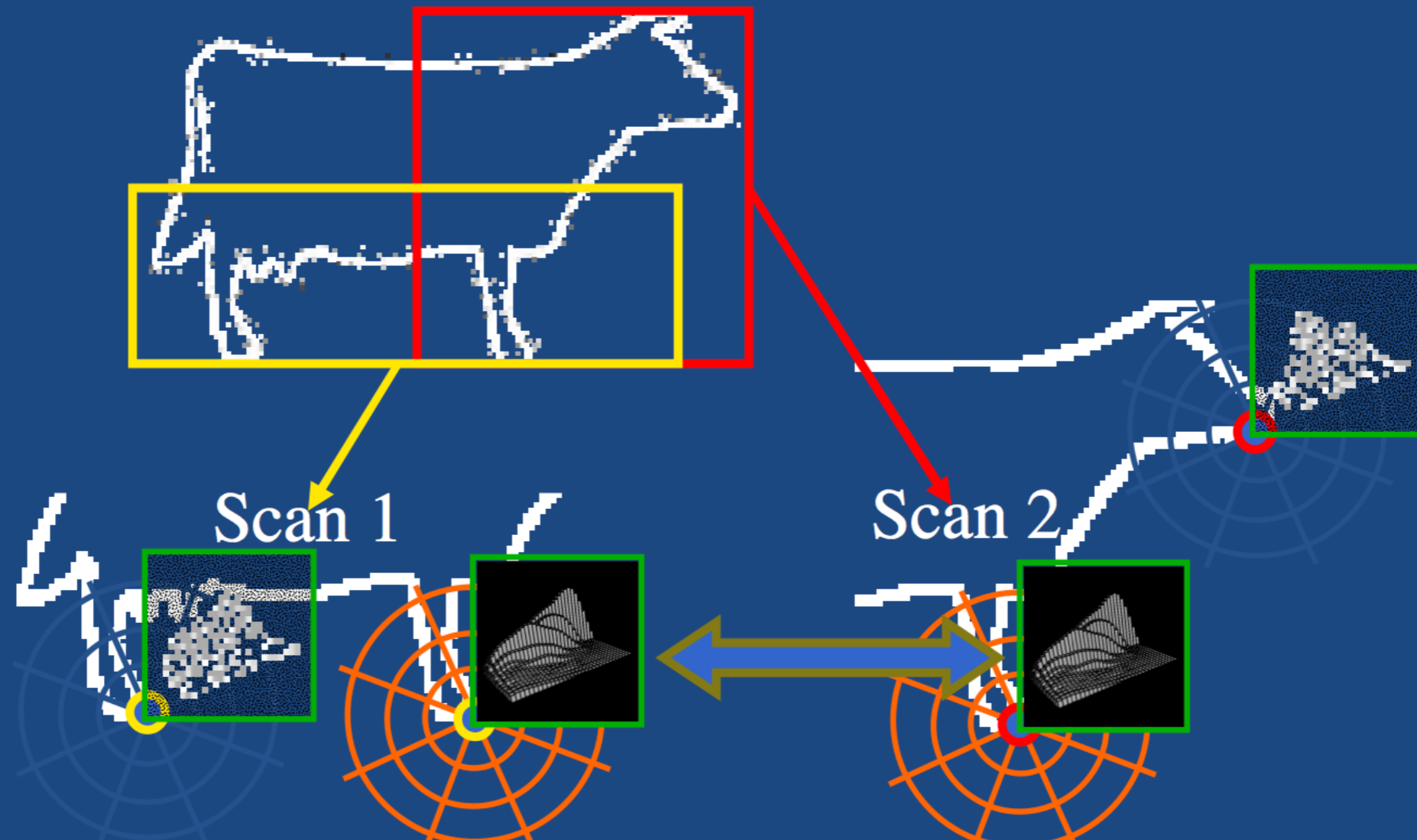


# From Global to Local

Identify the features

Computer a local descriptor for each feature

Feature correspond  $\rightarrow$  descriptors are similar

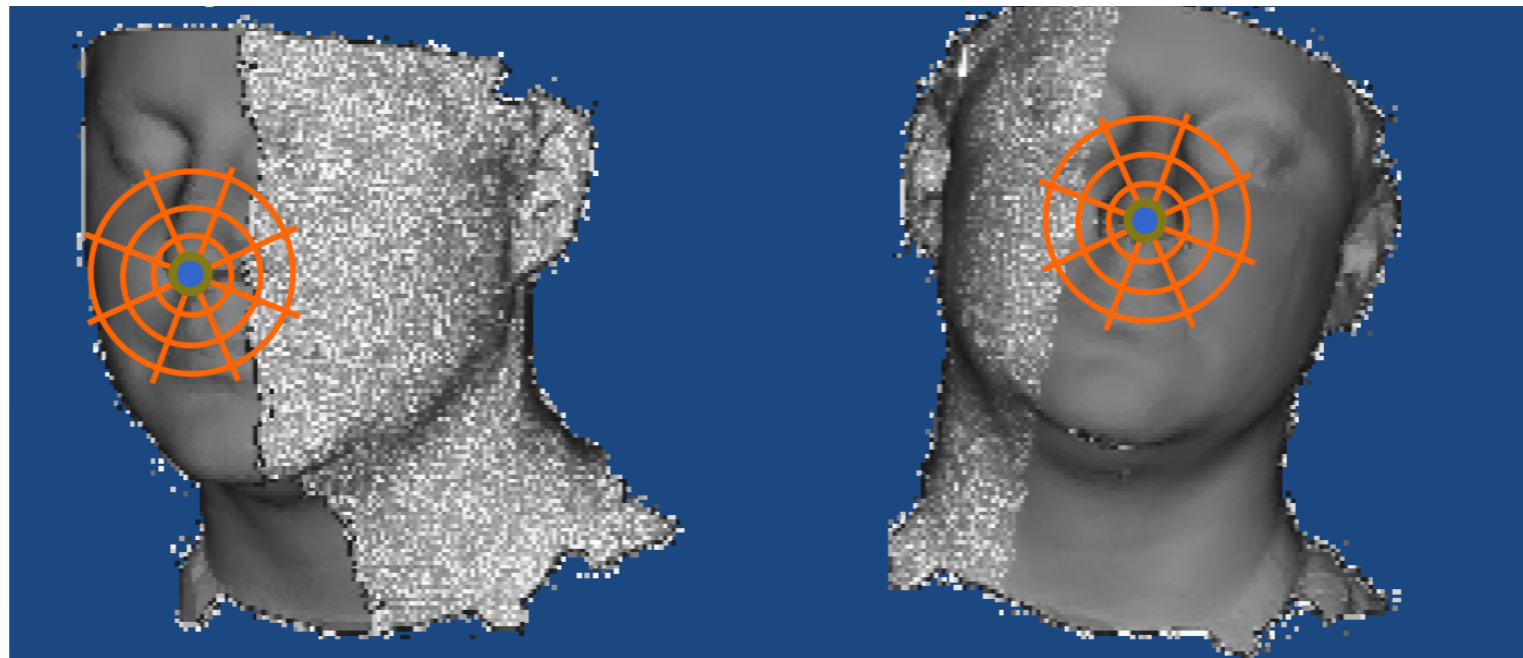




# Pose Normalization

## From Global to Local

- Translation: Accounted for by centering the descriptor at the point of interest.
- Rotation: We still need to be able to match descriptors across different rotations.



# Pose Normalization

## Challenge

- Since only parts of the models are given, we cannot use global normalization to align the local descriptors

