

# Extracting Structures

---

## Analysis of Individual Models

# Symmetry in Shapes





# Classification

- Global vs. Partial



*(a) complete symmetry group on parts of a shape*



*(b) partial translational symmetry*



*(c) partial rotational symmetry*

# Classification

- Global vs. Partial
- Exact vs. Approximate



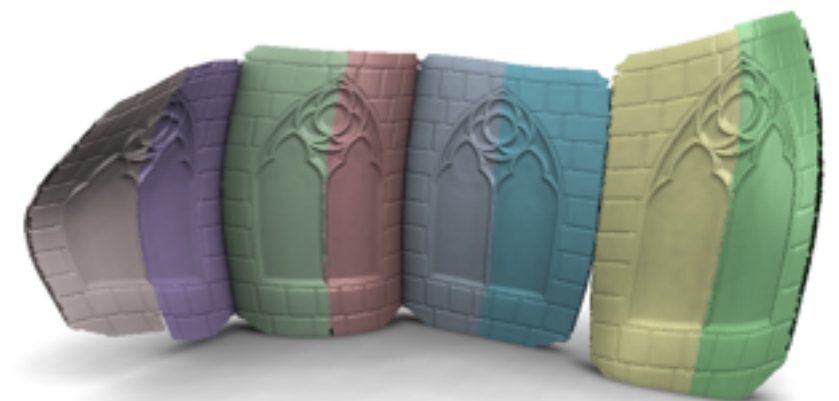
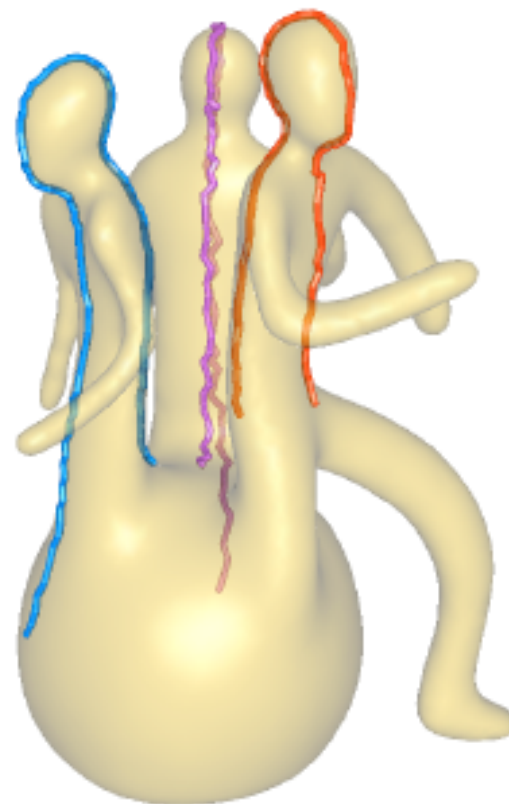
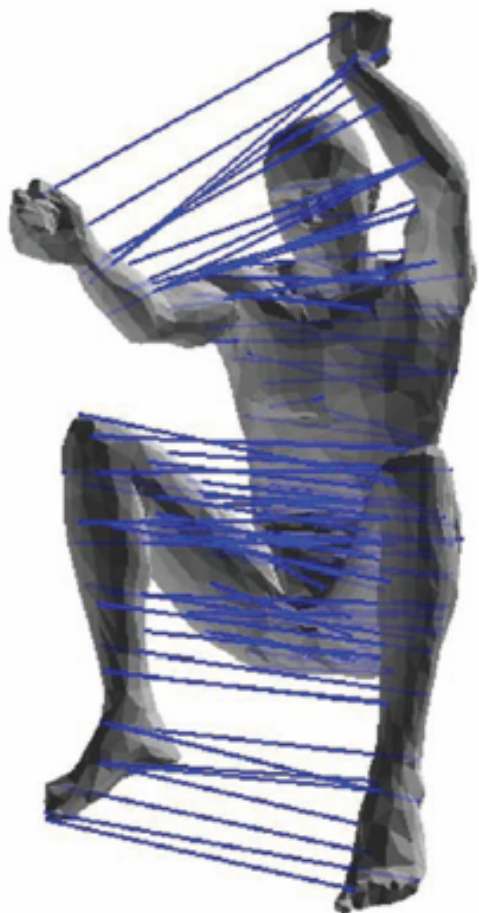
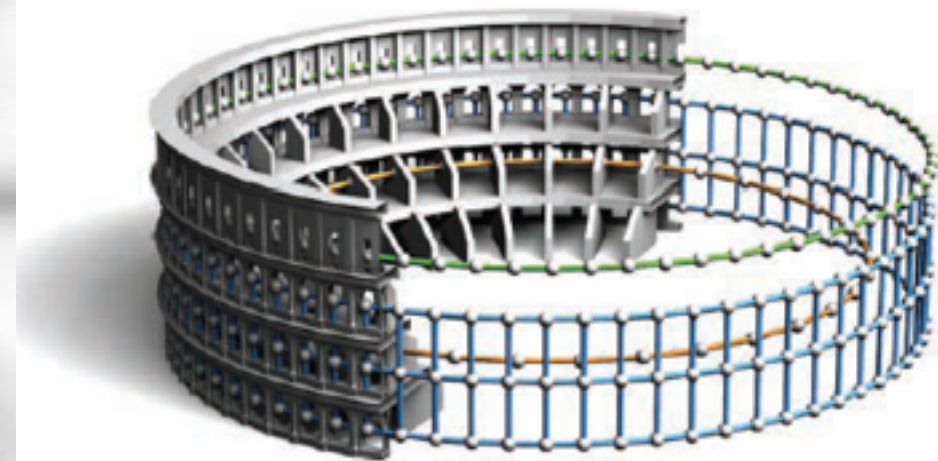
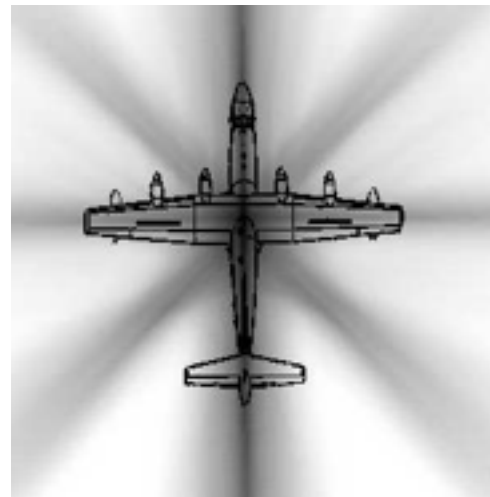
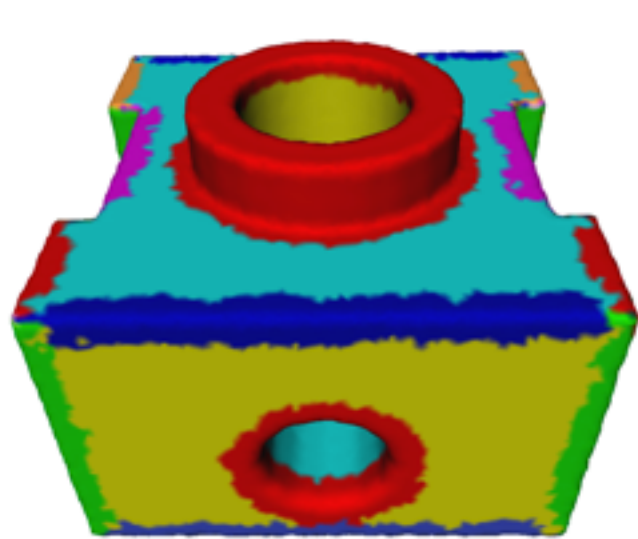
# Classification

- Global vs. Partial
- Exact vs. Approximate
- Intrinsic vs. Extrinsic





# Methods Gallery

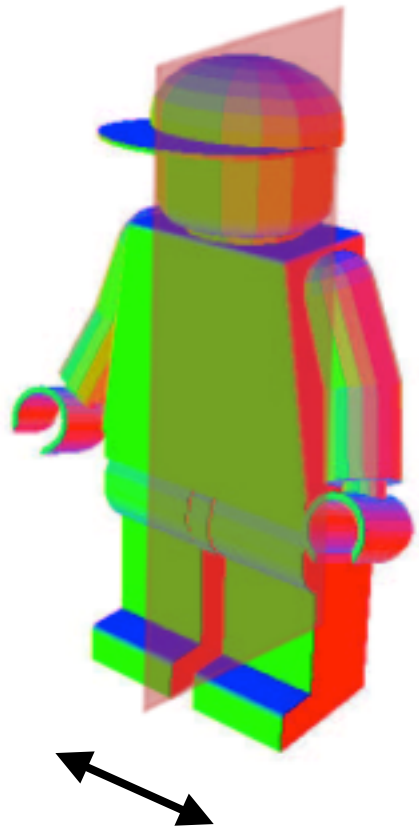




# Global vs. Partial

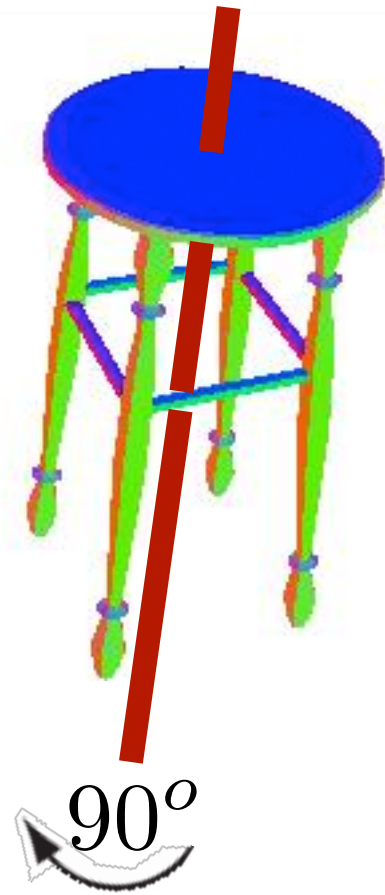
---

# Global vs. Partial



reflections

[Kazhdan et al. 2003] [Kazhdan et al. 2004]



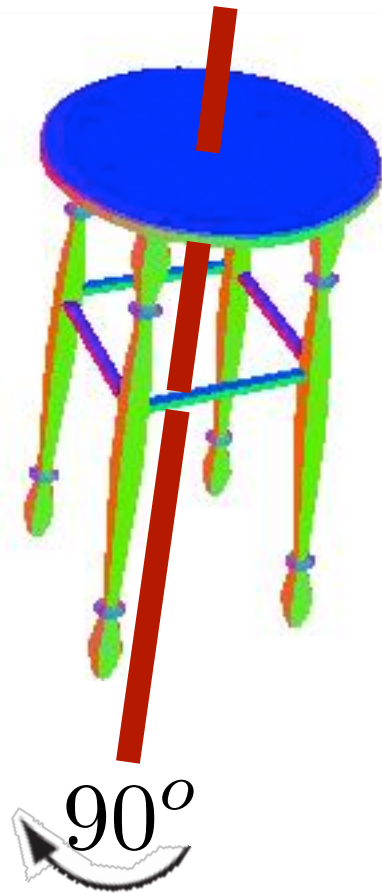
rotations

# Global vs. Partial



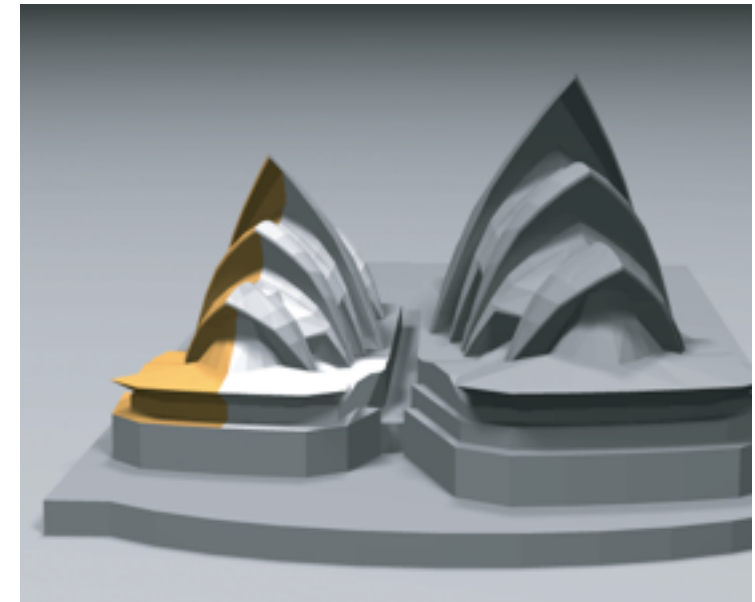
reflections

[Kazhdan et al. 2003]

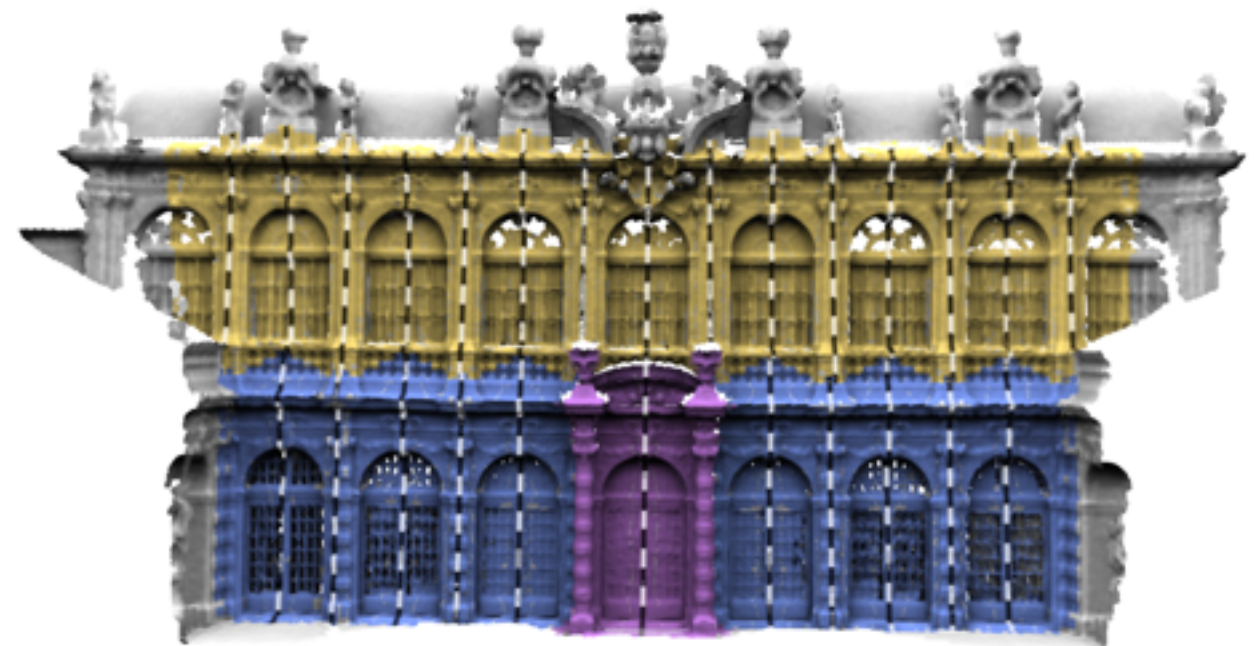


rotations

[Kazhdan et al. 2004]



[Mitra et al. 2006]



[Bokeloh et al. 2009]

# Extrinsic vs. Intrinsic



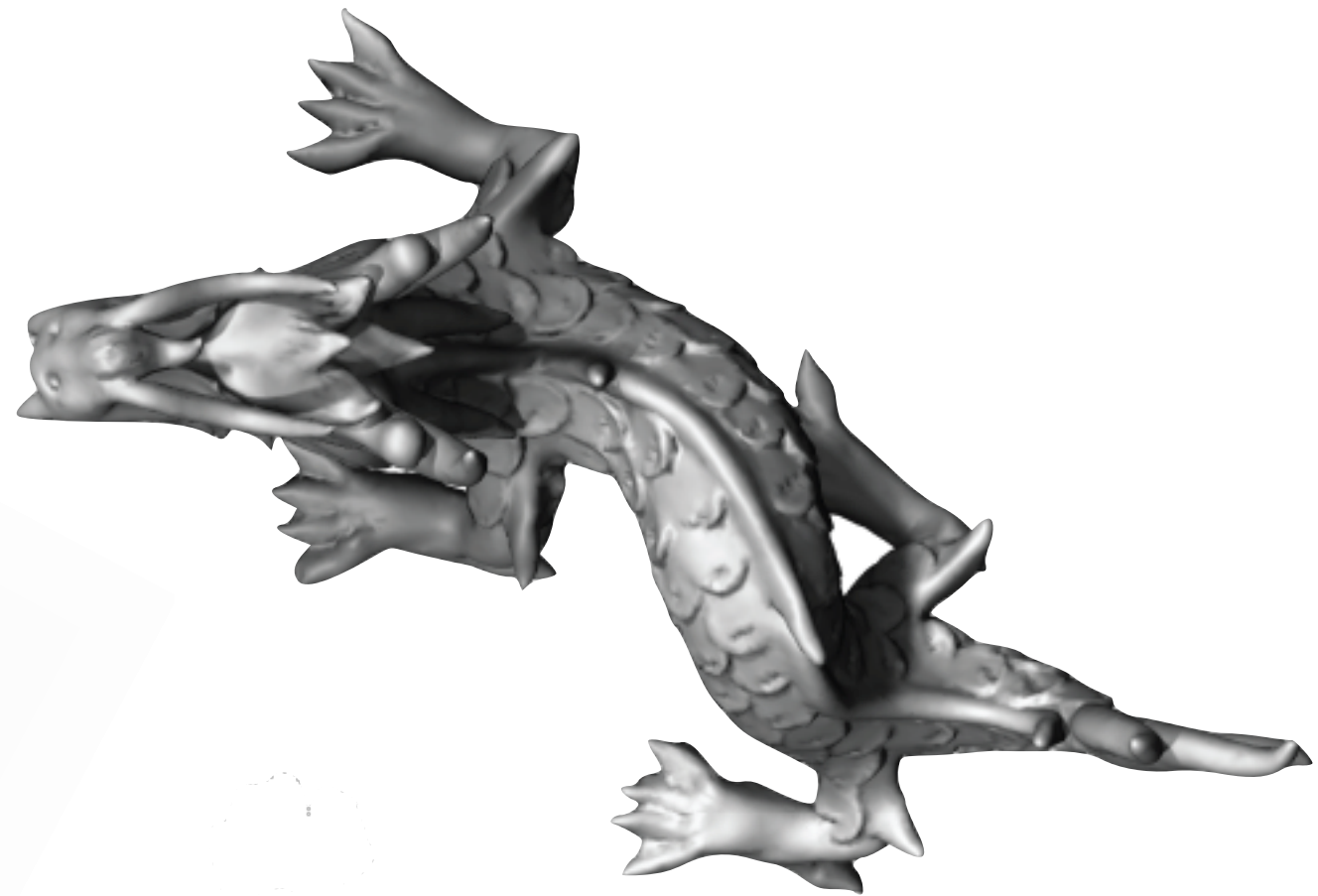
Extrinsic symmetries depend on the **embedding of the object** in space.



# Extrinsic vs. Intrinsic



Extrinsic symmetries depend on the **embedding of the object** in space.

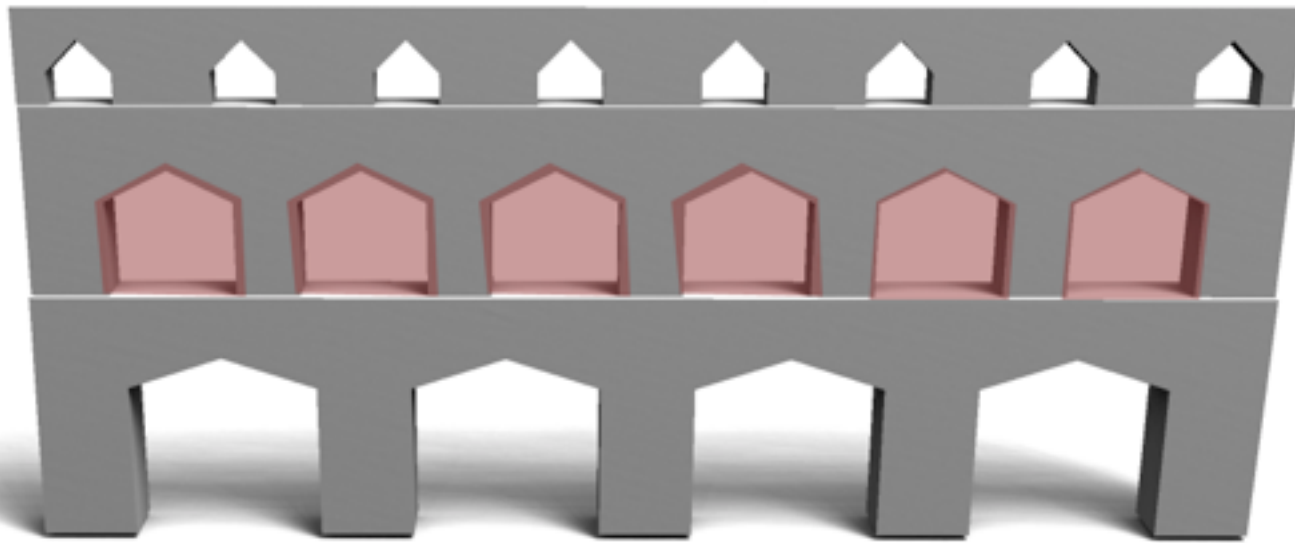


Intrinsic symmetries are defined with respect to an **intrinsic metric** of the surface.