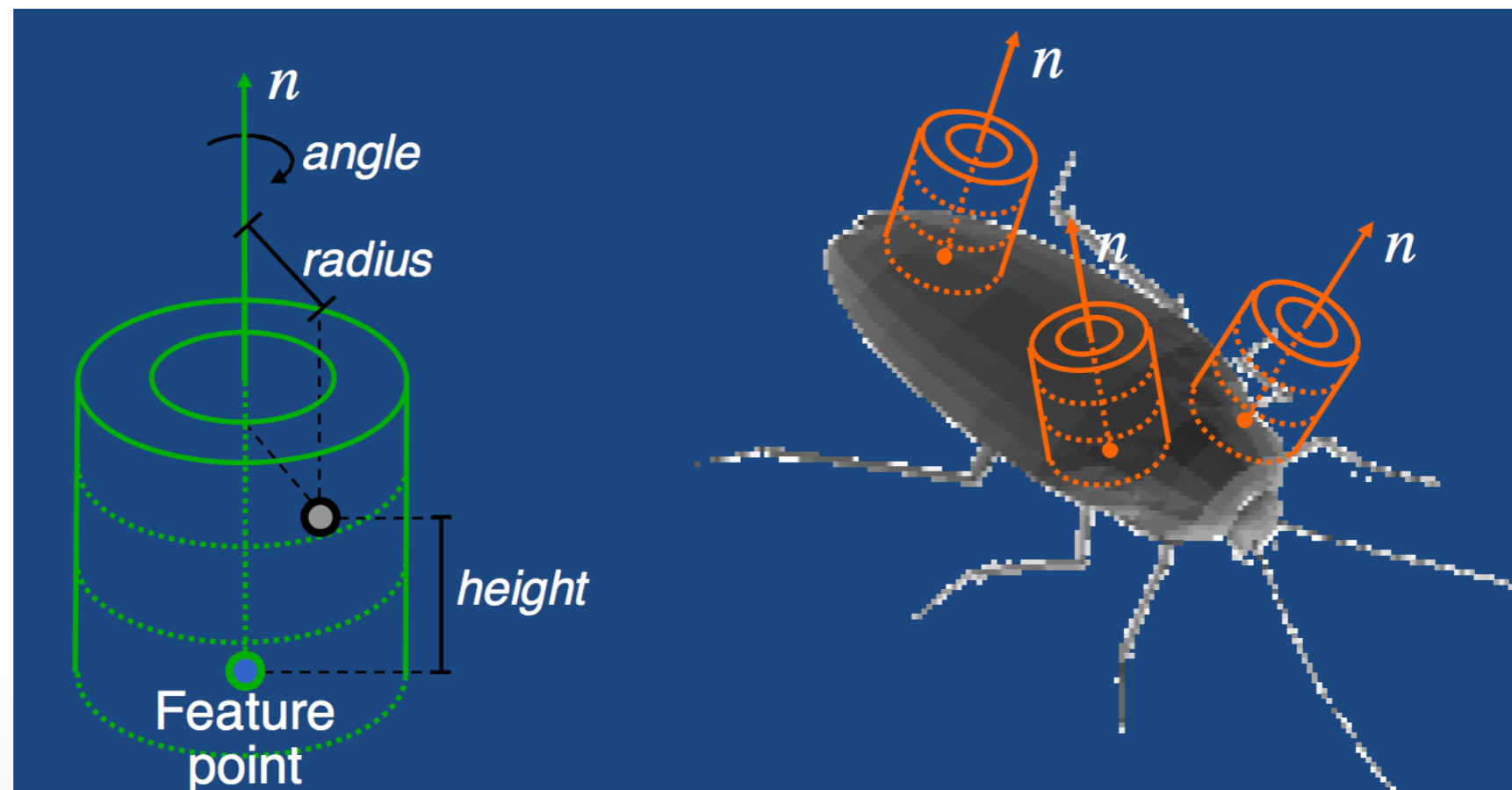
## Solutions

- Normalize using **local** information

# Local Descriptors: Examples

## Variations of Shape Histograms

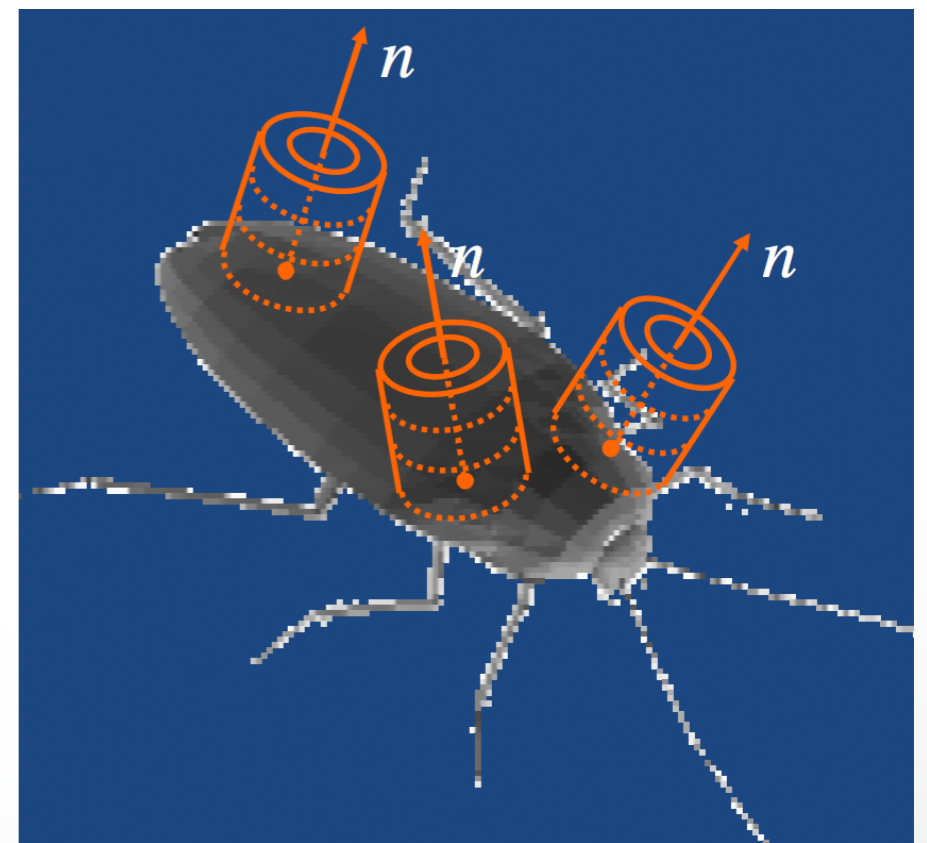
- For each feature, represent its local geometry in cylindrical coordinates about the normal



# Local Descriptors: Examples

## Variations of Shape Histograms

- For each feature, represent its local geometry in cylindrical coordinates about the normal
  - **Spin Images:** Store energy in each normal ring
  - **Harmonic Shape Contexts:** Store power spectrum of each normal ring
  - **3D Shape Contexts:** Search over all rotations about the normal for best match



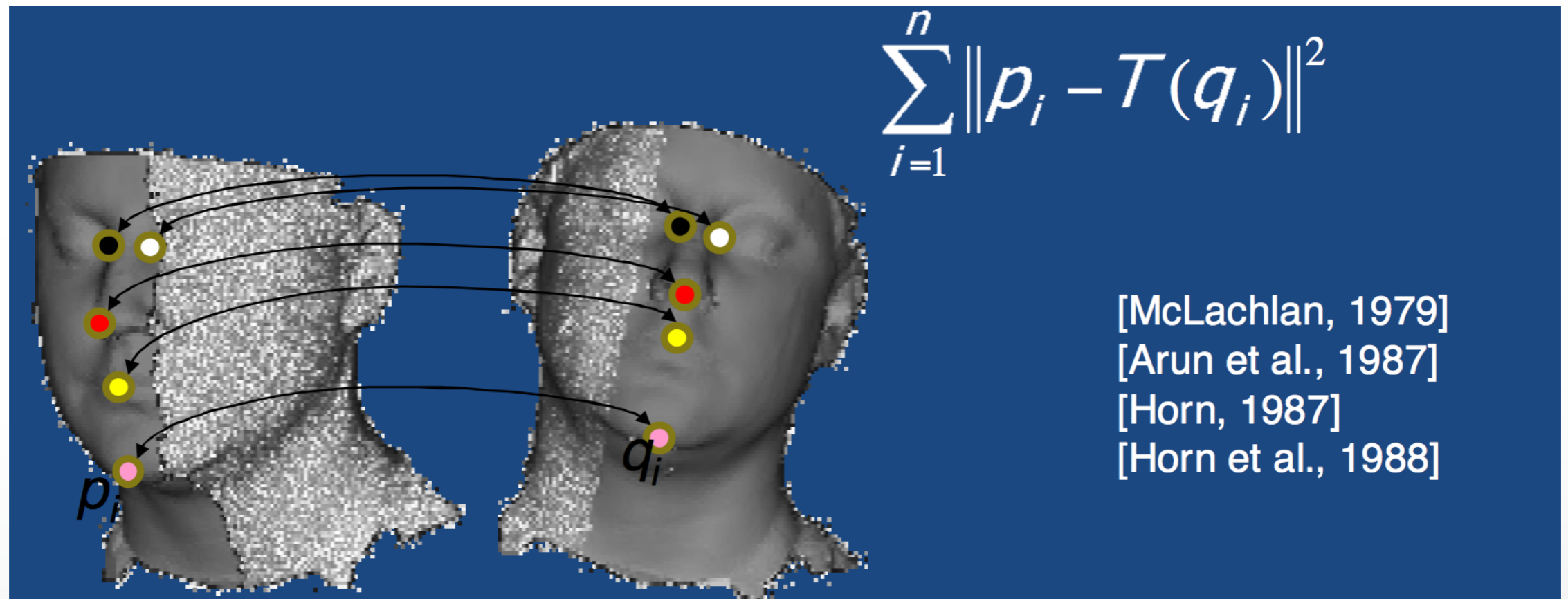
# Outline

- **Global Shape Correspondence**
  - Shape Descriptors
  - Alignment
- **Partial Shape Correspondence**
  - From Global to Local
  - Pose Normalization
  - Partial Shape Descriptors
- **Registration**
  - Closed Form Solutions
  - Branch & Bound
  - Random Sample Consensus (RANSAC)