# Extrinsic Symmetries

---

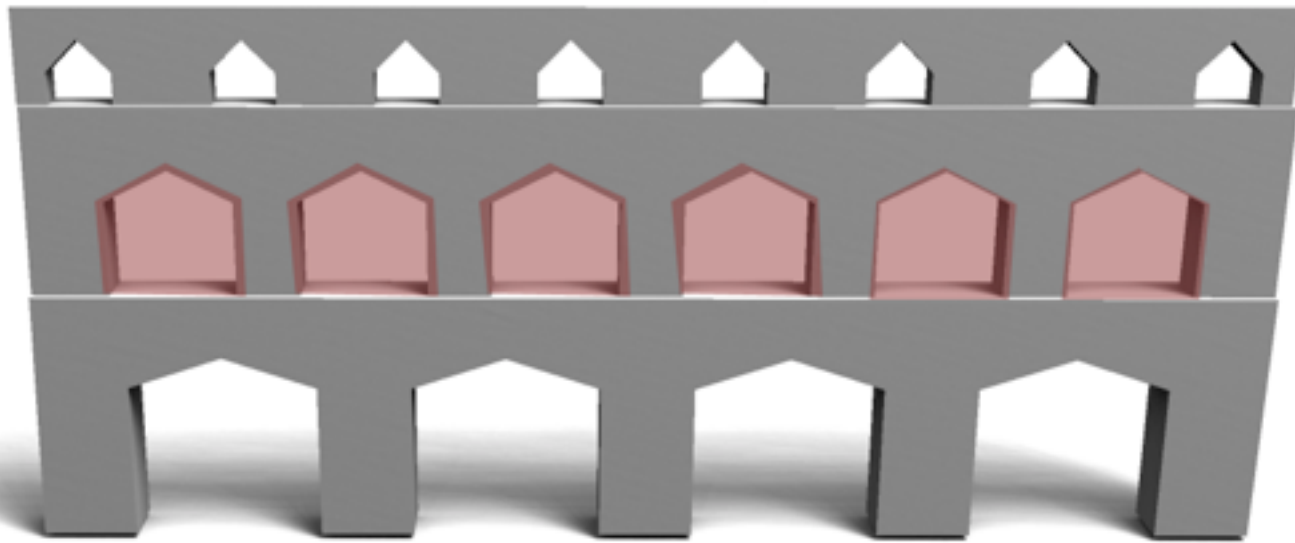
# Extrinsic Symmetries



translations

[Bokeloh et al. 2011]

# Extrinsic Symmetries



translations

[Bokeloh et al. 2011]



rotations

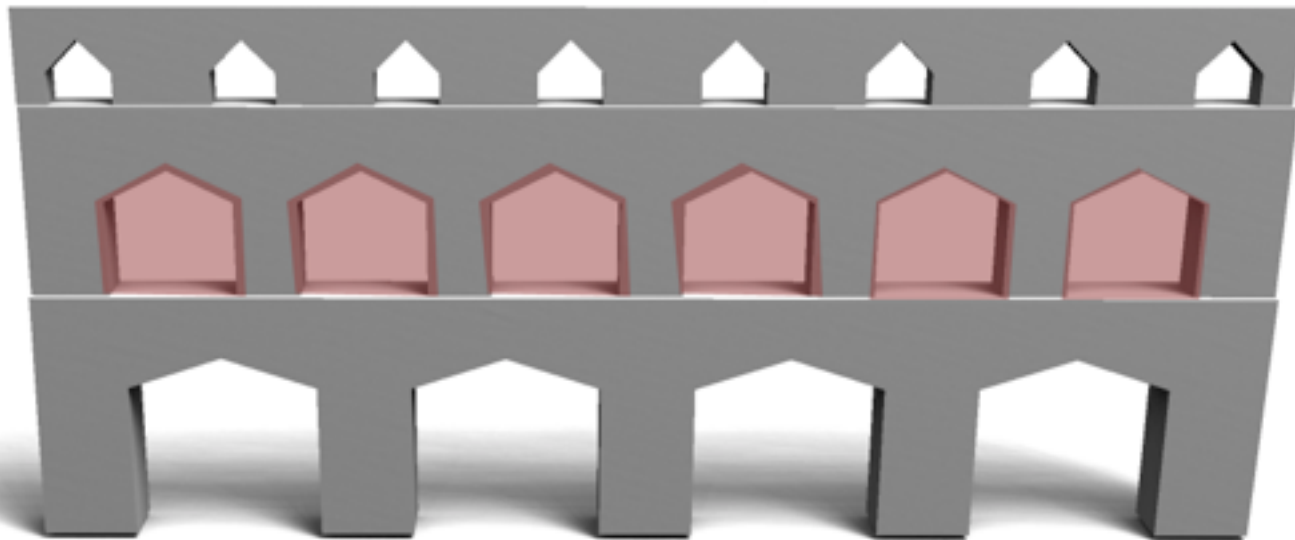


reflections

[Martinet et al. 2006]



# Extrinsic Symmetries



translations

[Bokeloh et al. 2011]



rotations



reflections

[Martinet et al. 2006]



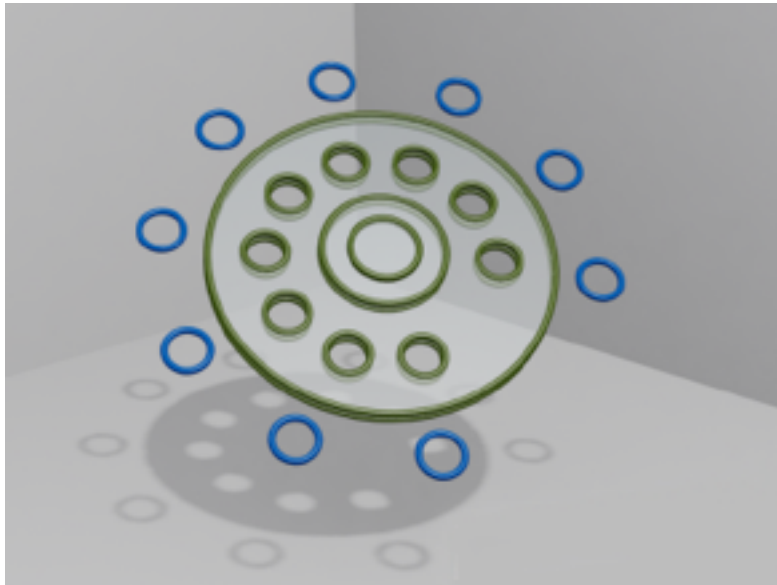
similarity transforms

[Gal et al. 2006]

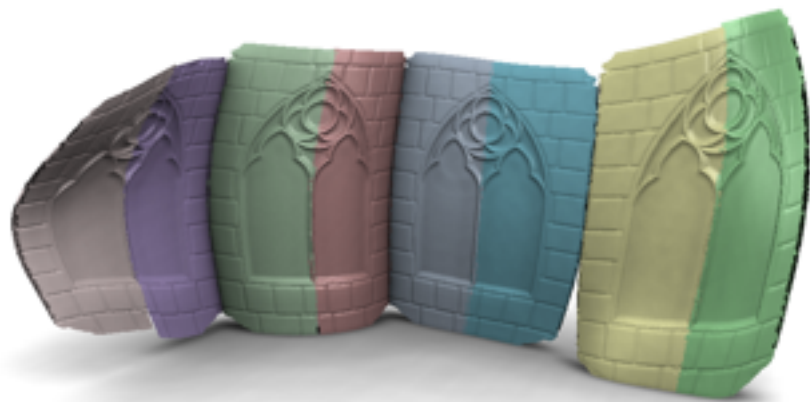
# Discrete vs. Continuous

---

# Discrete vs. Continuous

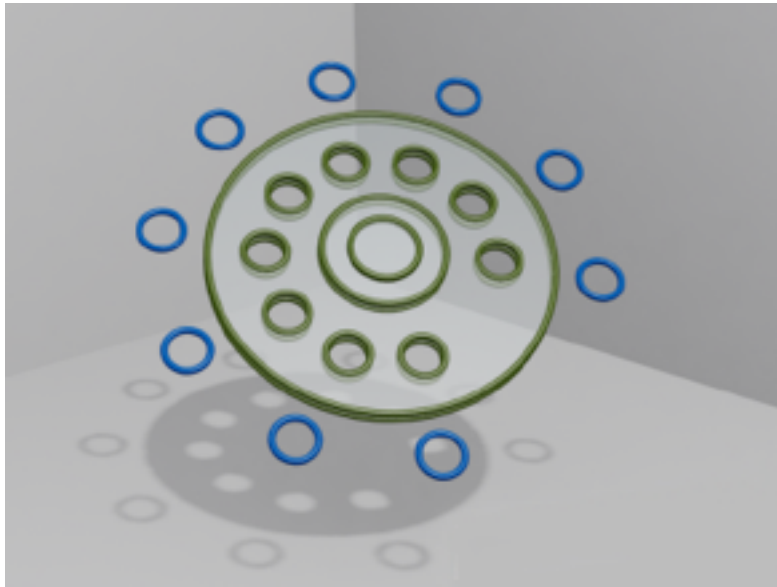


[Gal et al. 2009]

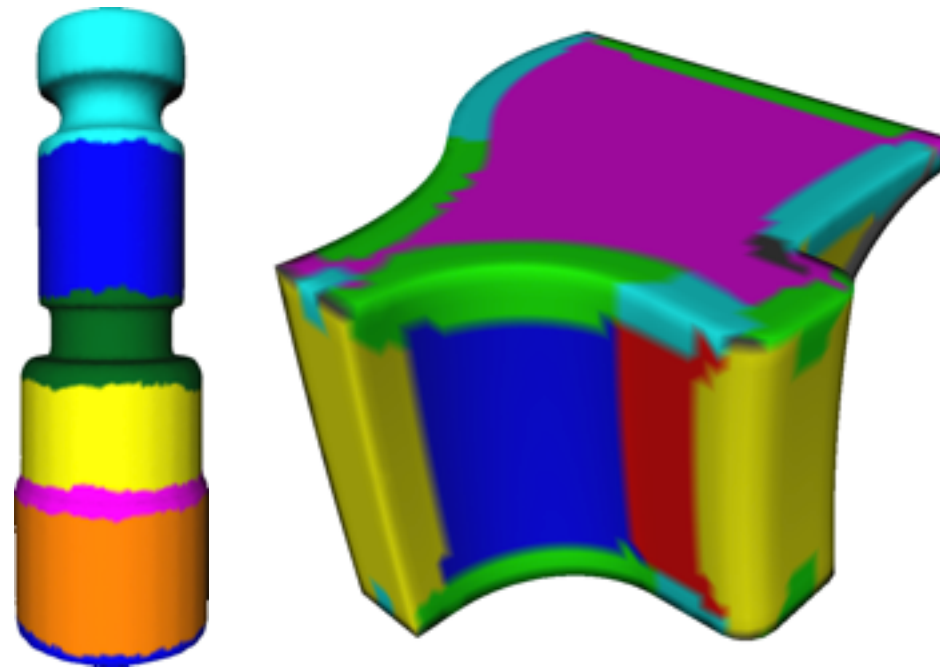


[Berner et al. 2009]

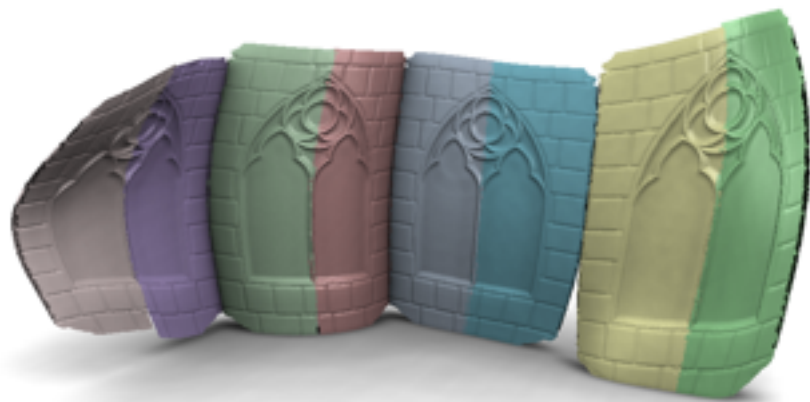
# Discrete vs. Continuous



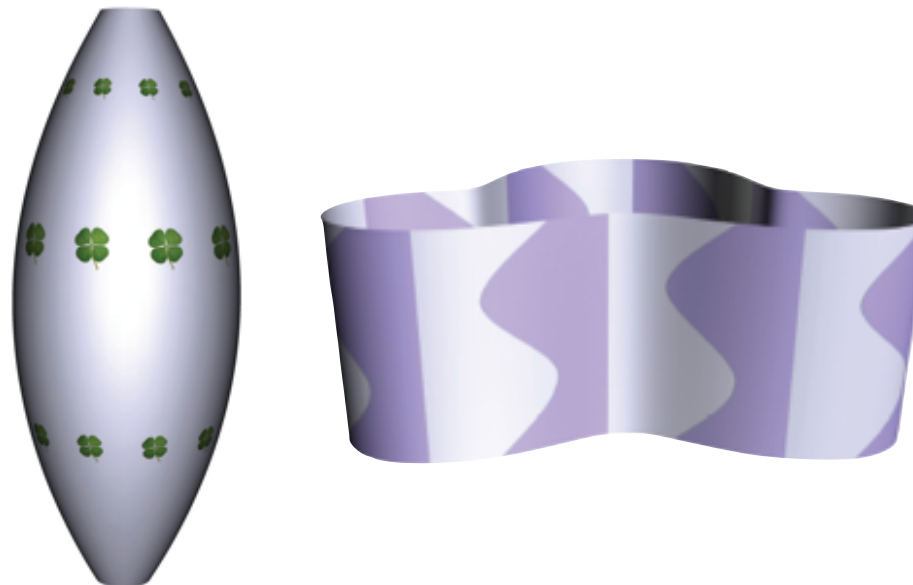
[Gal et al. 2009]



[Gelfand et al. 2004]



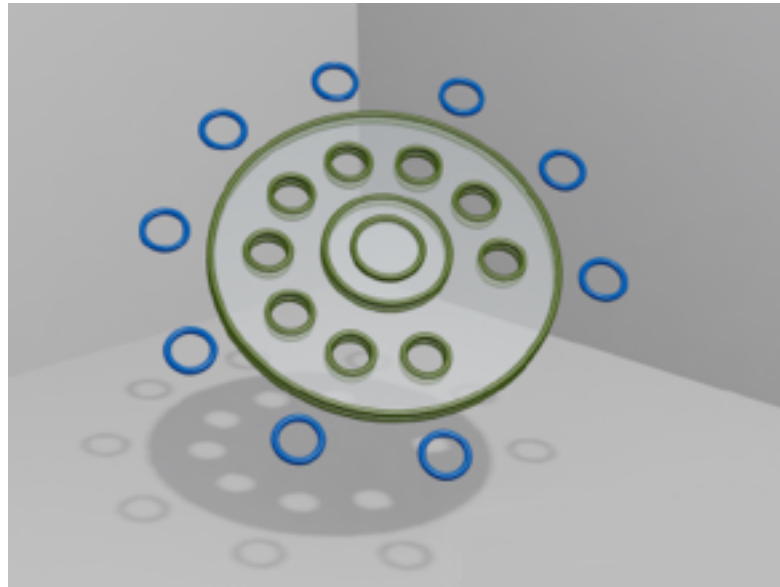
[Berner et al. 2009]



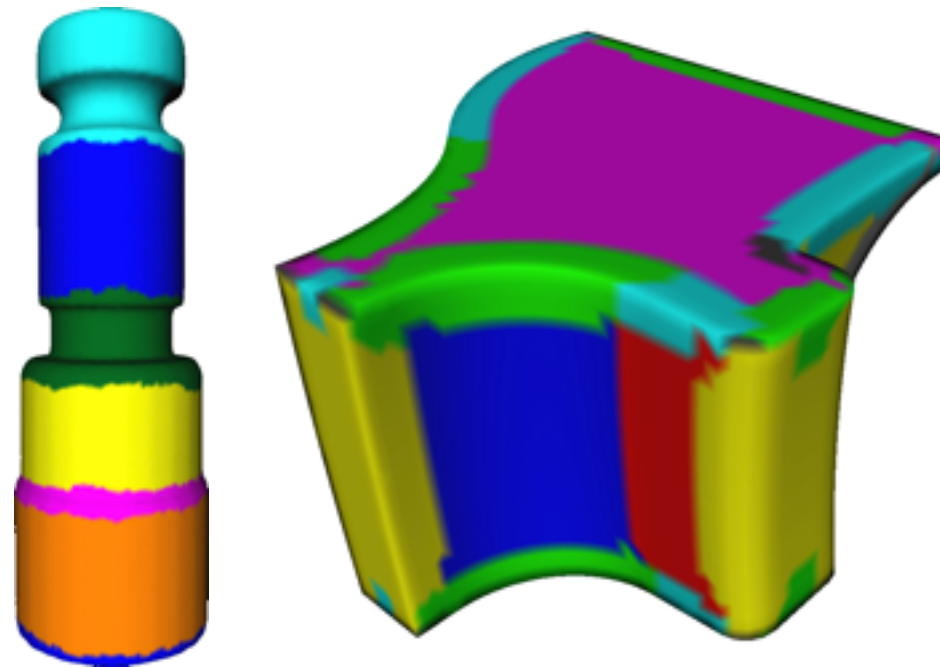
[Ben-Chen et al. 2010]



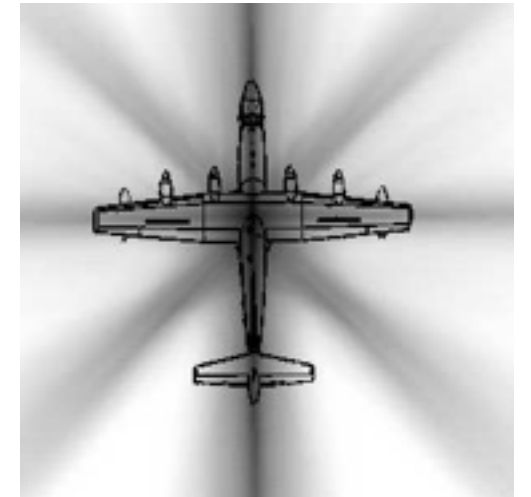
# Discrete vs. Continuous



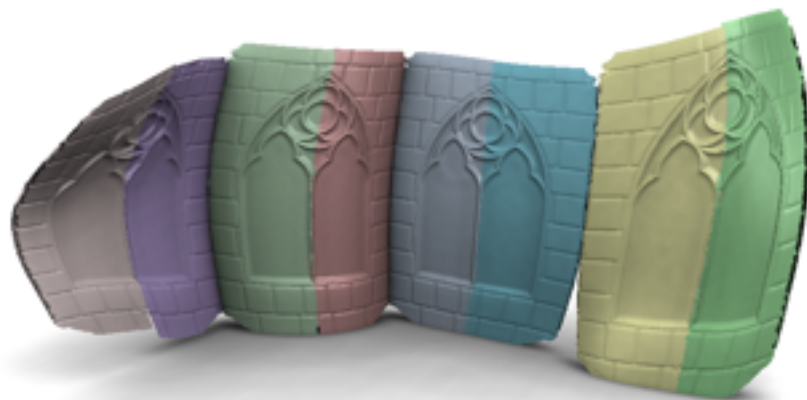
[Gal et al. 2009]



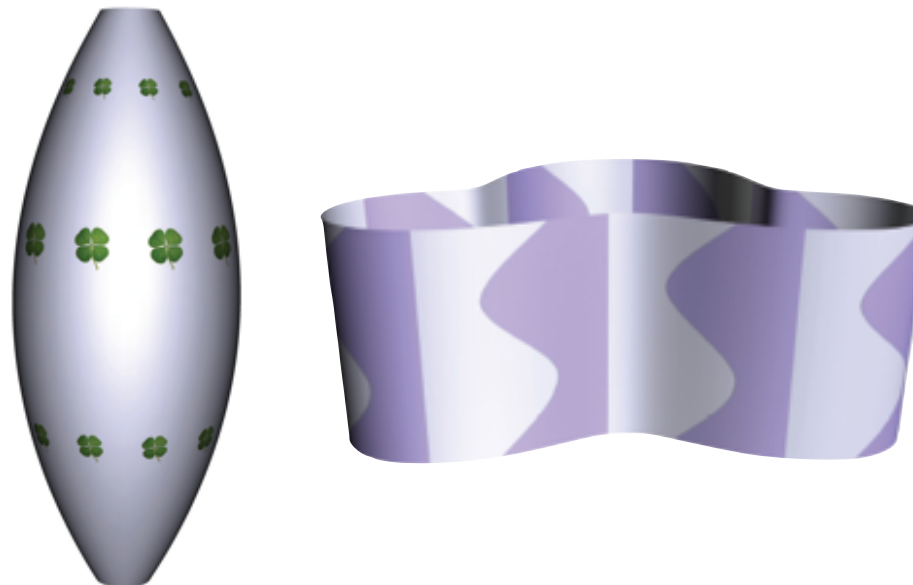
[Gelfand et al. 2004]



[Podalak et al. 2006]



[Berner et al. 2009]



[Ben-Chen et al. 2010]

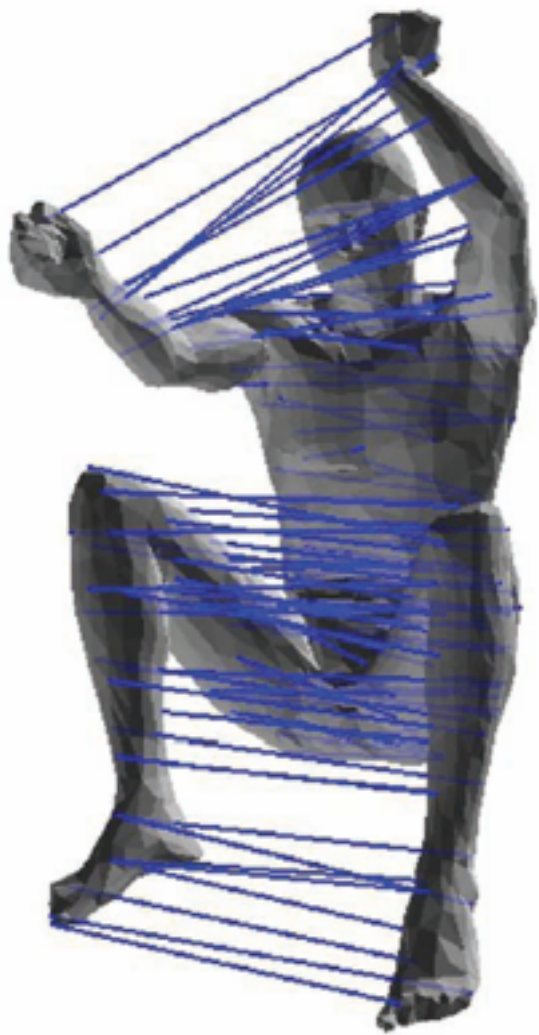


[Lipman et al. 2010]