# Registration

## Ideal Case

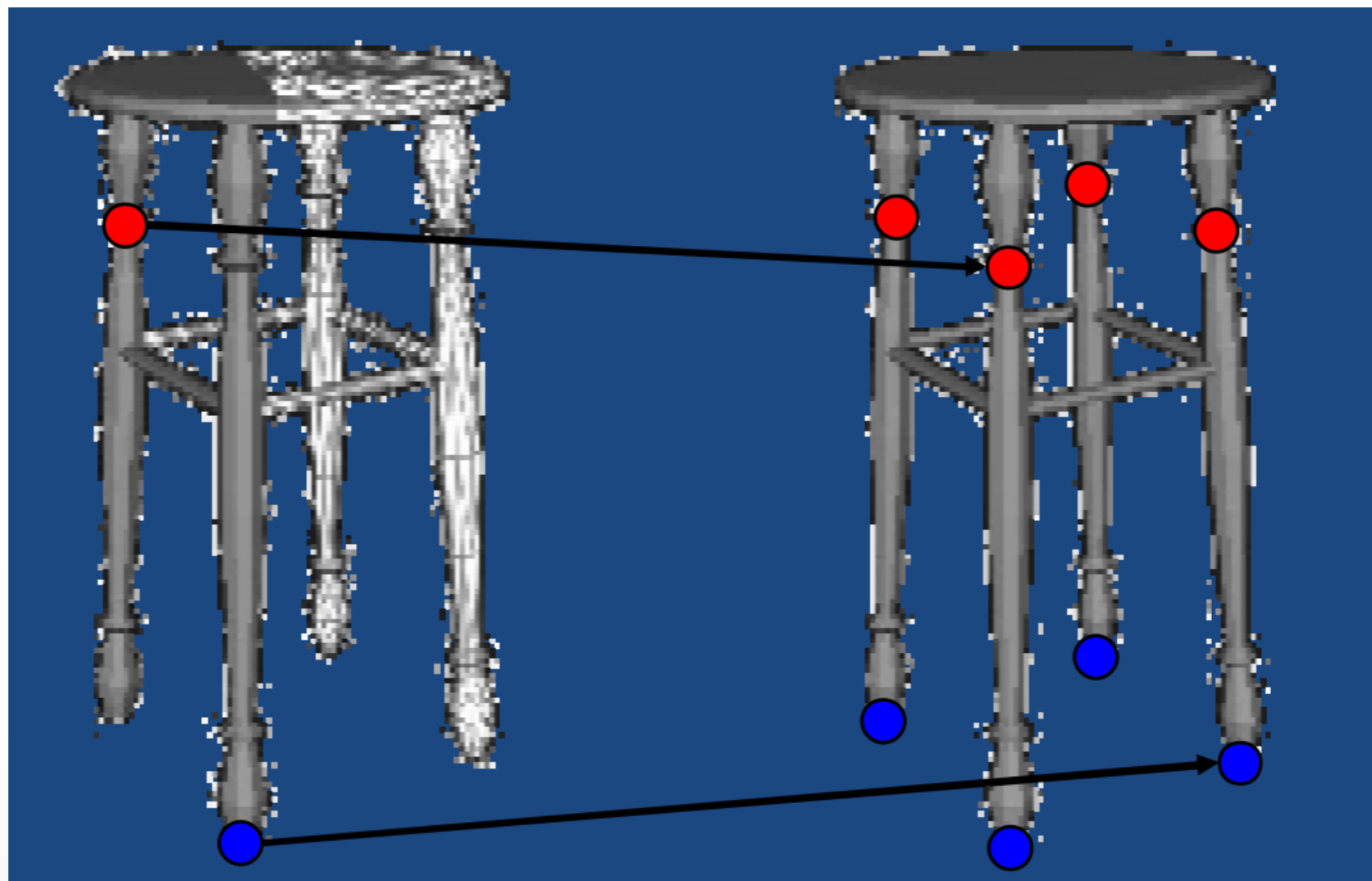
- Every feature point on one scan has a **single** corresponding feature on the other.
- Solve for optimal transformation  $T$



# Registration

## Challenge:

- Even with good descriptors, symmetries in the model and the locality of descriptors can result in multiple and incorrect correspondences