# Output Structure

---

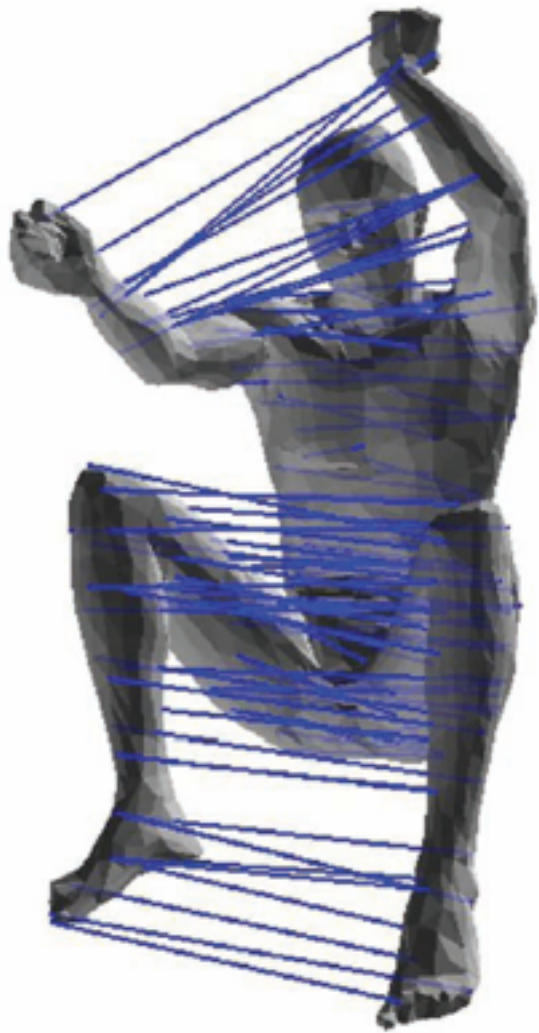
# Output Structure



pairwise

[Kim et al. 2010]

# Output Structure



pairwise

[Kim et al. 2010]

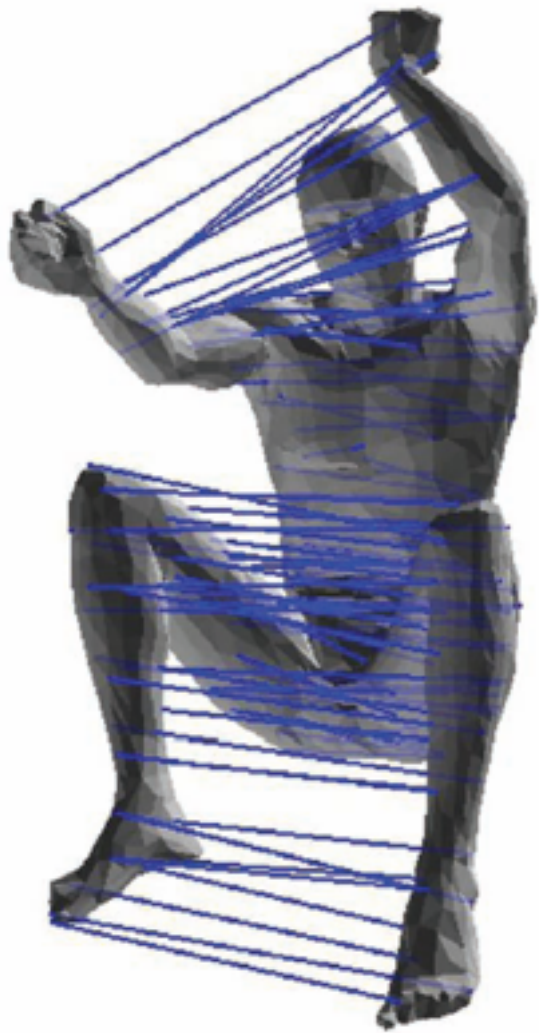


segmentation

[Mitra et al. 2006]



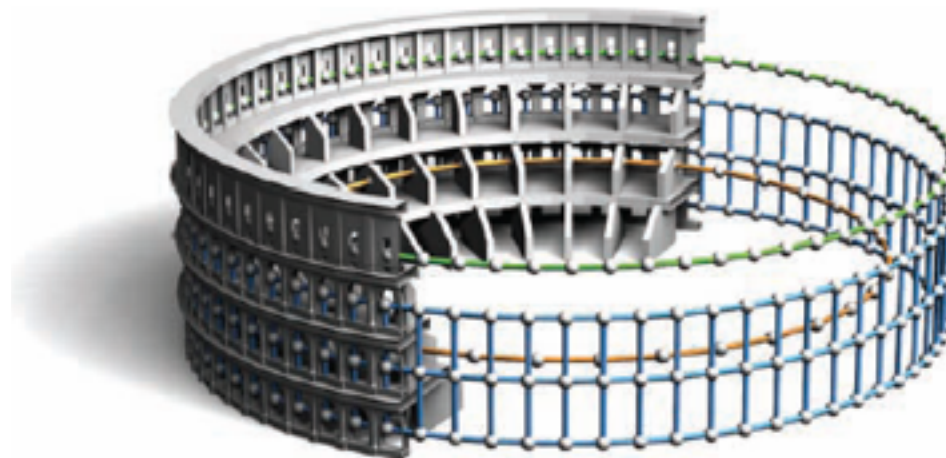
# Output Structure



pairwise  
[Kim et al. 2010]



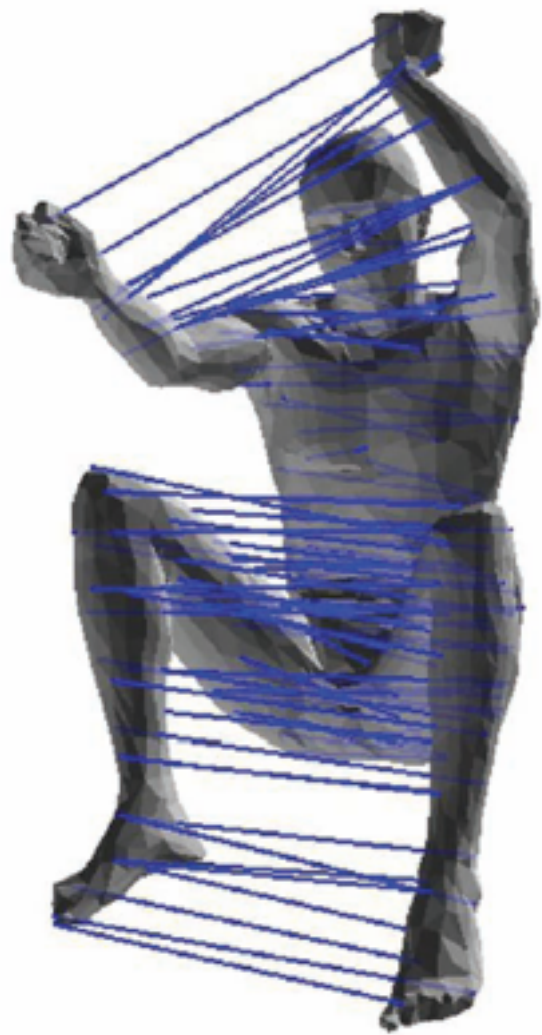
segmentation  
[Mitra et al. 2006]



symmetry groups  
[Pauly et al. 2008]



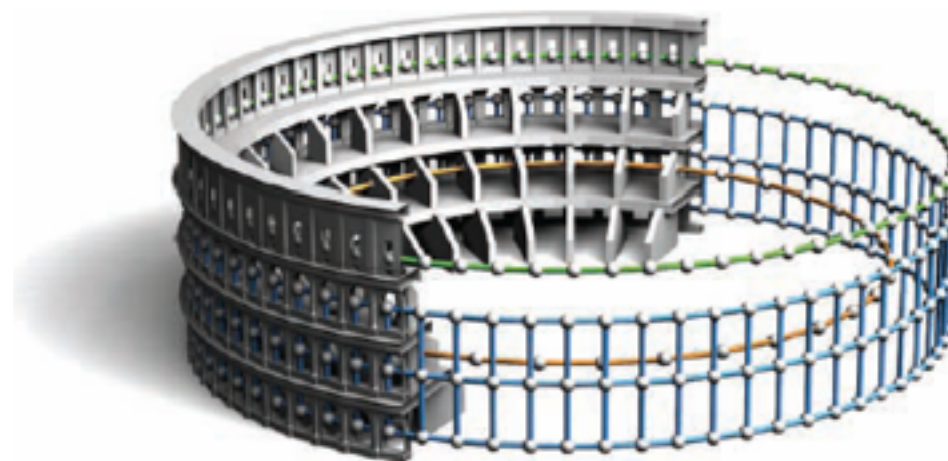
# Output Structure



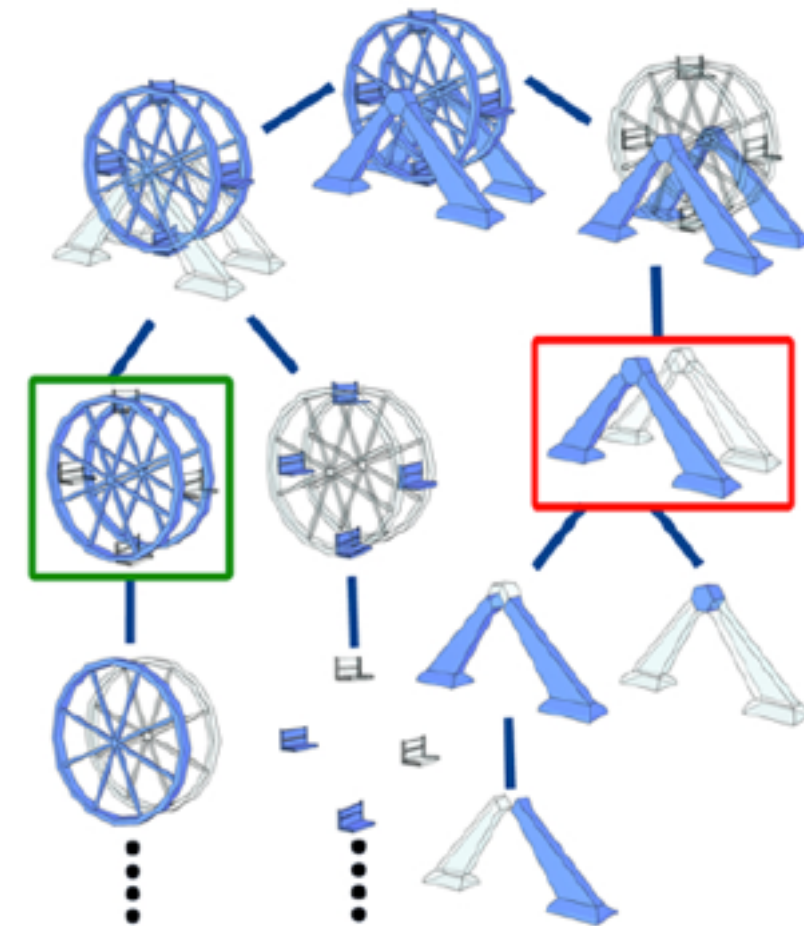
pairwise  
[Kim et al. 2010]



segmentation  
[Mitra et al. 2006]



symmetry groups  
[Pauly et al. 2008]

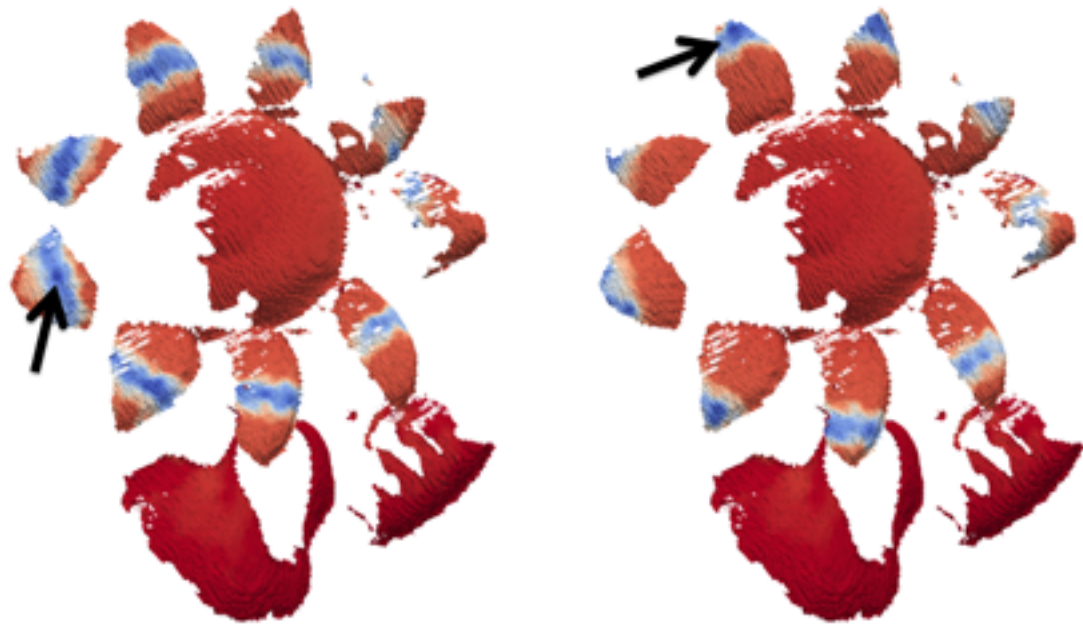


hierarchy  
[Wang et al. 2011]

# Output Structure

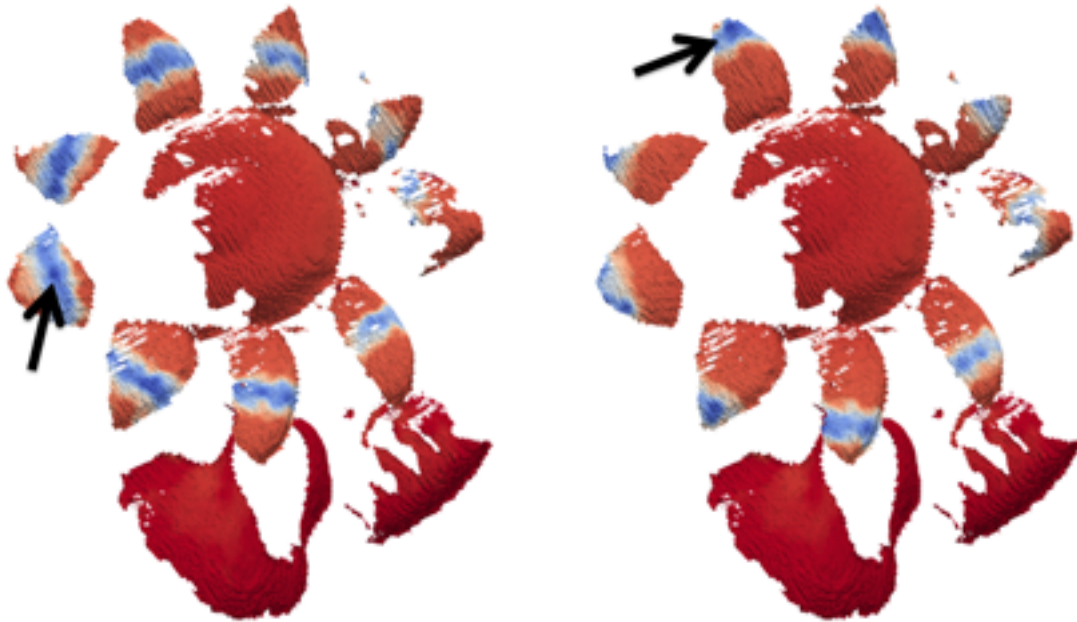
---

# Output Structure



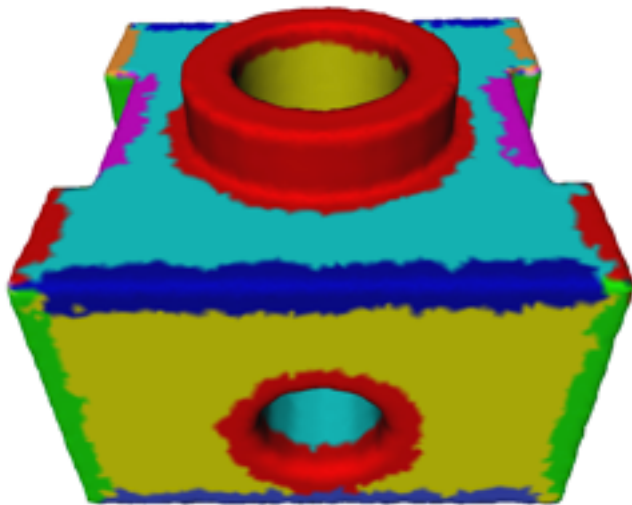
symmetry orbits  
[Lipman et al. 2010]

# Output Structure



symmetry orbits

[Lipman et al. 2010]

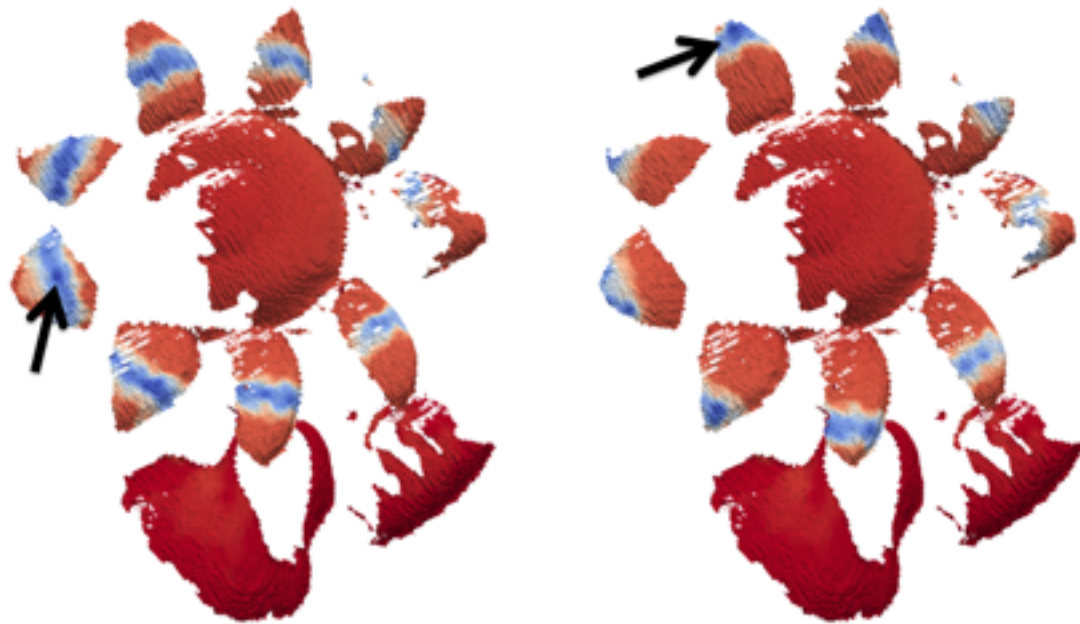


slippable regions

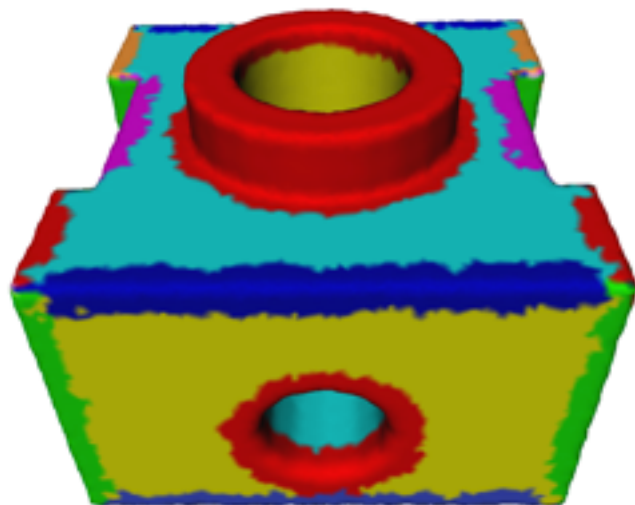
[Gelfand et al. 2004]



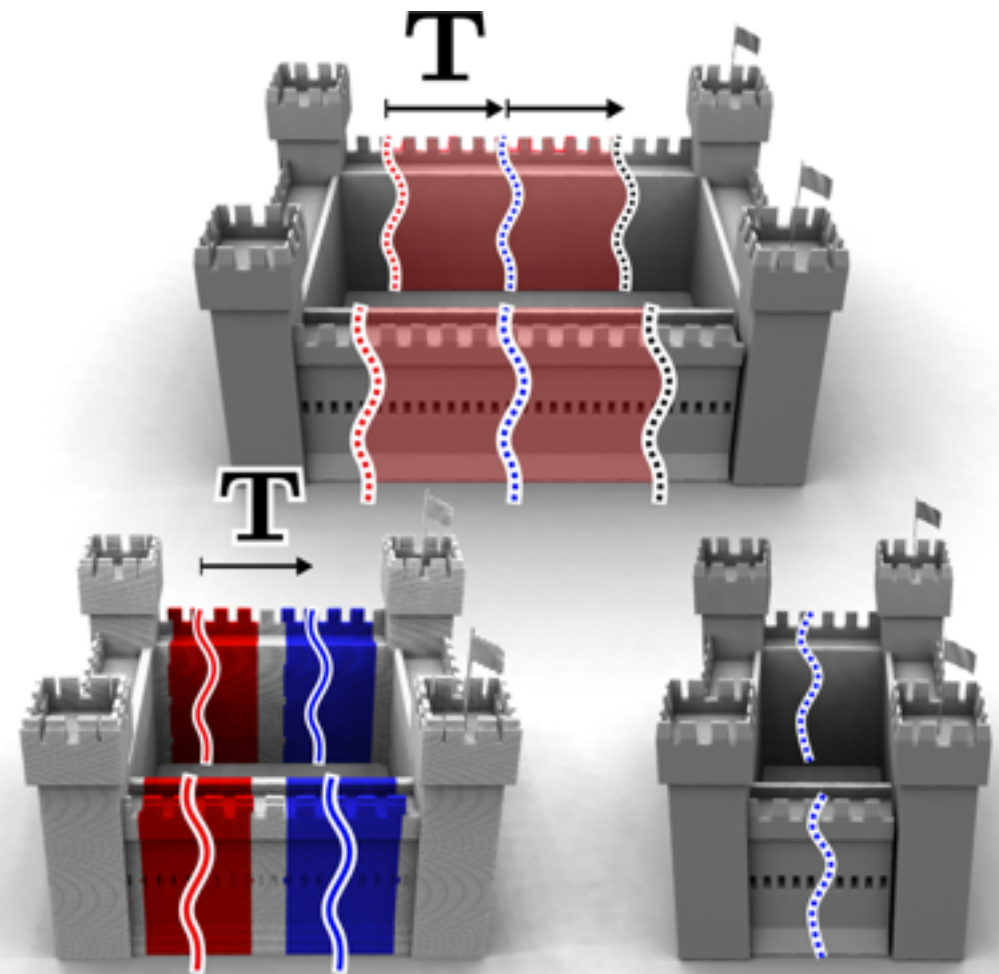
# Output Structure



symmetry orbits  
[Lipman et al. 2010]



slippable regions  
[Gelfand et al. 2004]



docking sites  
[Bokeloh et al. 2010]



# Problem Characteristics

---

# Problem Characteristics

---

**Difficult**

# Problem Characteristics

---

## Difficult

- Which parts are symmetric →  
objects are **not pre-segmented**

# Problem Characteristics

---

## Difficult

- Which parts are symmetric →  
objects are **not pre-segmented**
- **Transformation groups**: rotation, translation, scaling, etc.