# Registration

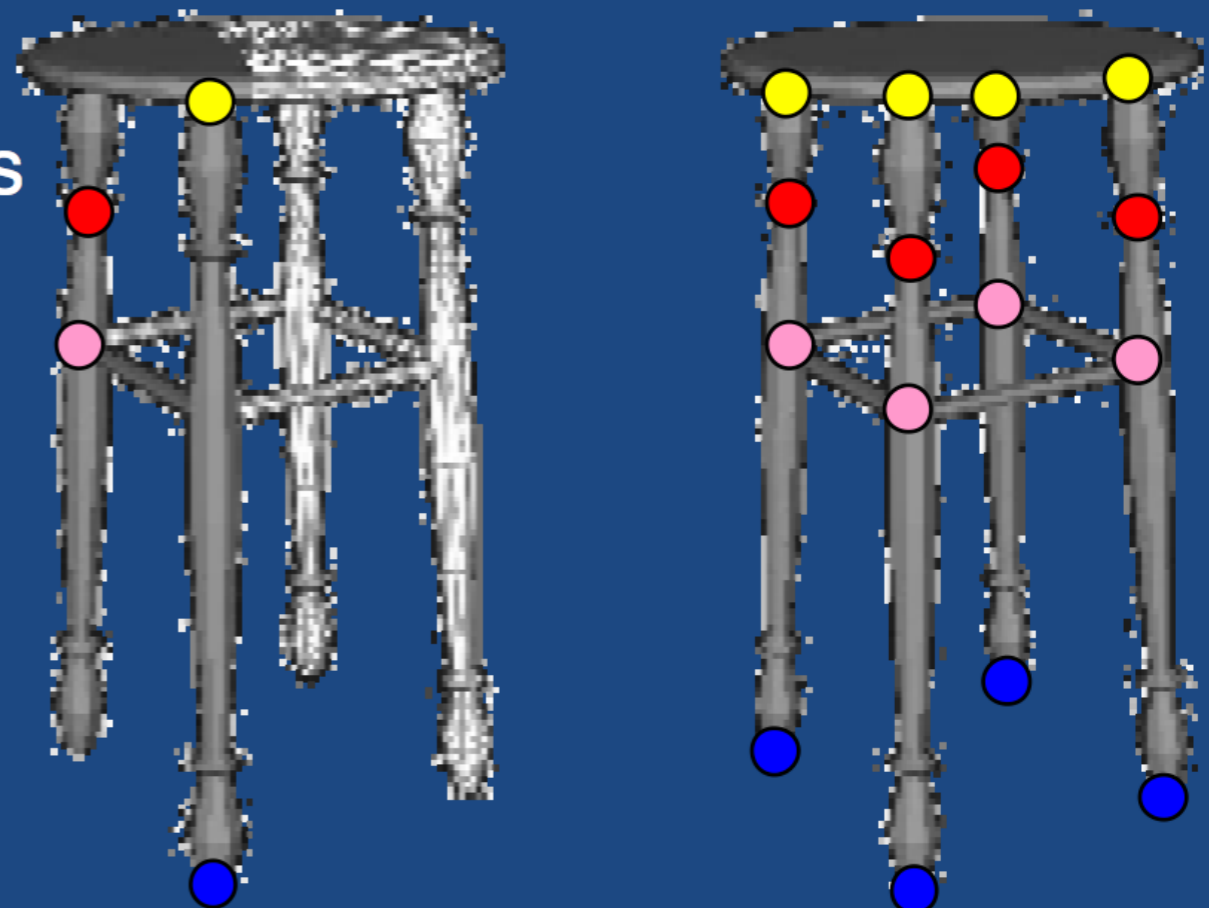
## Exhaustive Search

- Compute alignment error at each permutation of correspondences and use the optimal one

$$\text{Error} = \underset{\pi \in \Psi}{\text{argmin}} \left( \underset{T \in E^3}{\text{argmin}} \sum_{i=1}^n \|p_i - T(\pi(p_i))\|^2 \right)$$

$\Psi$  = Set of possible correspondence

$E^3$  = Group of rigid body transformations





# Registration

## Exhaustive Search

- Compute alignment error at each permutation of correspondences and use the optimal one

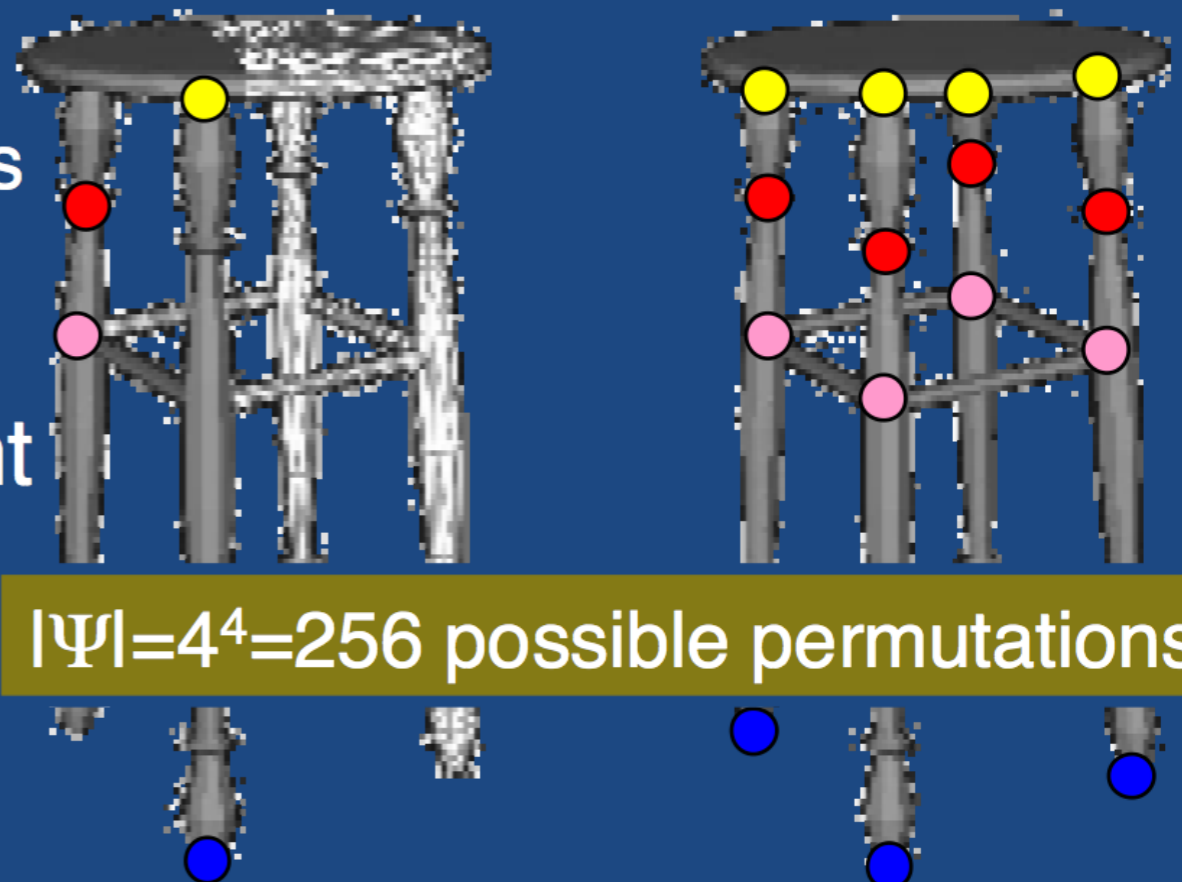
$$\text{Error} = \operatorname{argmin}_{\pi \in \Psi} \left( \operatorname{argmin}_{T \in E^3} \sum_{i=1}^n \|p_i - T(\pi(p_i))\|^2 \right)$$

$\Psi$  = Set of possible correspondence

$E^3$  = Group of rigid body transformations

Given points  $\{p_1, \dots, p_n\}$  on the query, if  $p_i$  matches  $m_i$  different target points:

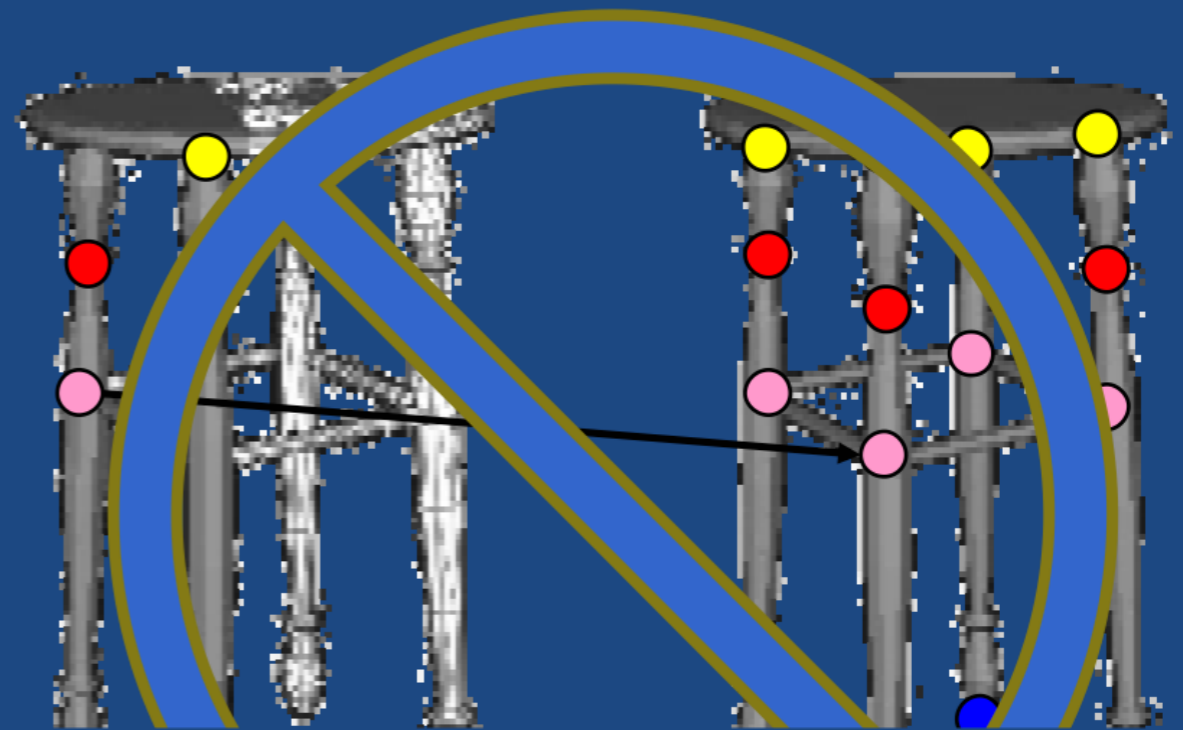
$$|\Psi| = \prod_{i=1}^n m_i$$



# Registration

## Branch & Bound (Decision tree)

- Try all permutations but terminate early if the alignment can be predicted to be bad



By performing two comparisons,  
it was possible to eliminate 16  
different possibilities

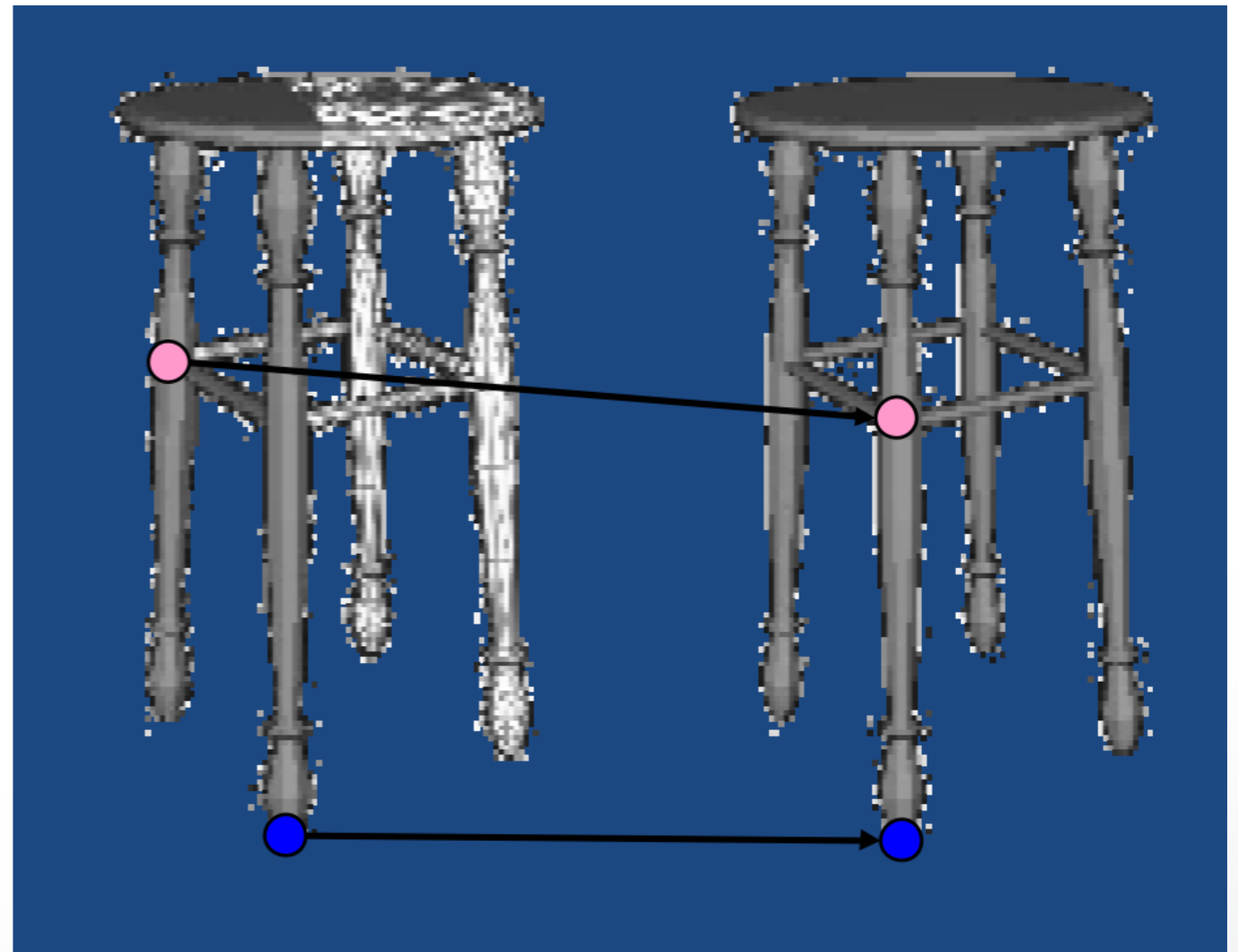
# Registration

## Goal

- Need to be able to determine if the alignment will be good without knowing all of the correspondences