## Difficult

- Which parts are symmetric → objects are **not pre-segmented**
- **Transformation groups**: rotation, translation, scaling, etc.
- Brute force search is **unrealistic**



# Problem Characteristics

---

## Difficult

- Which parts are symmetric → objects are **not pre-segmented**
- **Transformation groups**: rotation, translation, scaling, etc.
- Brute force search is **unrealistic**

## Easy

## Difficult

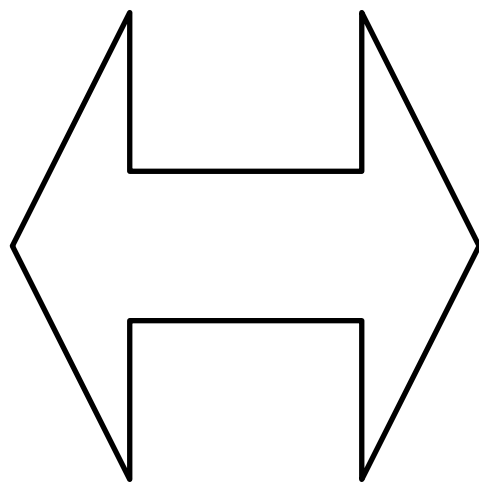
- Which parts are symmetric → objects are **not pre-segmented**
- **Transformation groups**: rotation, translation, scaling, etc.
- Brute force search is **unrealistic**

## Easy

- Proposed symmetries → **easy to validate**

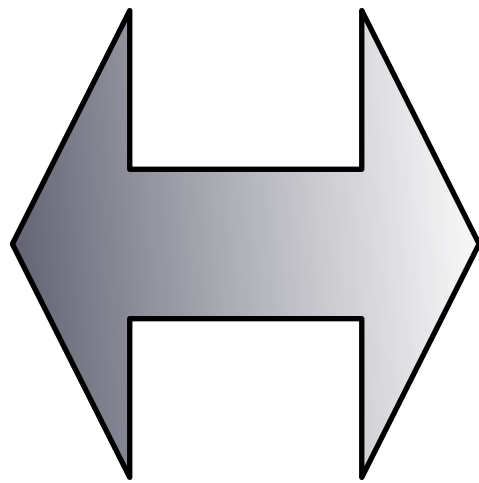
# Symmetry Detection

$M$



# Geometric Matching

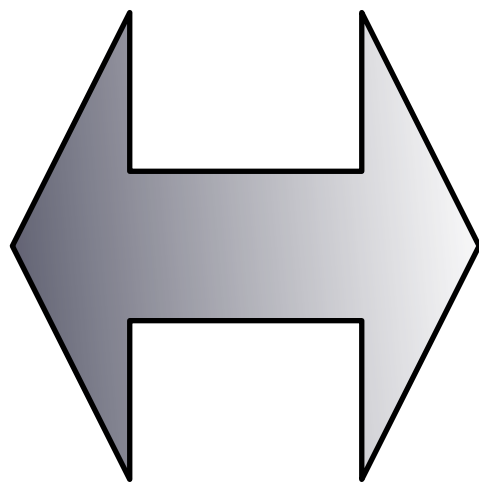
$M_1$



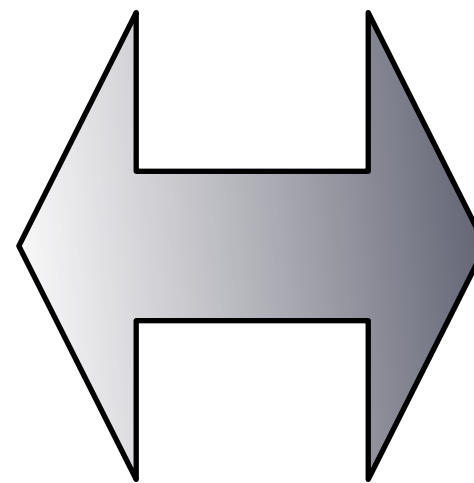


# Geometric Matching

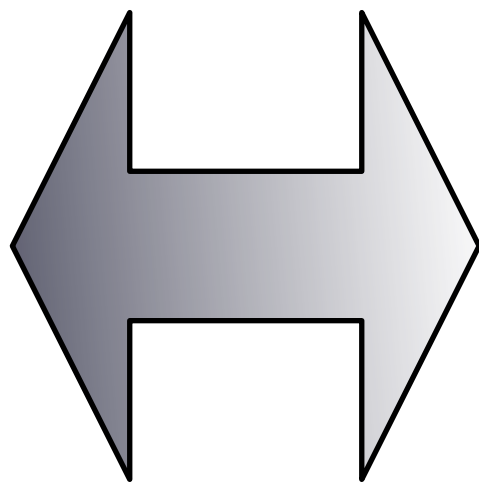
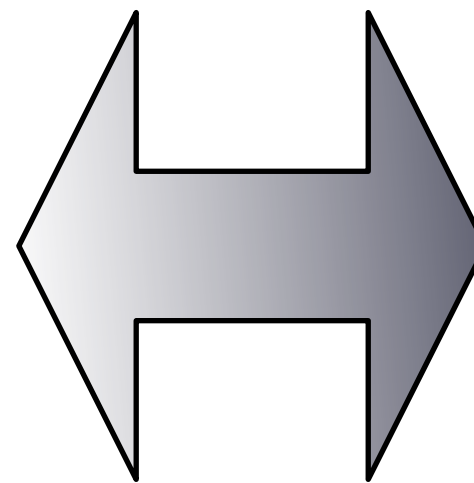
$M_1$



$M_2$

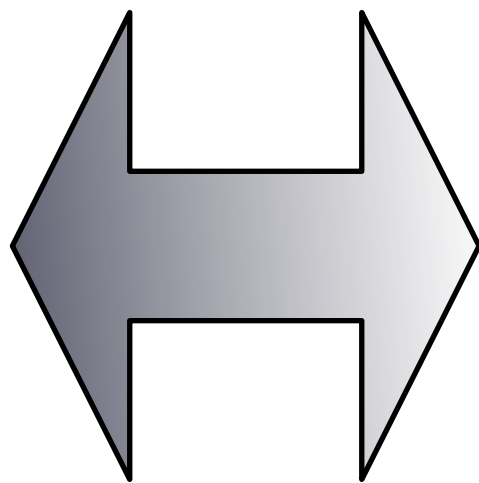
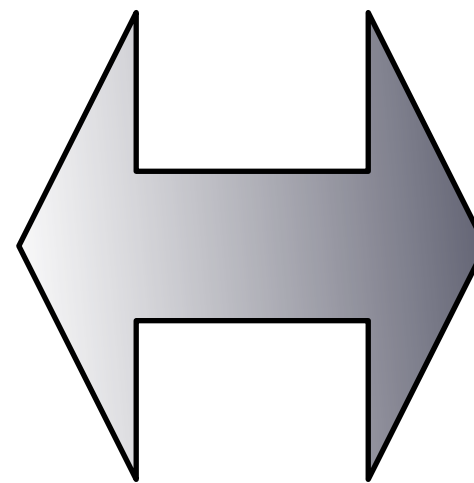


# Geometric Matching

 $M_1$  $M_2$ 

$$M_1 \approx T(M_2)$$

# Geometric Matching

 $M_1$  $M_2$ 

$$M_1 \approx T(M_2)$$

symmetry detection

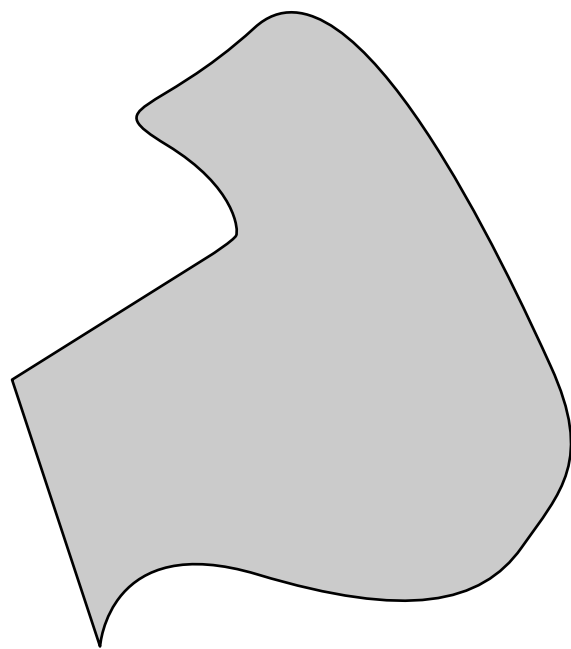
$$M \approx T(M)$$

# Geometric Matching

---

# Geometric Matching

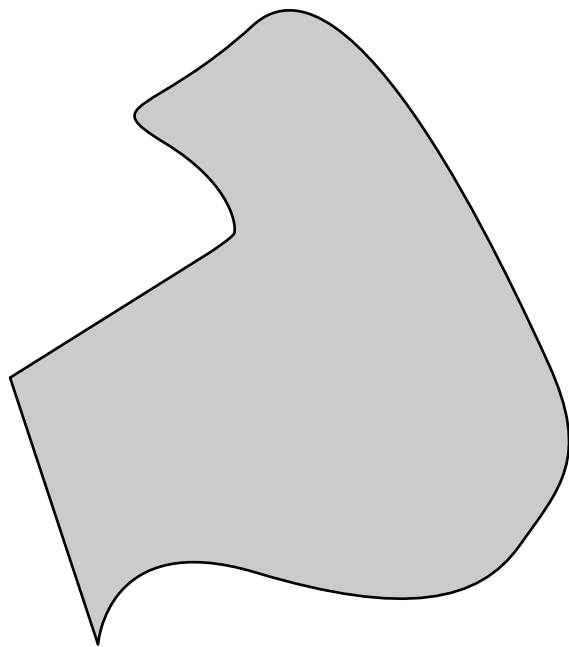
$M_1$



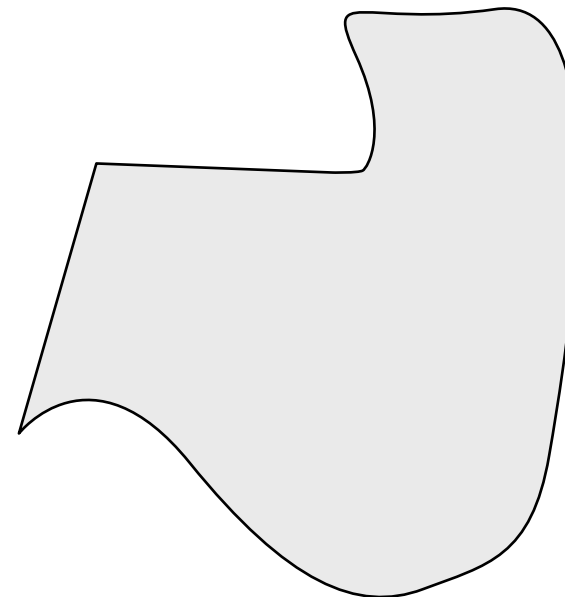


# Geometric Matching

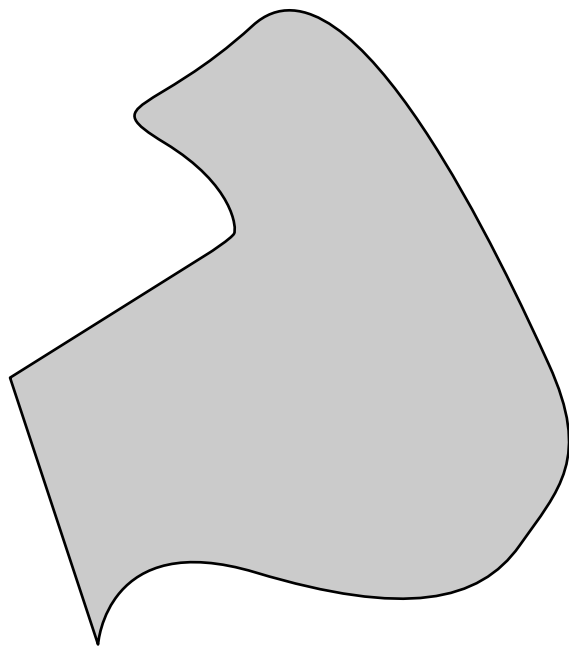
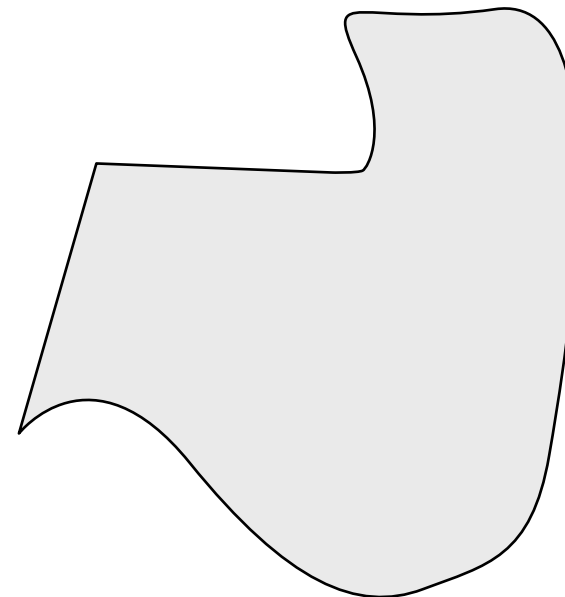
$M_1$



$M_2$



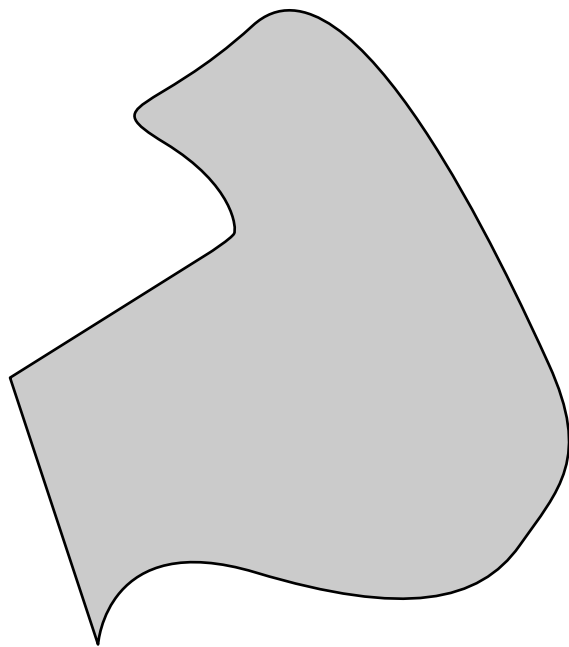
# Geometric Matching

 $M_1$  $M_2$ 

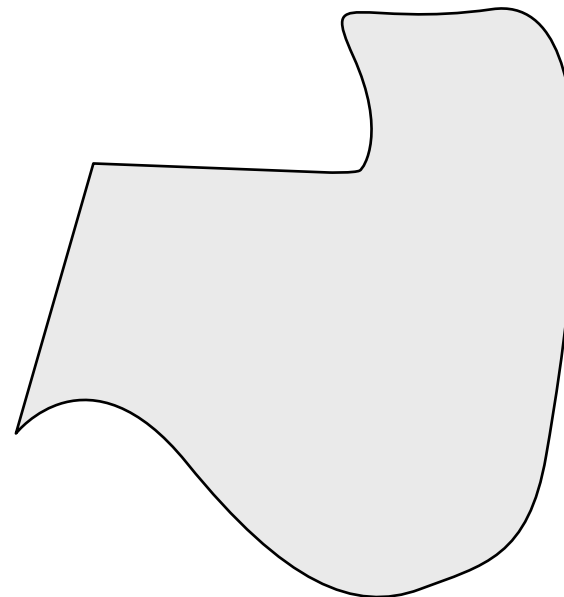
$$M_1 \approx T(M_2)$$

# Matching with Translation

$M_1$

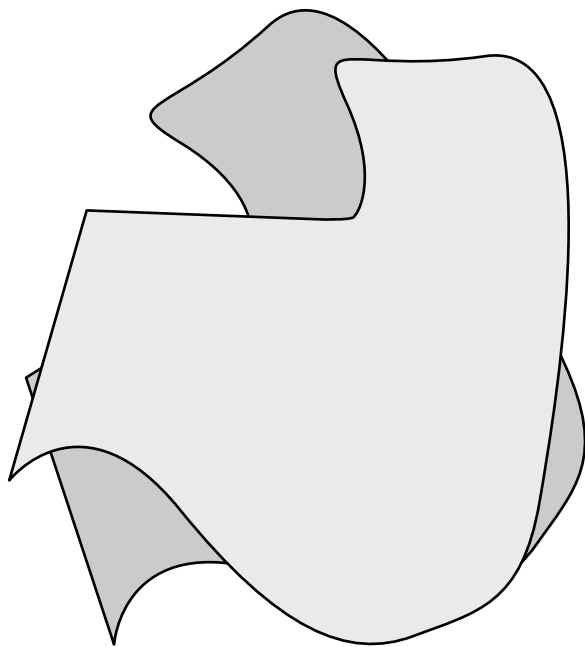


$M_2$



$$M_1 \approx T(M_2)$$

$T$  : translation

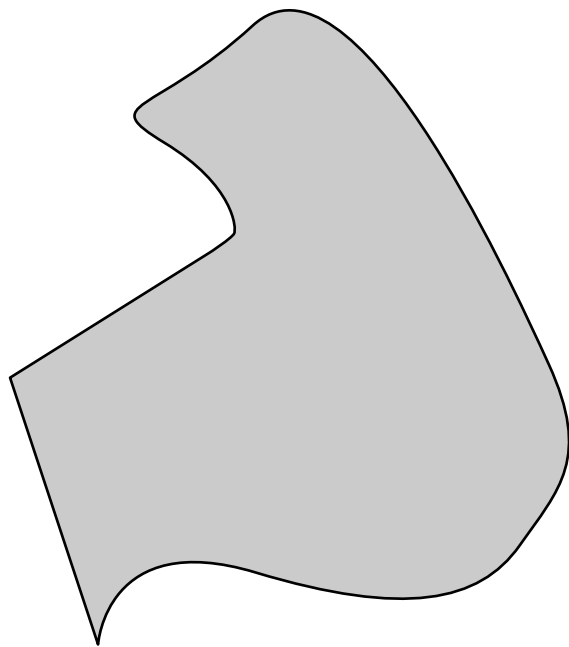
$M_1$  $M_2$ 

$$M_1 \approx T(M_2)$$

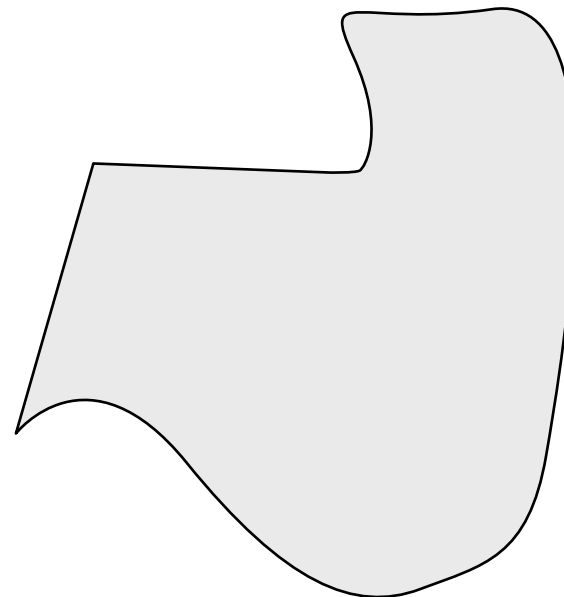
$T$  : translation

# Matching with Rigid Transforms

$M_1$



$M_2$



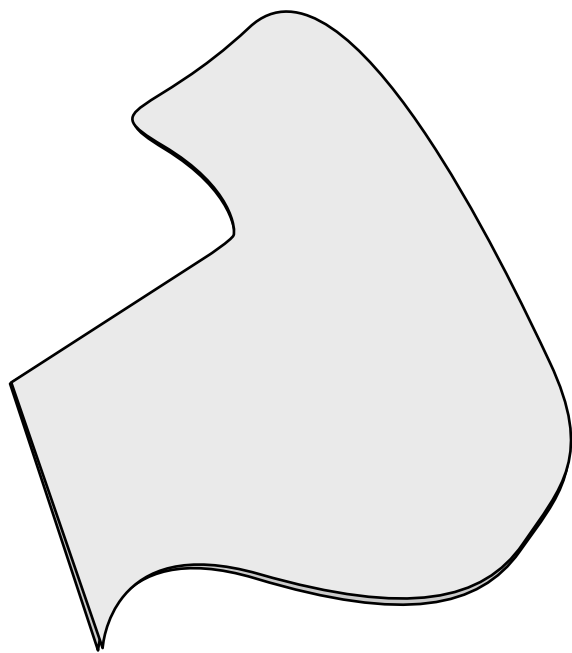
$$M_1 \approx T(M_2)$$

$T$  : translation + rotation

# Matching with Rigid Transforms

$M_1$

$M_2$

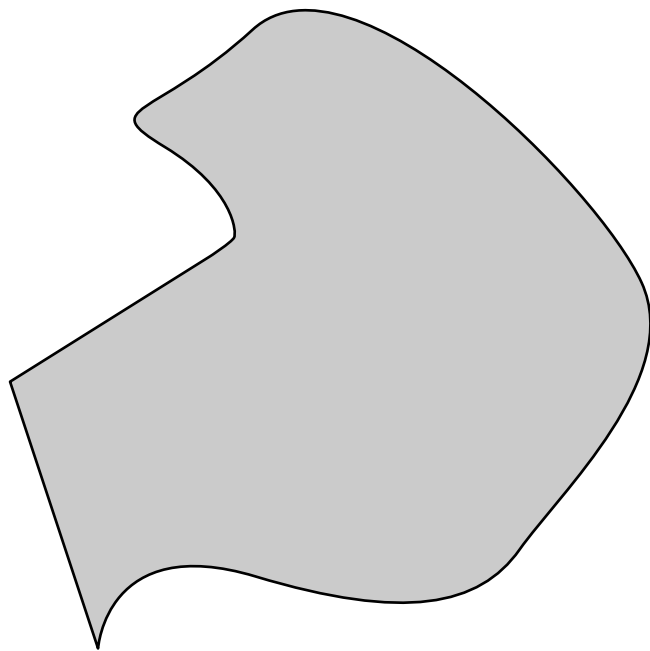
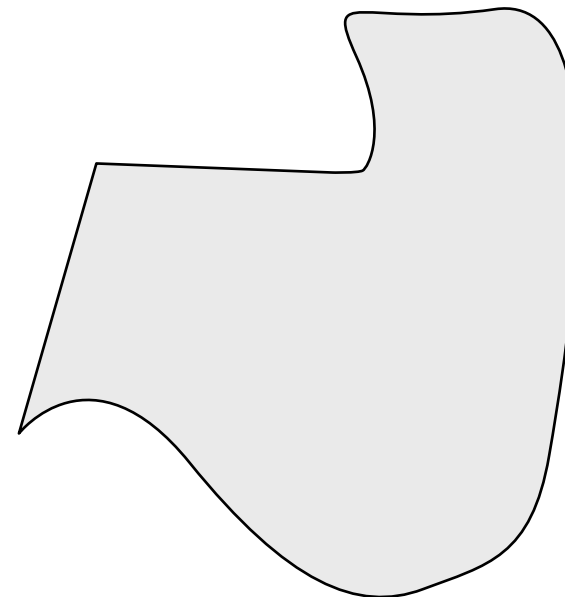


$$M_1 \approx T(M_2)$$

$T$  : translation + rotation



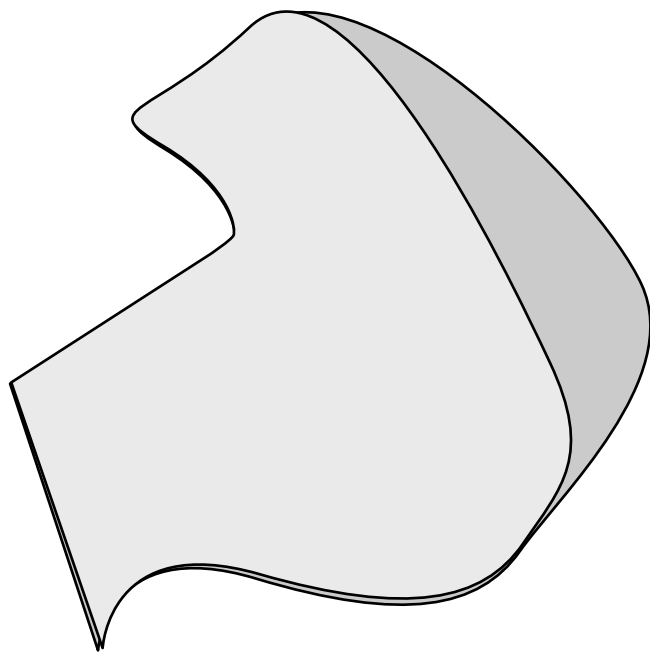
# Partial Matching

 $M_1$  $M_2$ 

$$M_1 \approx T(M_2)$$

$T$  : translation + rotation

# Partial Matching

 $M_1$  $M_2$ 

$$M_1 \approx T(M_2)$$

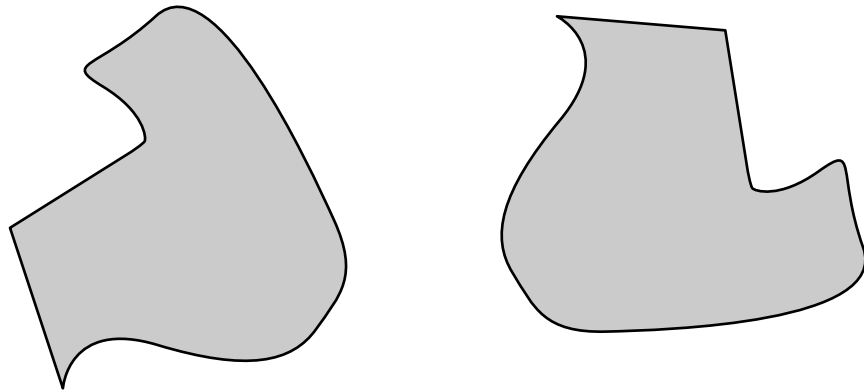
$T$  : translation + rotation

# Local vs. Global Matching

---



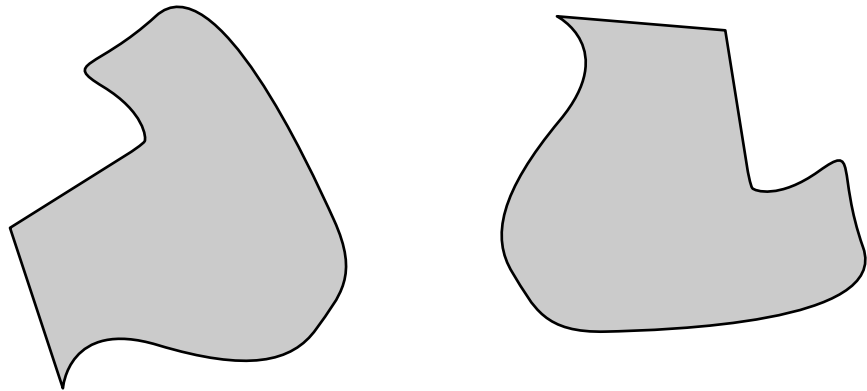
# Local vs. Global Matching



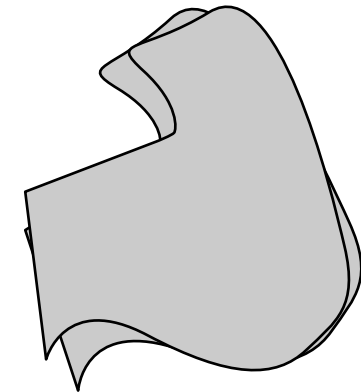
***global registration***

any rigid transform

# Local vs. Global Matching



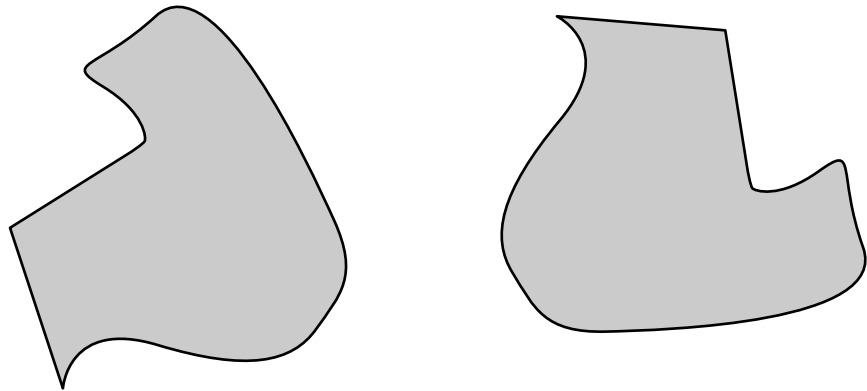
***global registration***  
any rigid transform



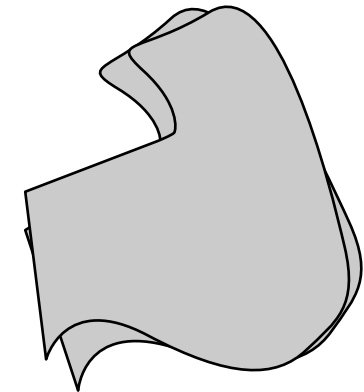
***local registration***  
nearly aligned



# Local vs. Global Matching



***global registration***  
any rigid transform



***local registration***  
nearly aligned

Given  $M_1, \dots, M_n$ , find  $T_2, \dots, T_n$  such that

$$M_1 \approx T_2(M_2) \cdots \approx T_n(M_n)$$

# ICP: Local, Partial, Rigid Transforms

---

- How many point-pairs are needed to *uniquely* define a rigid transform?

# ICP: Local, Partial, Rigid Transforms

---

- How many point-pairs are needed to *uniquely* define a rigid transform?

$$\mathbf{p}_1 \rightarrow \mathbf{q}_1$$

$$\mathbf{p}_2 \rightarrow \mathbf{q}_2$$

$$\mathbf{p}_3 \rightarrow \mathbf{q}_3$$

# ICP: Local, Partial, Rigid Transforms

---

- How many point-pairs are needed to *uniquely* define a rigid transform?

$$\mathbf{p}_1 \rightarrow \mathbf{q}_1$$

$$\mathbf{p}_2 \rightarrow \mathbf{q}_2$$

$$\mathbf{p}_3 \rightarrow \mathbf{q}_3$$

$$R\mathbf{p}_i + t \approx \mathbf{q}_i$$

# ICP: Local, Partial, Rigid Transforms

---

- How many point-pairs are needed to *uniquely* define a rigid transform?

$$\mathbf{p}_1 \rightarrow \mathbf{q}_1$$

$$\mathbf{p}_2 \rightarrow \mathbf{q}_2$$

$$\mathbf{p}_3 \rightarrow \mathbf{q}_3$$

$$R\mathbf{p}_i + t \approx \mathbf{q}_i$$

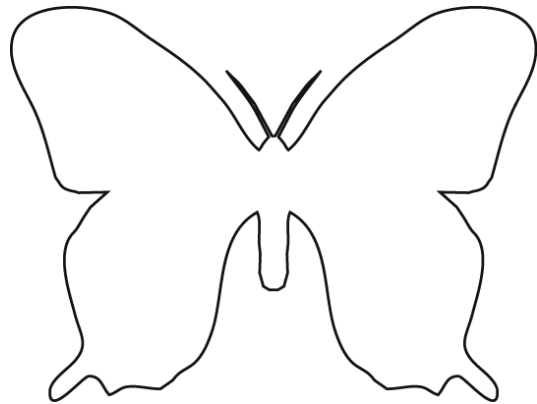
***Correspondence  
problem:***

$$\mathbf{p}_i \overset{?}{\rightarrow} \mathbf{q}_j$$

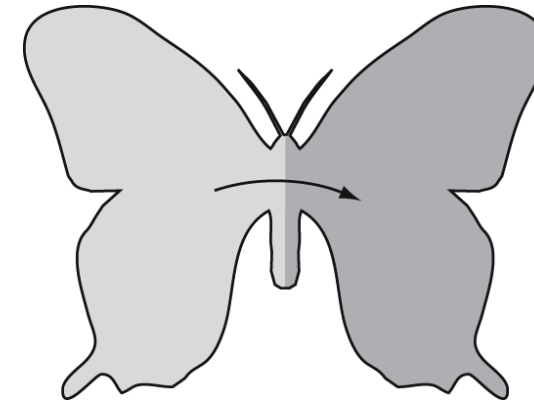
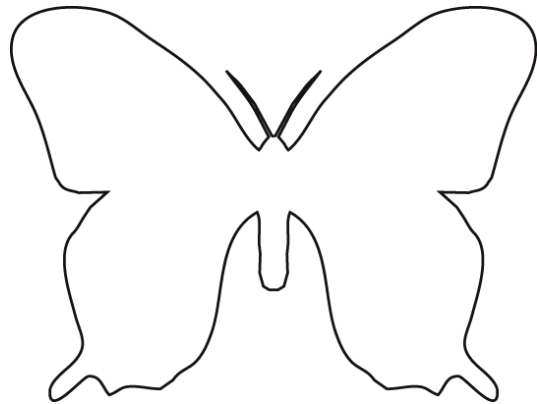


# Types of Symmetry

---

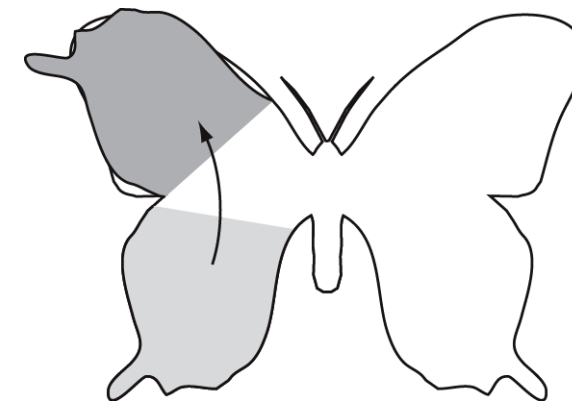
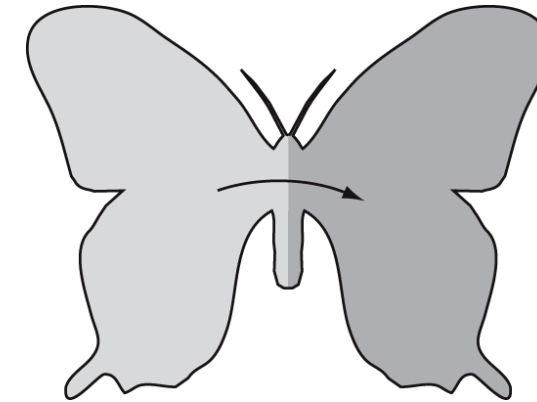
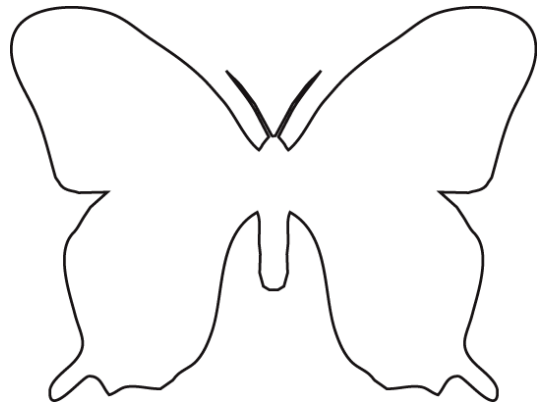


# Types of Symmetry



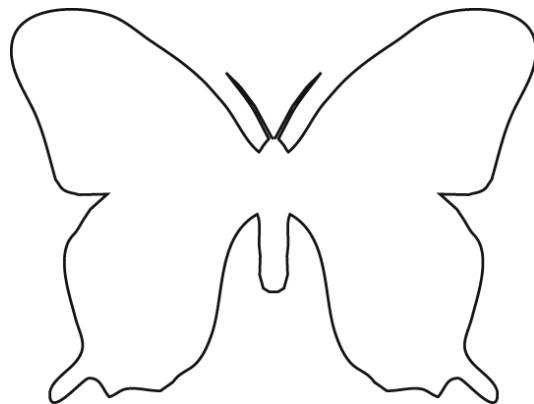
- Reflection

# Types of Symmetry

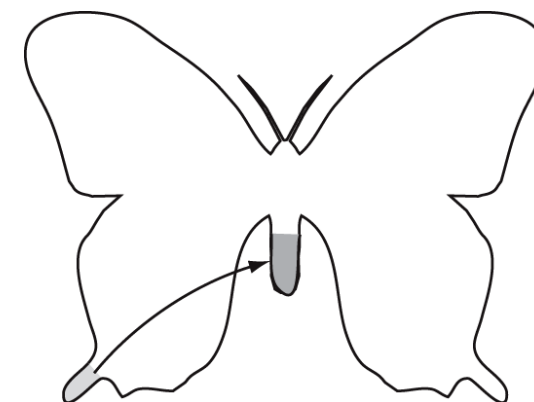
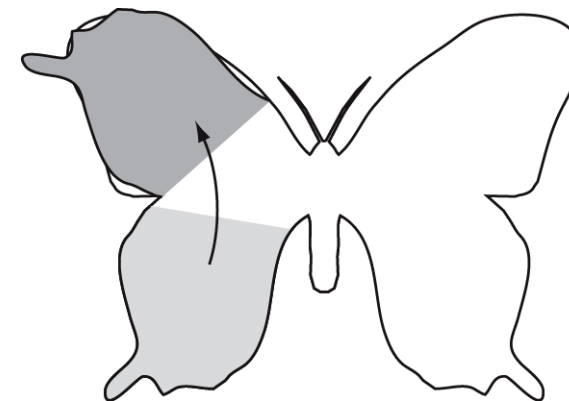
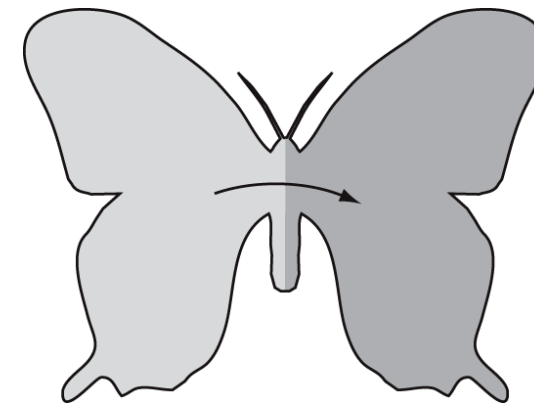


- Reflection
- Rotation + translation

# Types of Symmetry



- Reflection
- Rotation + translation
- Uniform scaling



# Typical Stages

---

# Typical Stages

---

- Feature selection

$$\mathcal{F}(M) = \mathcal{F}(T(M))$$



# Typical Stages

---

- Feature selection

$$\mathcal{F}(M) = \mathcal{F}(T(M))$$

- Aggregation

# Typical Stages

---

- Feature selection

$$\mathcal{F}(M) = \mathcal{F}(T(M))$$

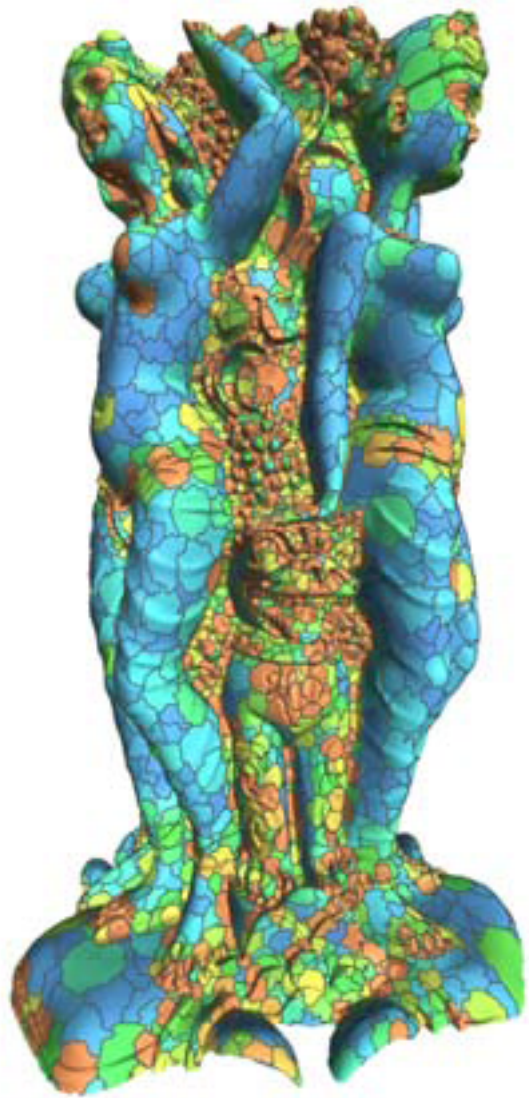
- Aggregation

- Extraction

# Type of Features

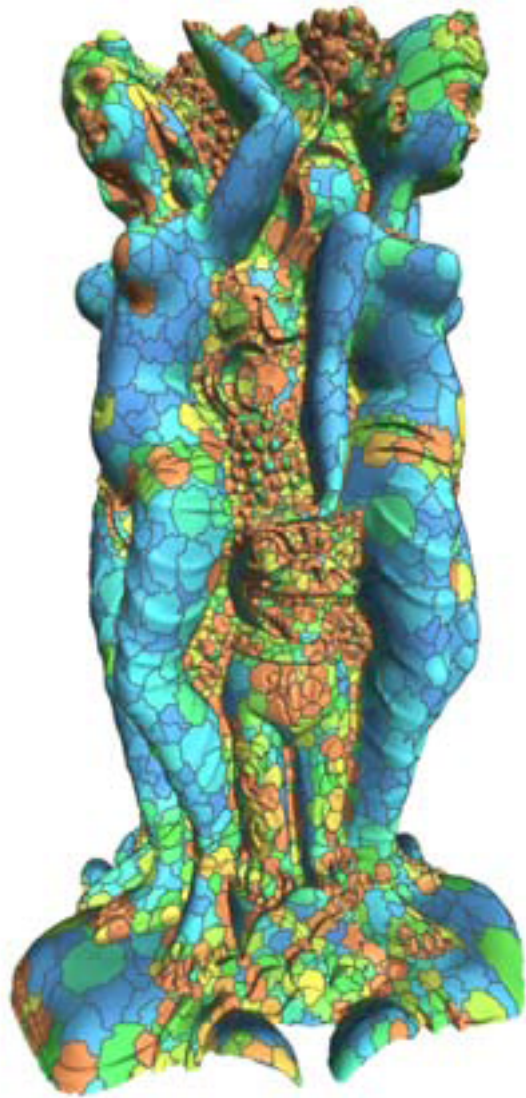
---

# Type of Features



surface curvature  
[Gal et al. 2006]