## Observation

- Alignment needs to preserve the lengths between points in a single scan



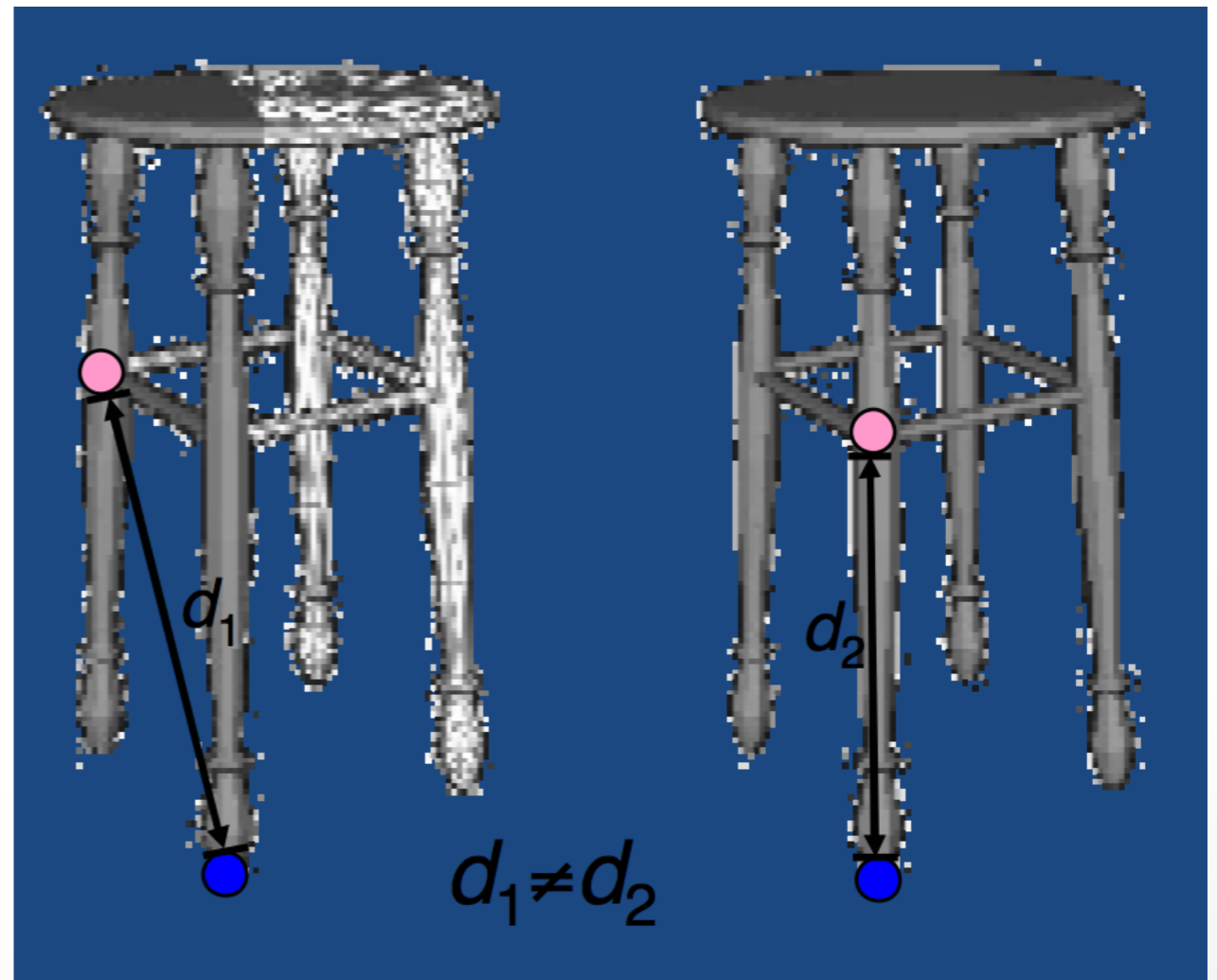
# Registration

## Goal

- Need to be able to determine if the alignment will be good without knowing all of the correspondences