# Type of Features



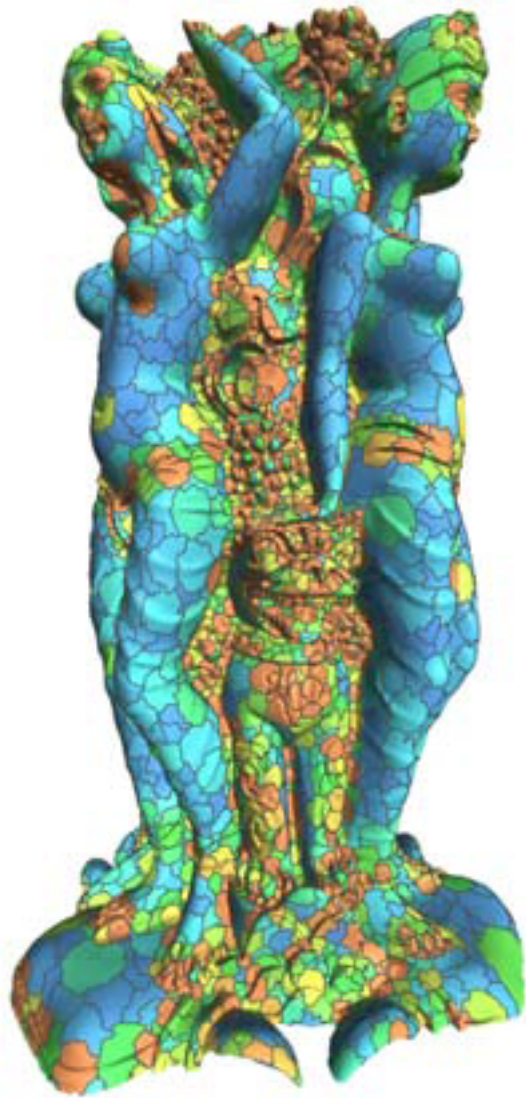
surface curvature  
[Gal et al. 2006]



line features  
[Bokeloh et al. 2009]



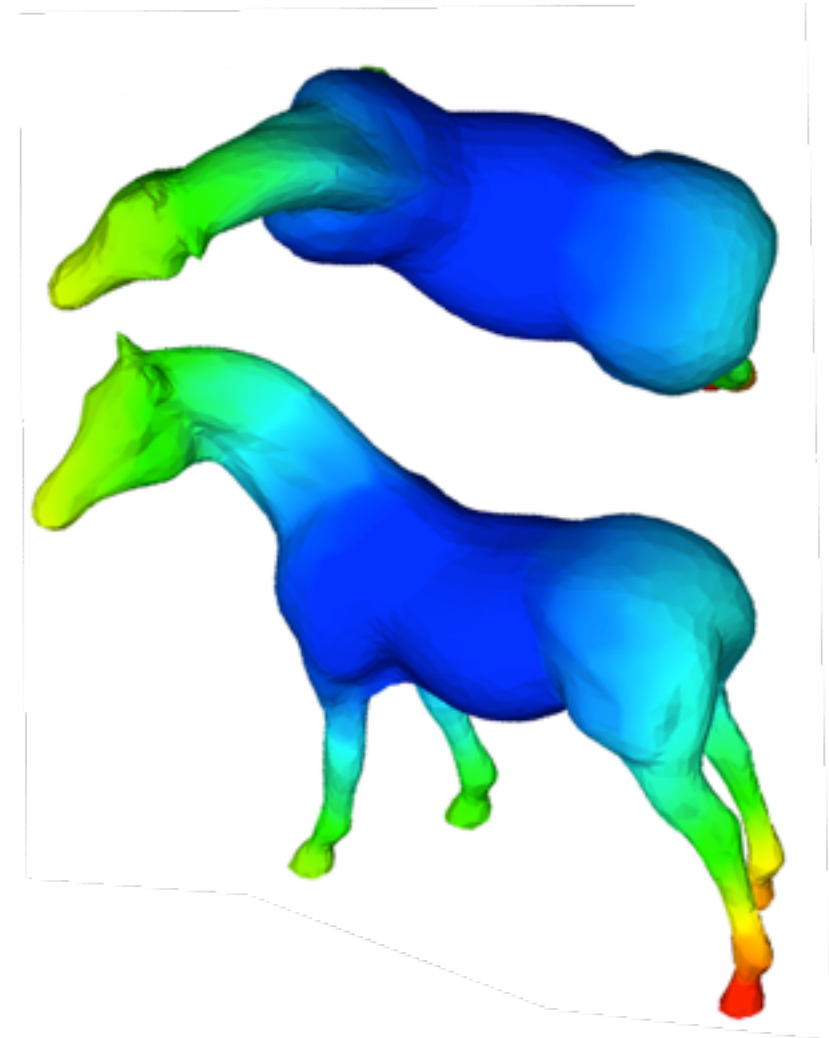
# Type of Features



surface curvature  
[Gal et al. 2006]



line features  
[Bokeloh et al. 2009]



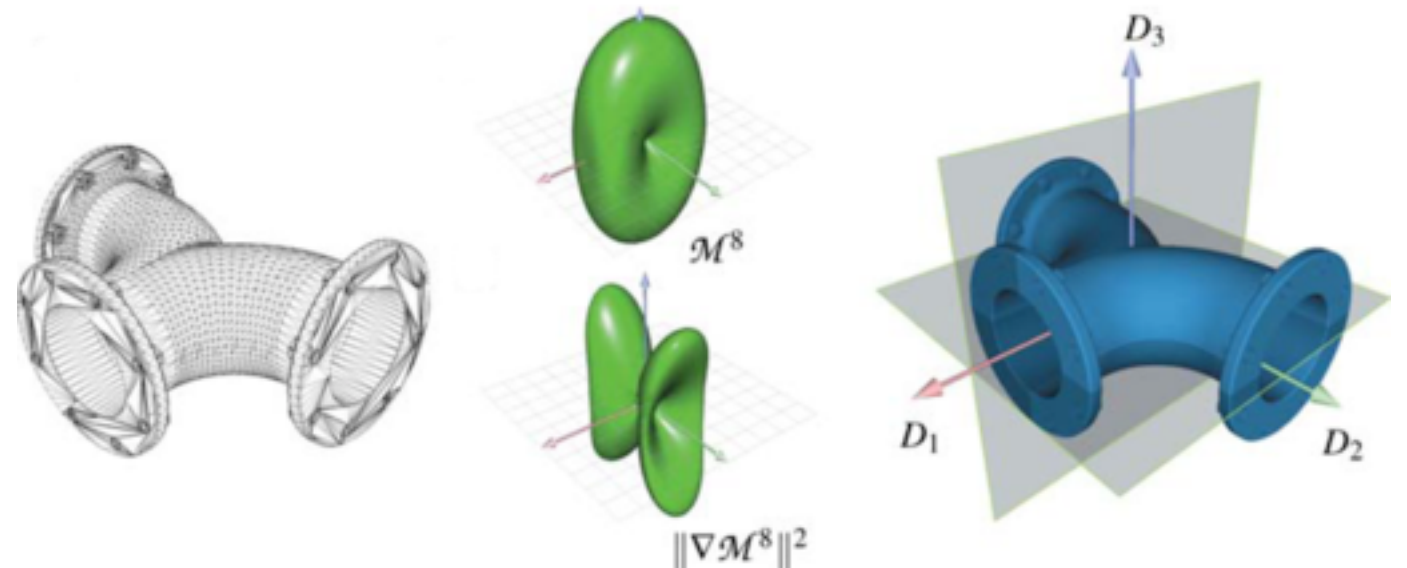
geodesic distances  
[Kim et al. 2010]

# Type of Features

---

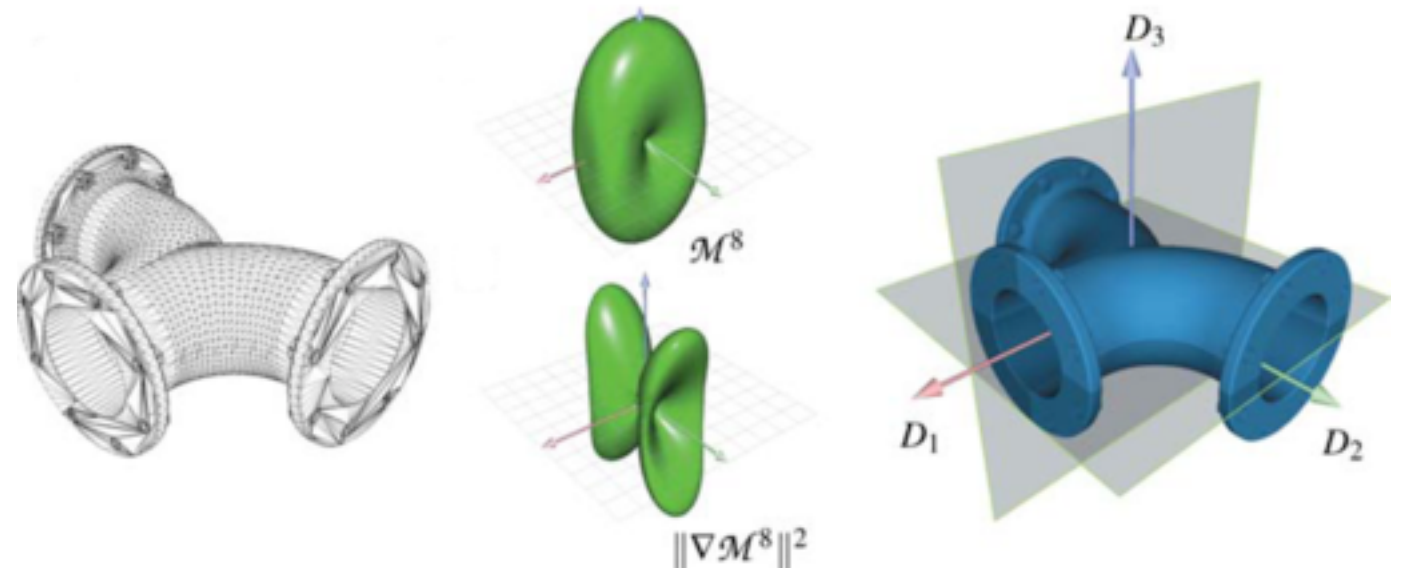


# Type of Features

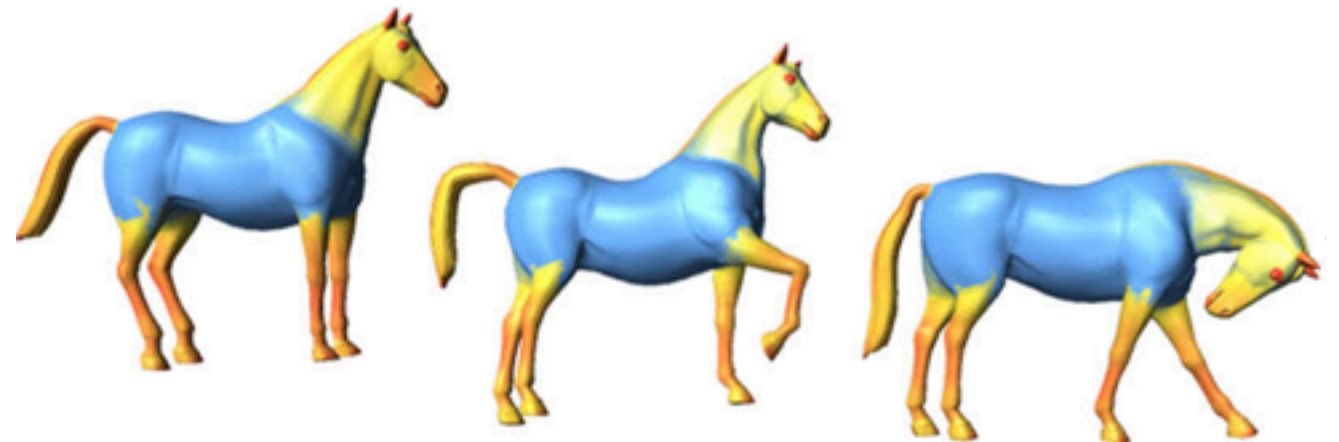


generalized even moments  
[Martinet et al. 2006]

# Type of Features

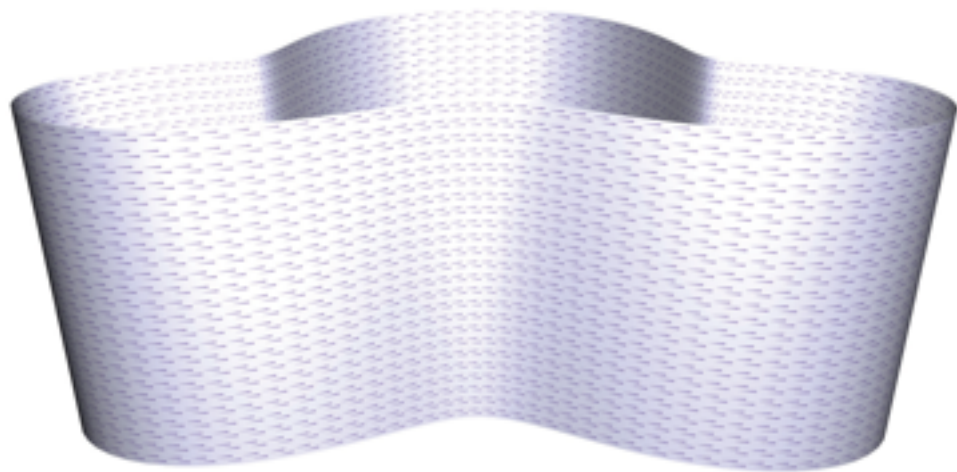


generalized even moments  
[Martinet et al. 2006]

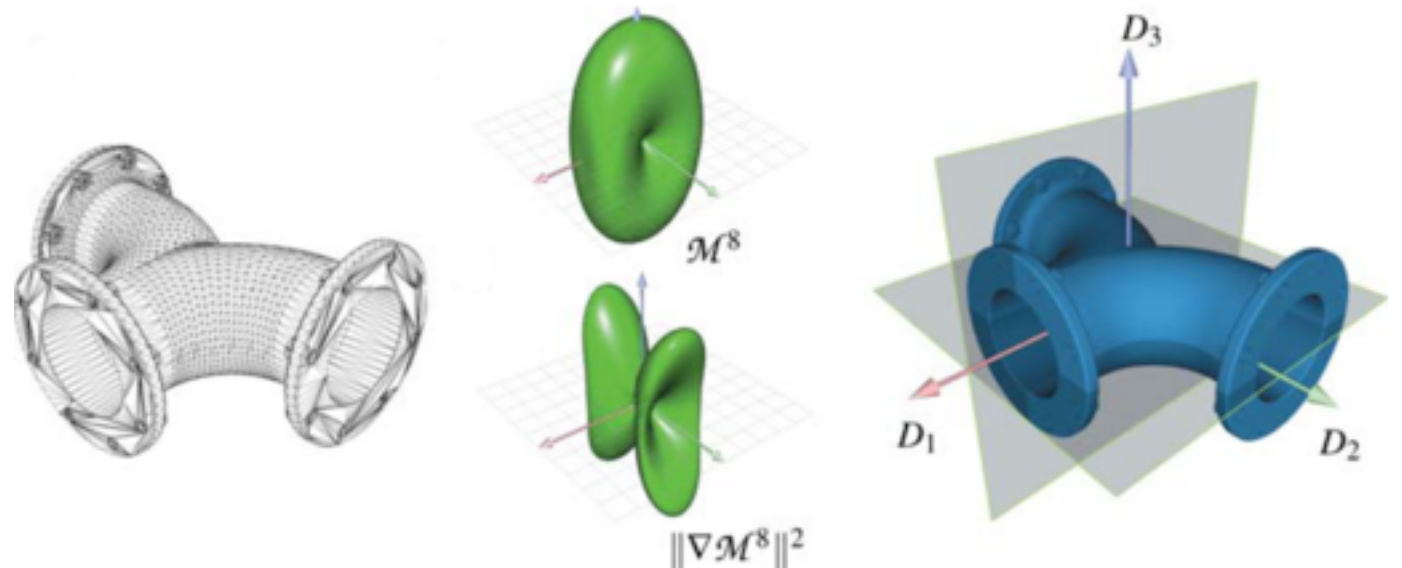


shape diameter functions (SDF)  
[Shapira et al. 2008]

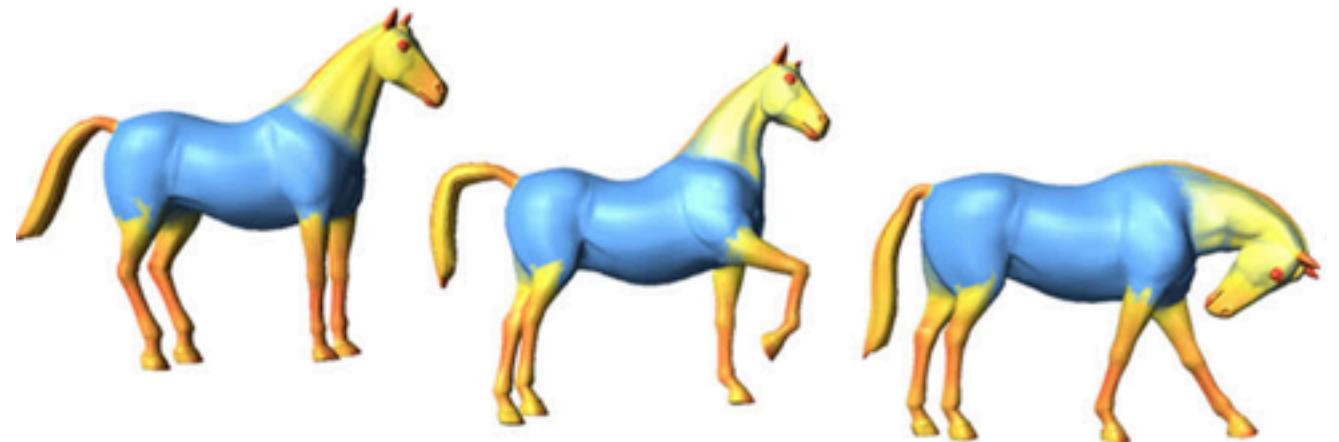
# Type of Features



Killing vector fields  
[Ben-Chen et al. 2010]



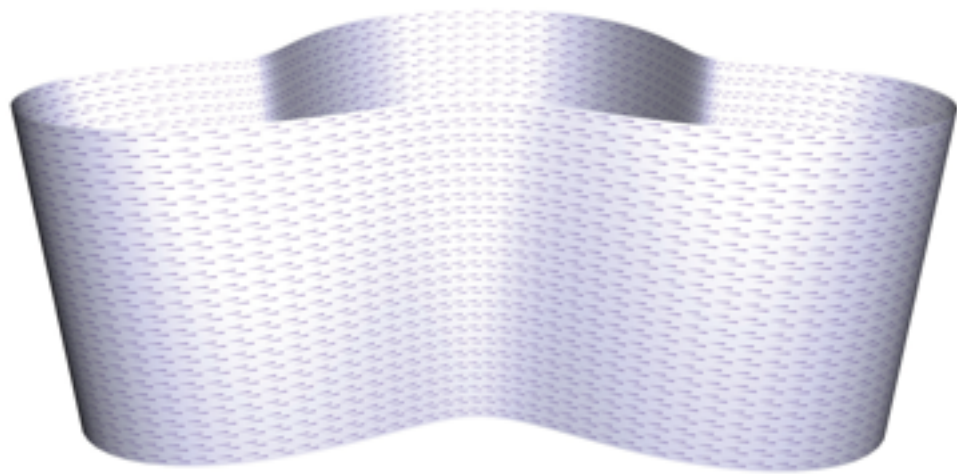
generalized even moments  
[Martinet et al. 2006]



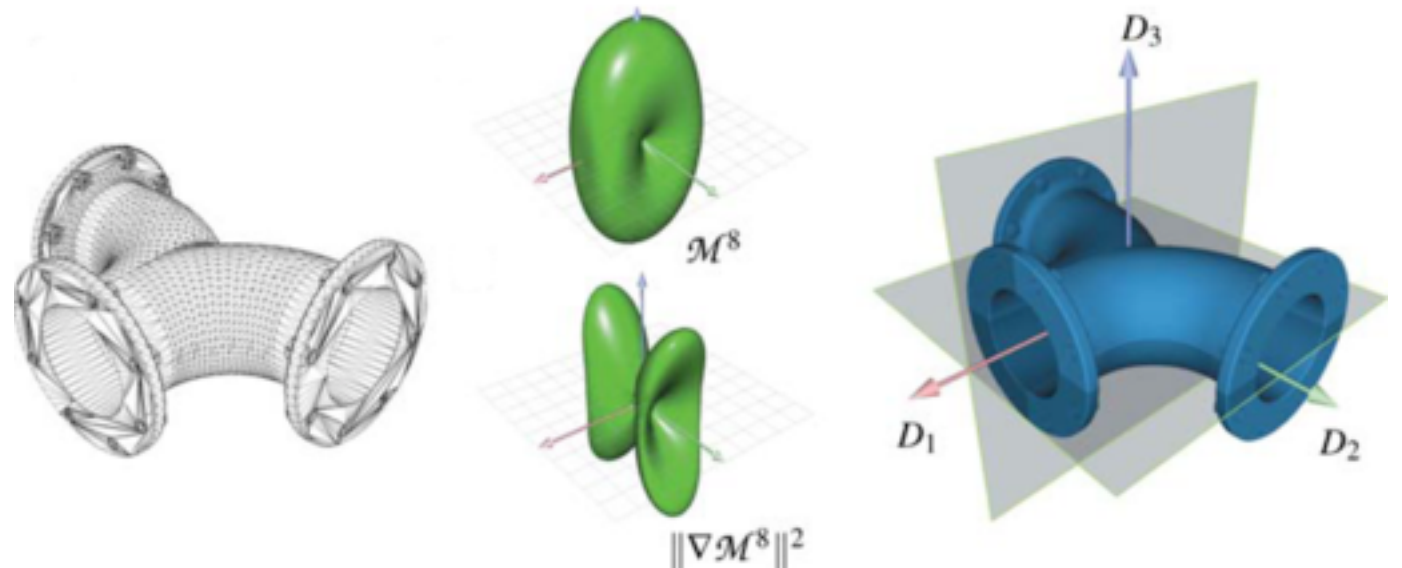
shape diameter functions (SDF)  
[Shapira et al. 2008]



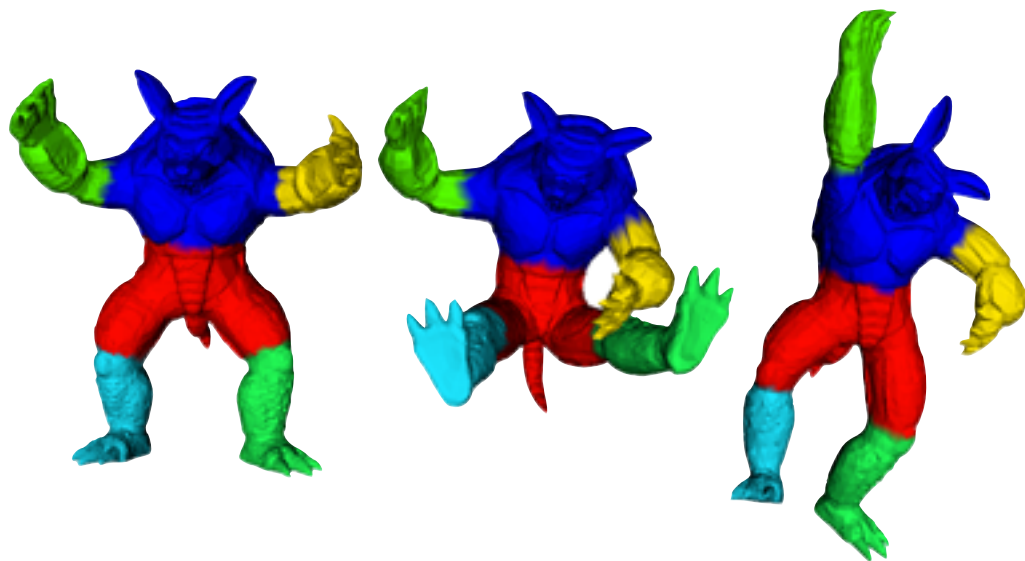
# Type of Features



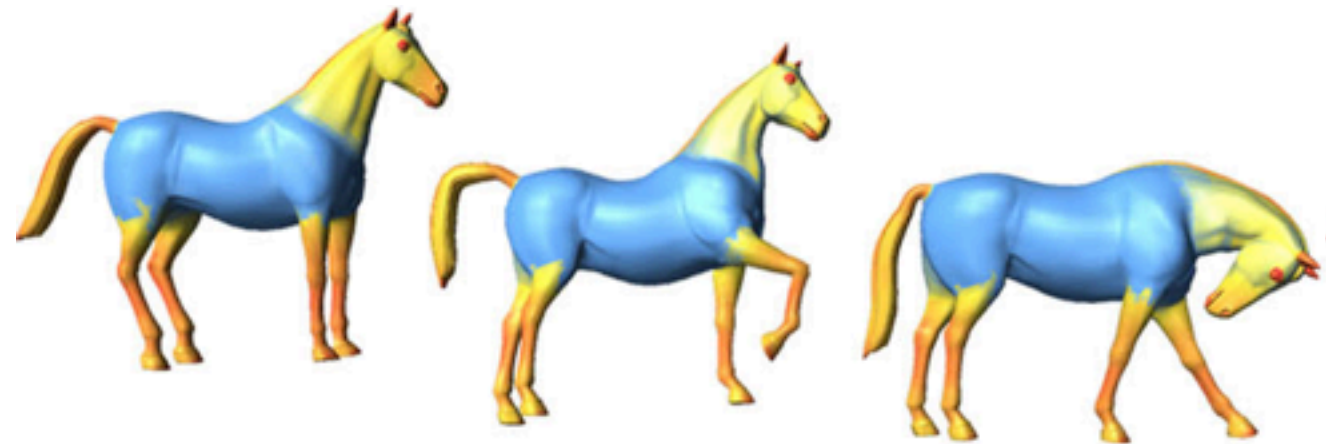
Killing vector fields  
[Ben-Chen et al. 2010]



generalized even moments  
[Martinet et al. 2006]

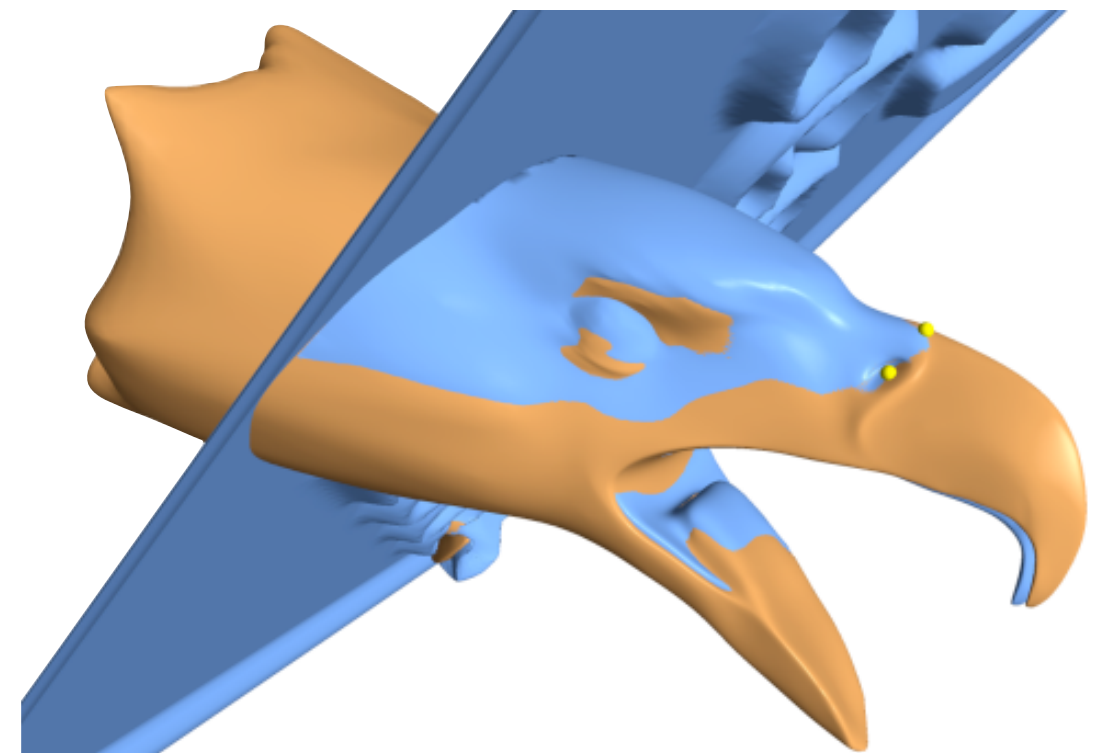


global point signatures (GPS)  
[Rustamov 2007]



shape diameter functions (SDF)  
[Shapira et al. 2008]

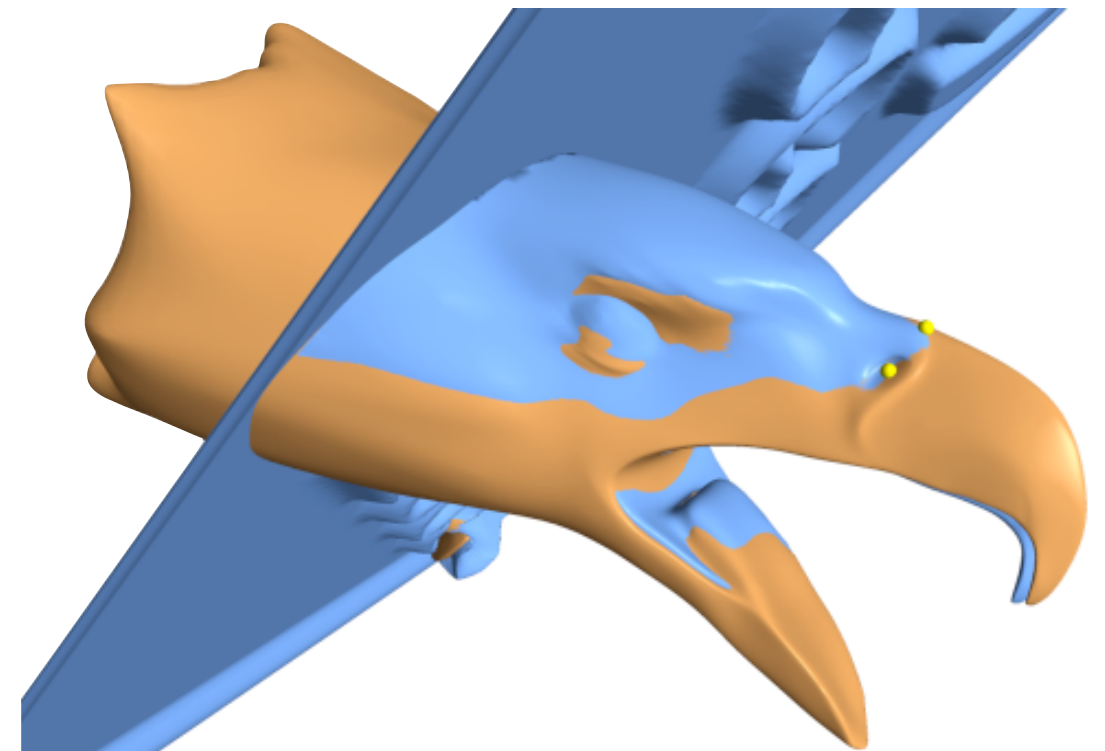
# Geometric Hashing



[Gal et al., TOG 2006]

# Geometric Hashing

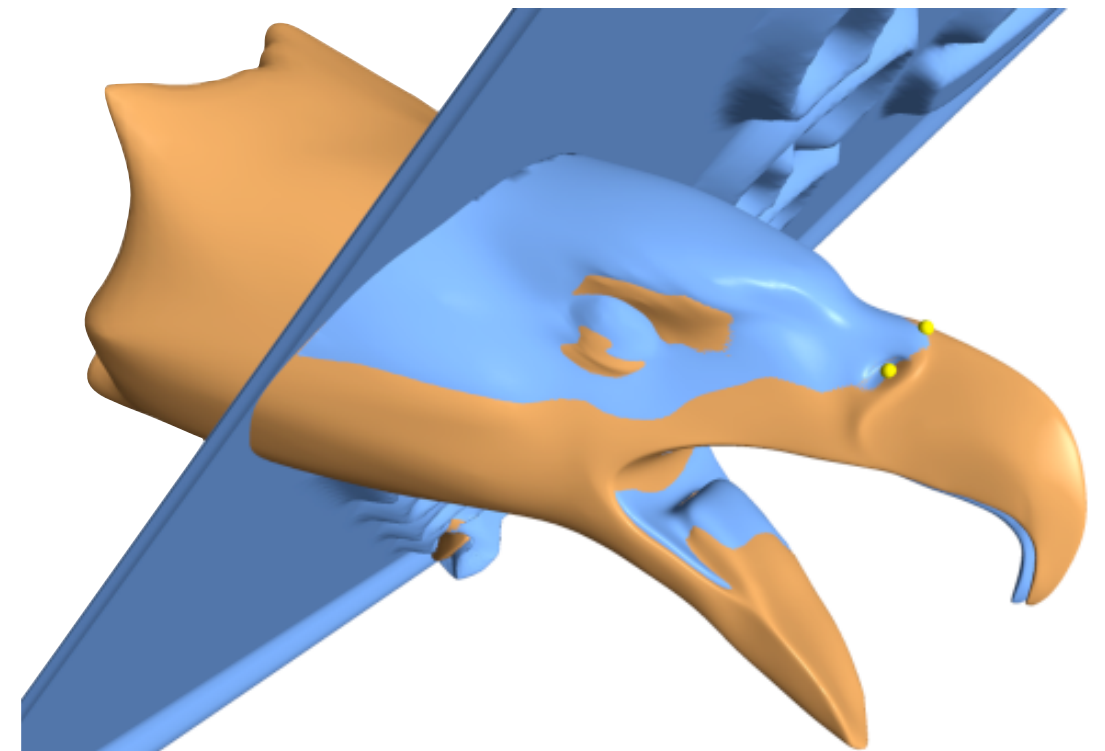
- ***Features:*** quadratic patch parameters



[Gal et al., TOG 2006]

# Geometric Hashing

- ***Features:*** quadratic patch parameters

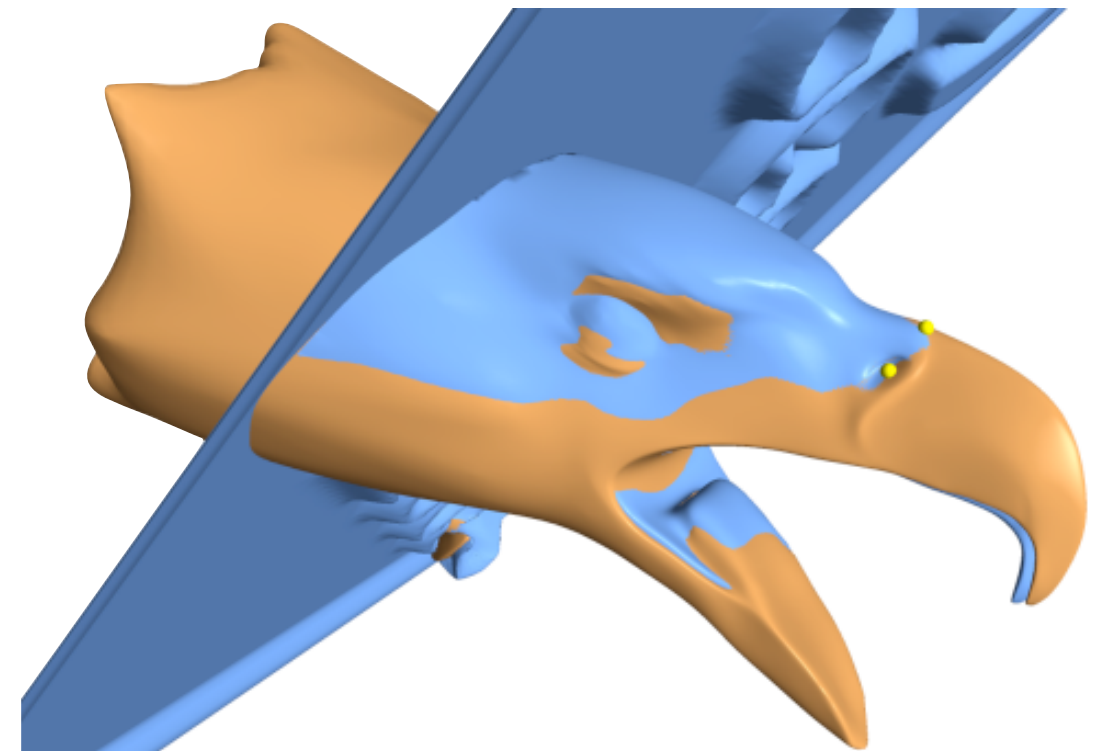


[Gal et al., TOG 2006]



# Geometric Hashing

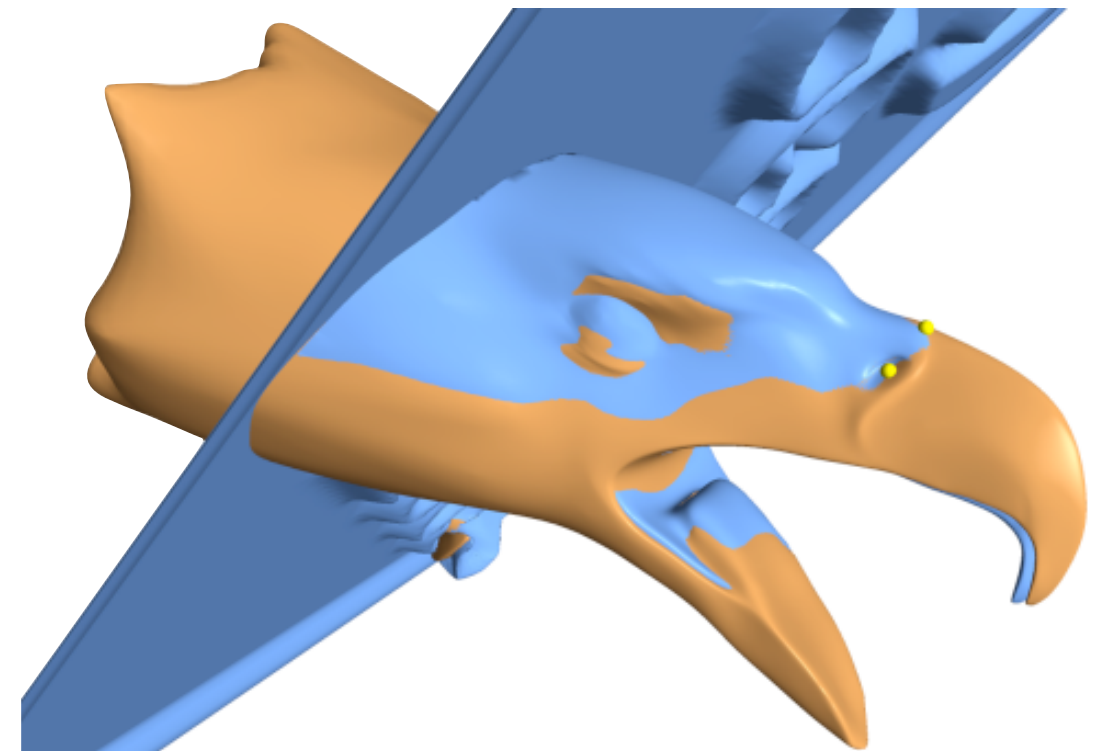
- **Features:** quadratic patch parameters
- **Aggregation:** geometric hashing



[Gal et al., TOG 2006]

# Geometric Hashing

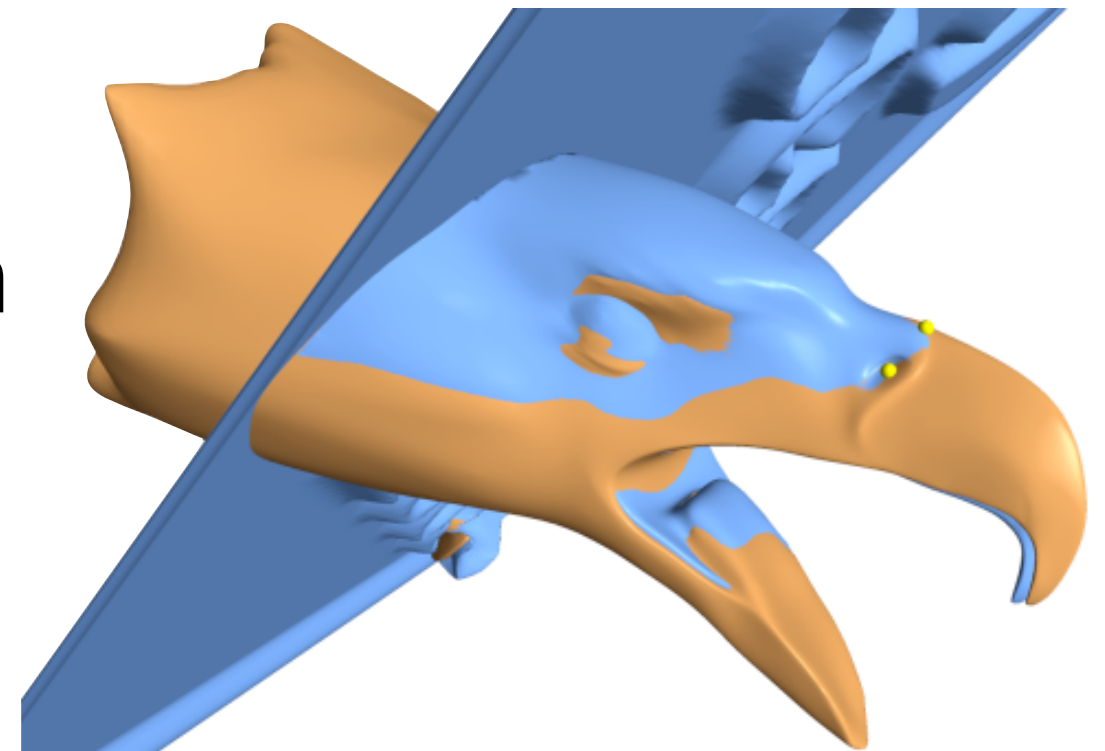
- **Features:** quadratic patch parameters
- **Aggregation:** geometric hashing



[Gal et al., TOG 2006]