## Observation

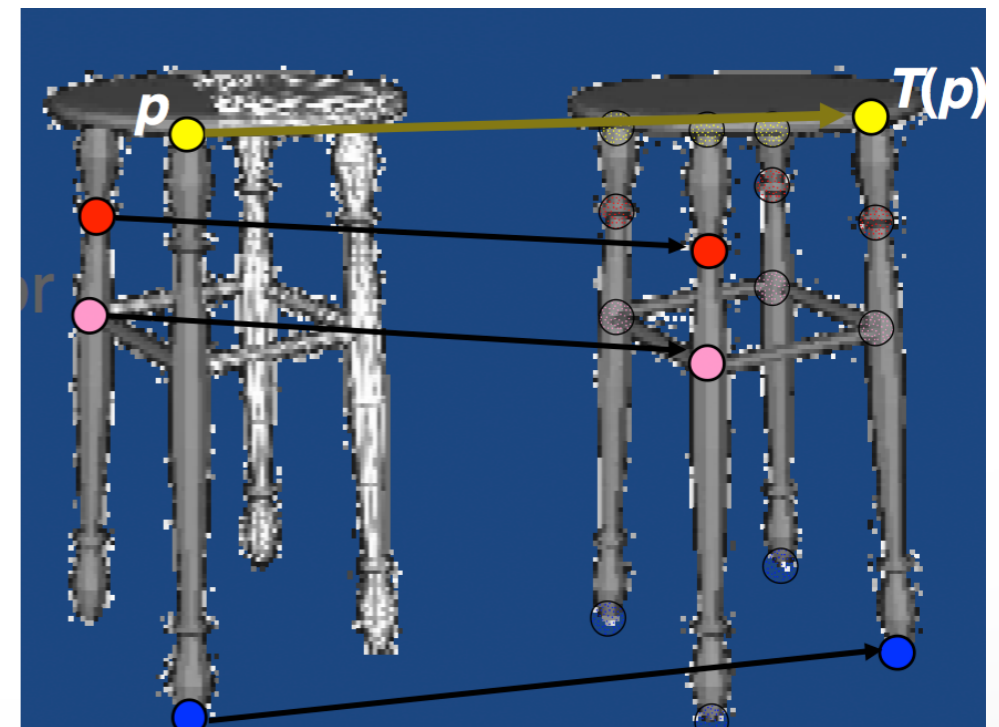
- Alignment needs to preserve the lengths between points in a single scan



# RANdom SAmple Consensus

## Algorithm (iterate 100 times)

- Randomly choose 3 points on source
- For all possible correspondences on target:
  - Compute  $T$
  - For every other source  $p$ :
    - find closest correspondence  $T(p)$
  - Compute alignment error



# Summary

## Global Shape Correspondences

- Shape Descriptors
  - Shells (1D)
  - Sectors (2D)
  - Sectors & Shells (3D)
- Alignment
  - Exhaustive Search
  - Normalization
  - Invariance

# Summary

## Partial-Shape/Point Correspondences

- From Global to Local
  - Center at feature
  - Restrict extent
- Pose Normalization
  - Normal-based alignment
- Partial Shape Descriptors
  - Normalization/invariance
  - Normalization/exhaustive-search

# Summary

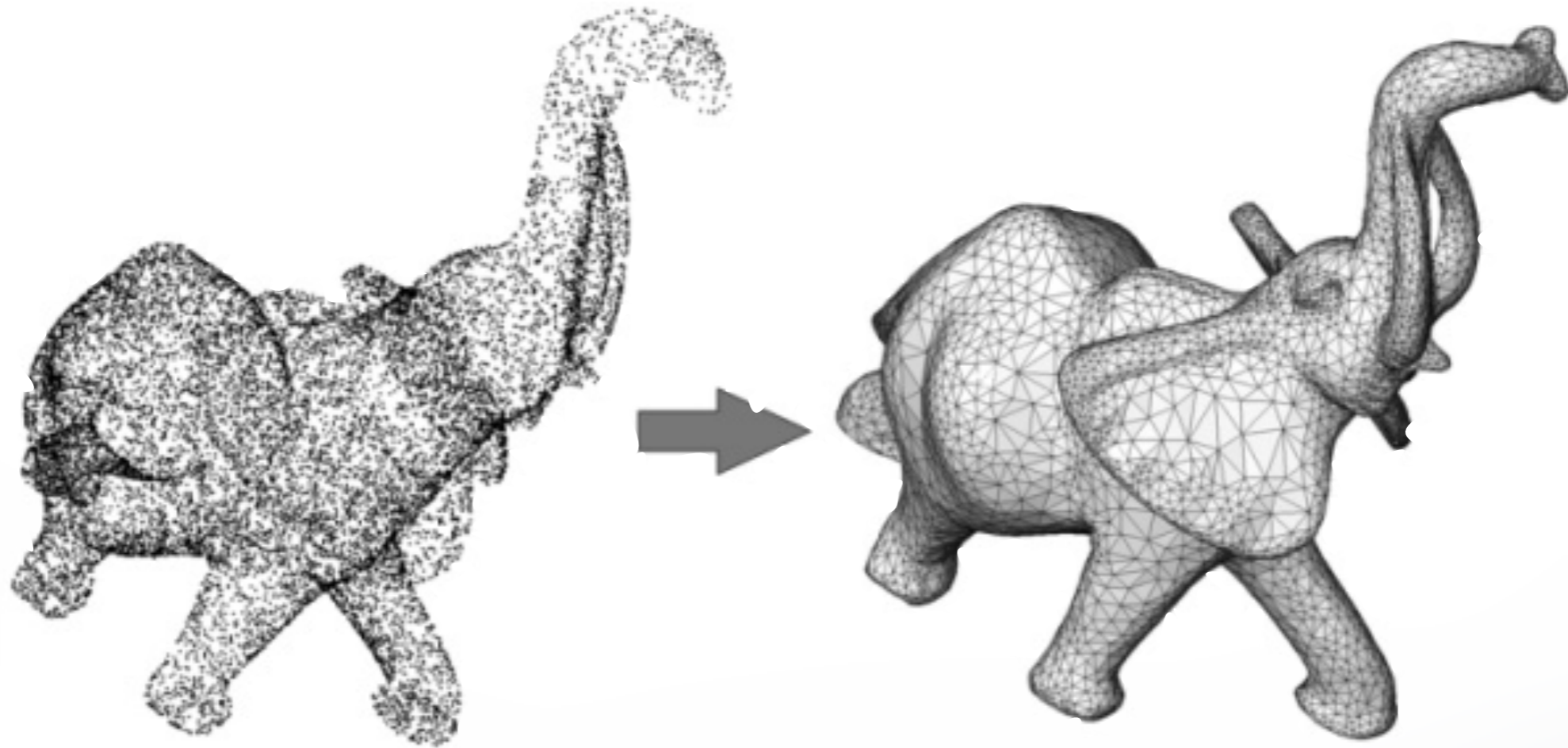
## Registration

- Closed Form Solutions
  - Global symmetry
  - Local self similarity
- Branch & Bound
  - Inter-feature distances for early termination
- RANdom SAmple Consensus
  - Efficient transformation computation



# Next Time

## Surface Reconstruction



<http://cs621.hao-li.com>

# Thanks!