# Geometric Hashing

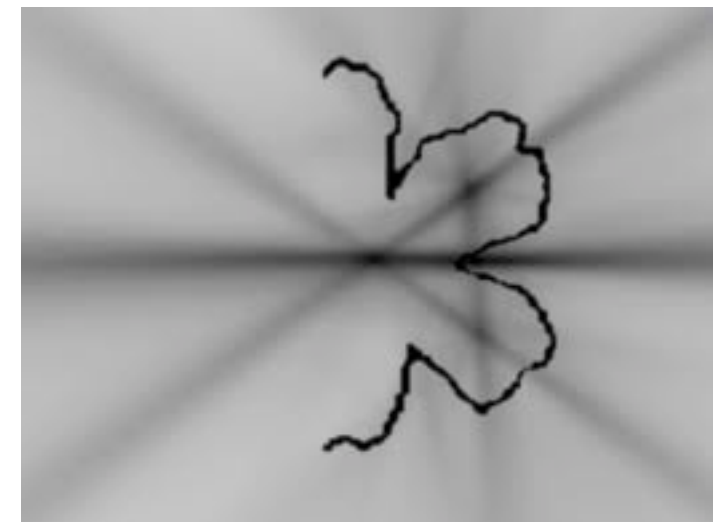
- **Features:** quadratic patch parameters
- **Aggregation:** geometric hashing
- **Extraction:** pre-segmentation



[Gal et al., TOG 2006]

# Symmetry Transform

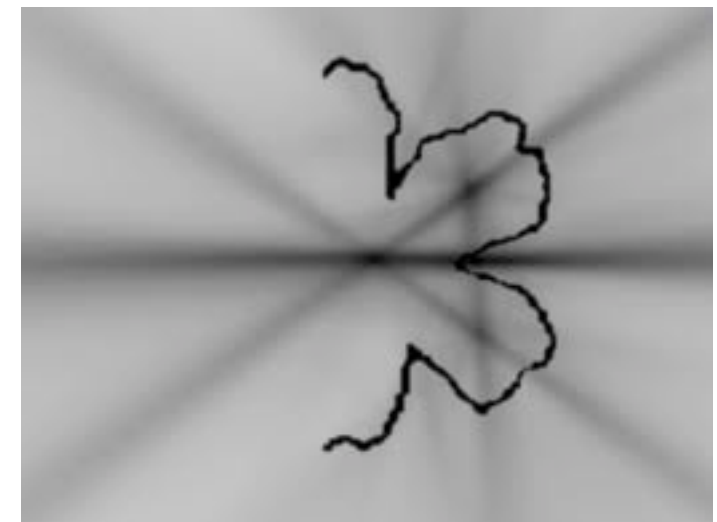
---



[Podolak et al., Siggraph 2006]

# Symmetry Transform

- ***Features:***

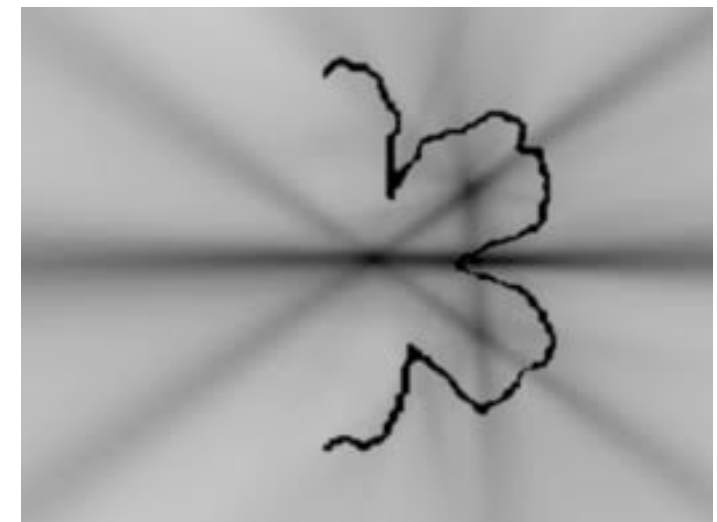


[Podolak et al., Siggraph 2006]

# Symmetry Transform

---

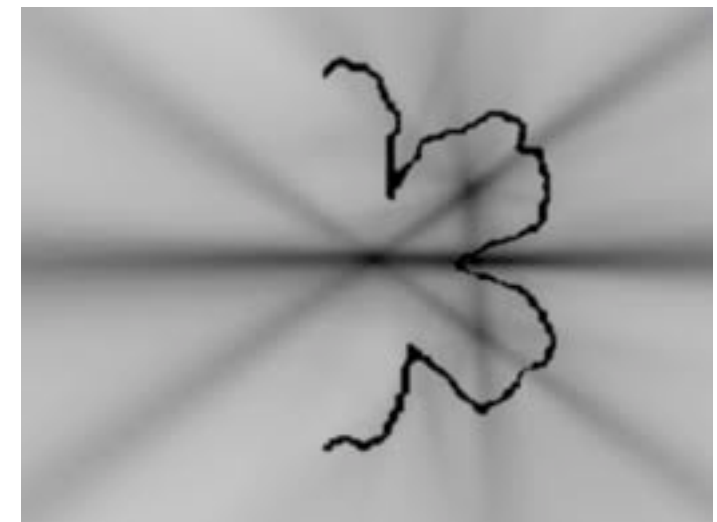
- ***Features:***



[Podolak et al., Siggraph 2006]

# Symmetry Transform

- **Features:**
- **Aggregation:** FFT in transform domain

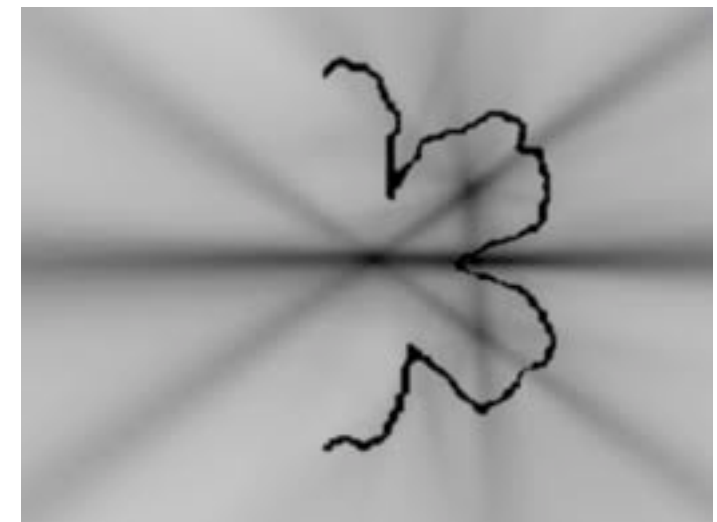


[Podolak et al., Siggraph 2006]



# Symmetry Transform

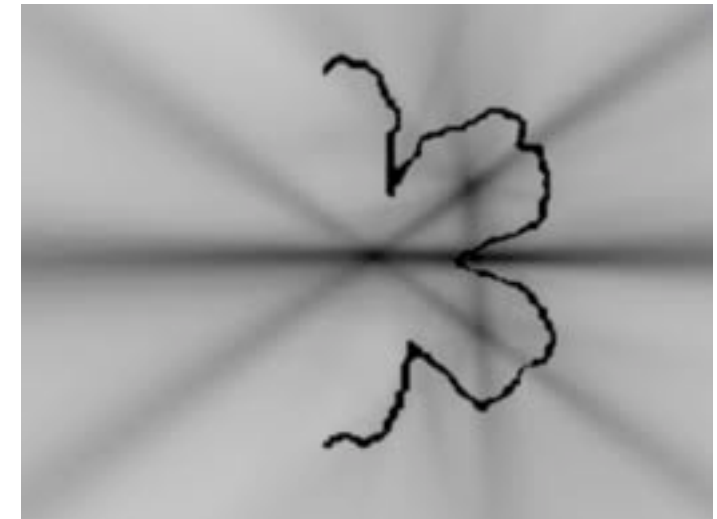
- **Features:**
- **Aggregation:** FFT in transform domain



[Podolak et al., Siggraph 2006]

# Symmetry Transform

- **Features:**
- **Aggregation:** FFT in transform domain
- **Extraction:** clustering, region growing



[Podolak et al., Siggraph 2006]

# Transform Domain Analysis

---



[Mitra et al., Siggraph 2006]

# Transform Domain Analysis

---

- ***Features:*** curvatures



[Mitra et al., Siggraph 2006]

# Transform Domain Analysis

---

- ***Features:*** curvatures



[Mitra et al., Siggraph 2006]

# Transform Domain Analysis

---

- **Features:** curvatures
- **Aggregation:** transform domain analysis



[Mitra et al., Siggraph 2006]

# Transform Domain Analysis

---

- **Features:** curvatures
- **Aggregation:** transform domain analysis



[Mitra et al., Siggraph 2006]



# Transform Domain Analysis

---

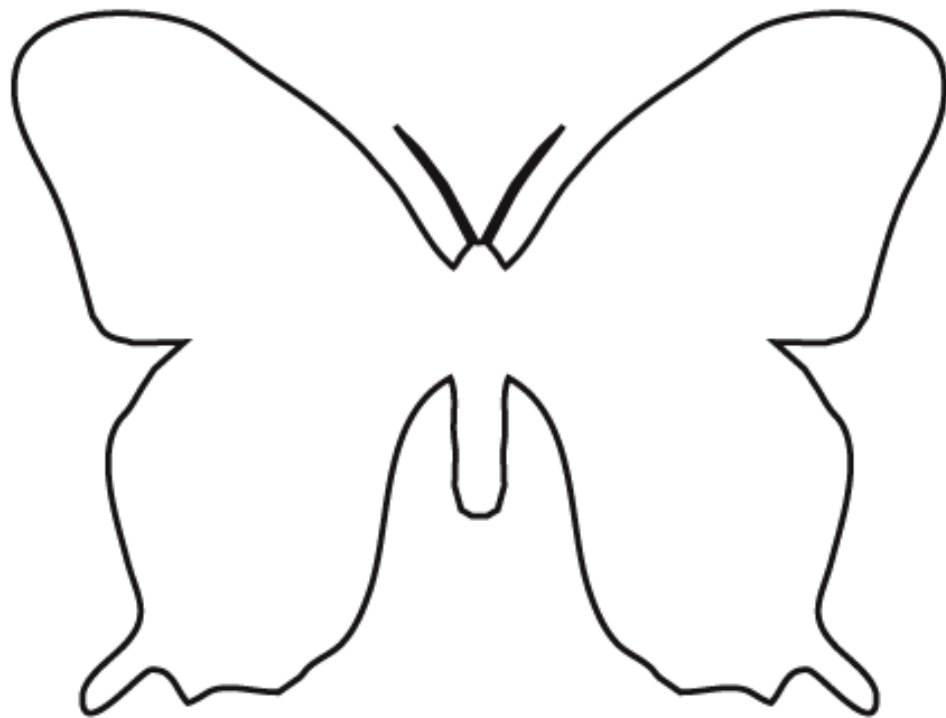
- ***Features:*** curvatures
- ***Aggregation:*** transform domain analysis
- ***Extraction:*** region growing



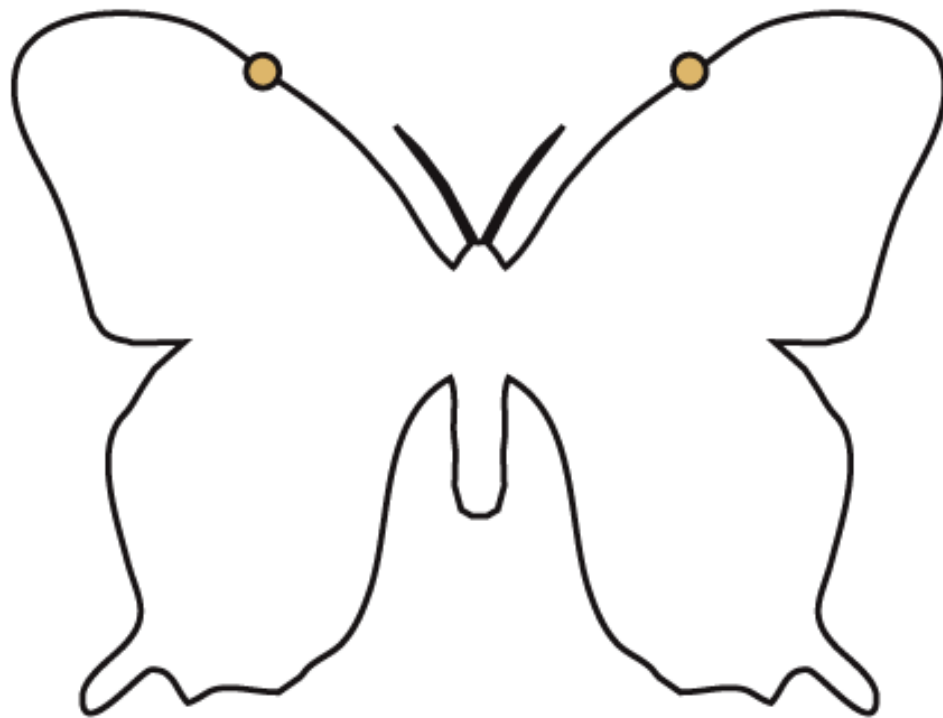
[Mitra et al., Siggraph 2006]

# Reflective Symmetry

---

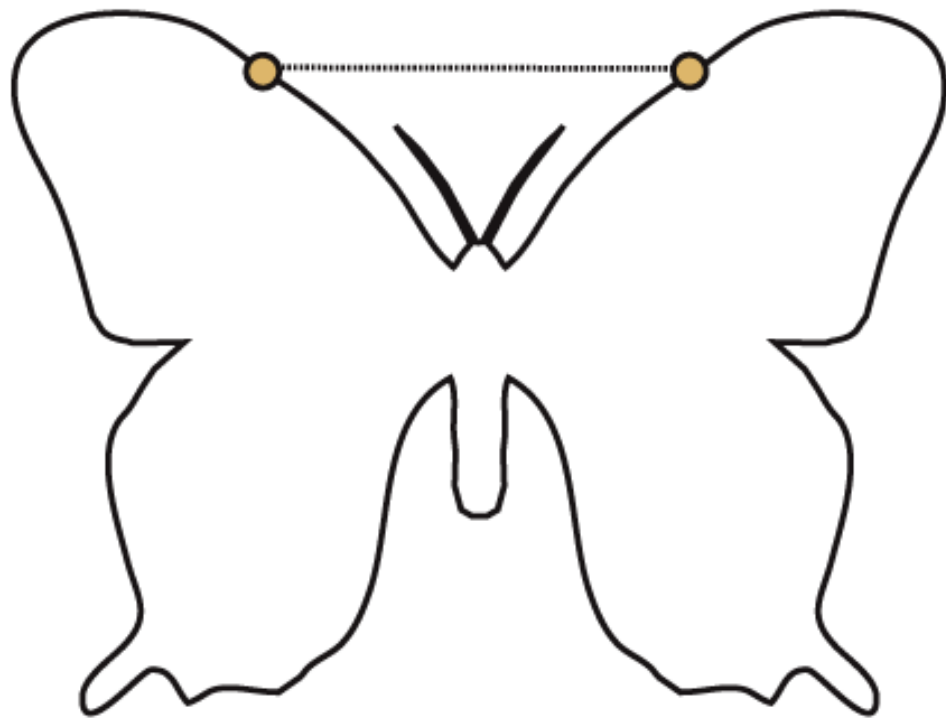


# Reflective Symmetry



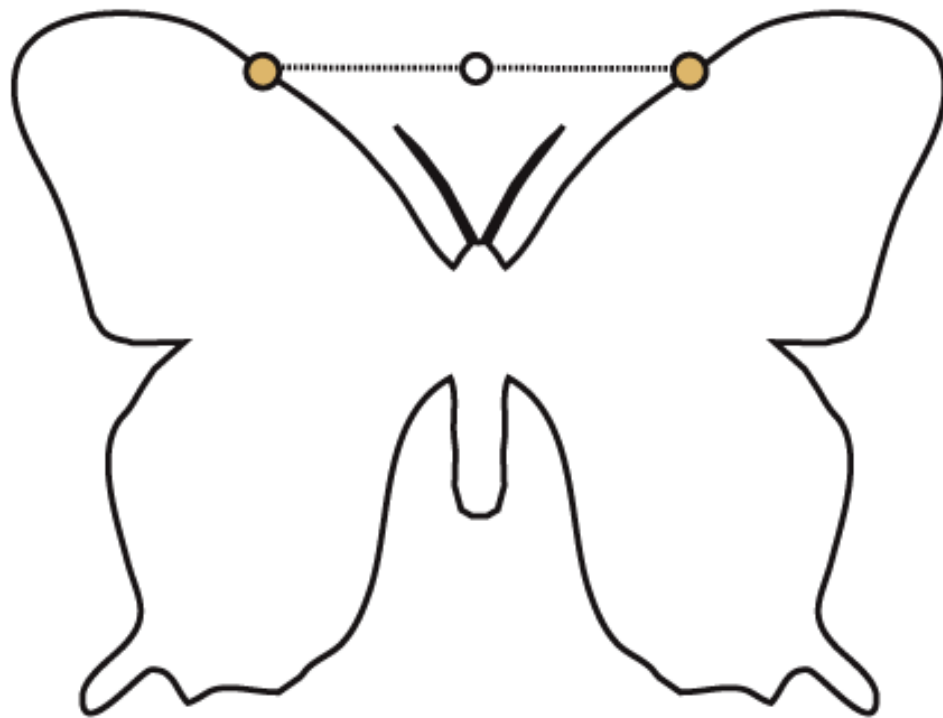
# Reflective Symmetry: A Pair Votes

---



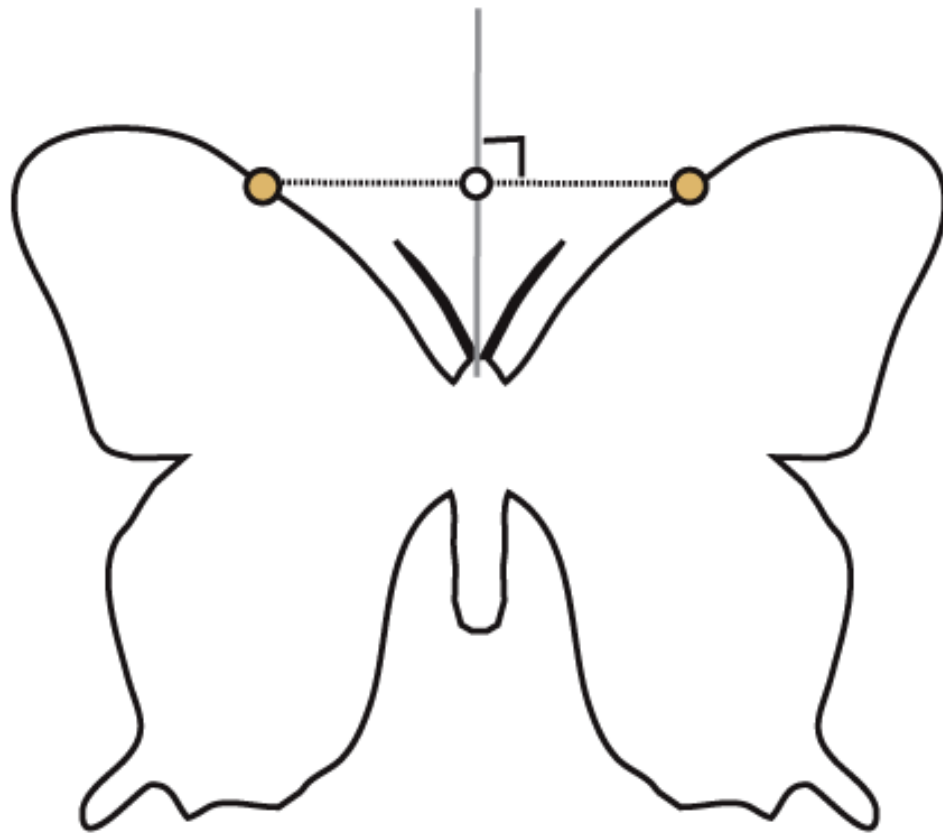
# Reflective Symmetry: A Pair Votes

---

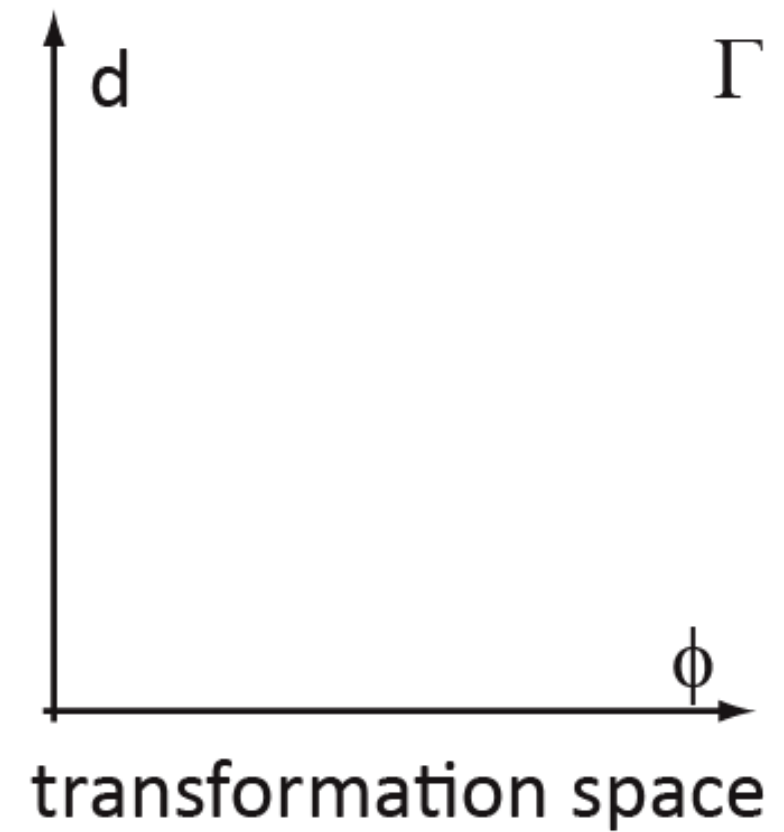
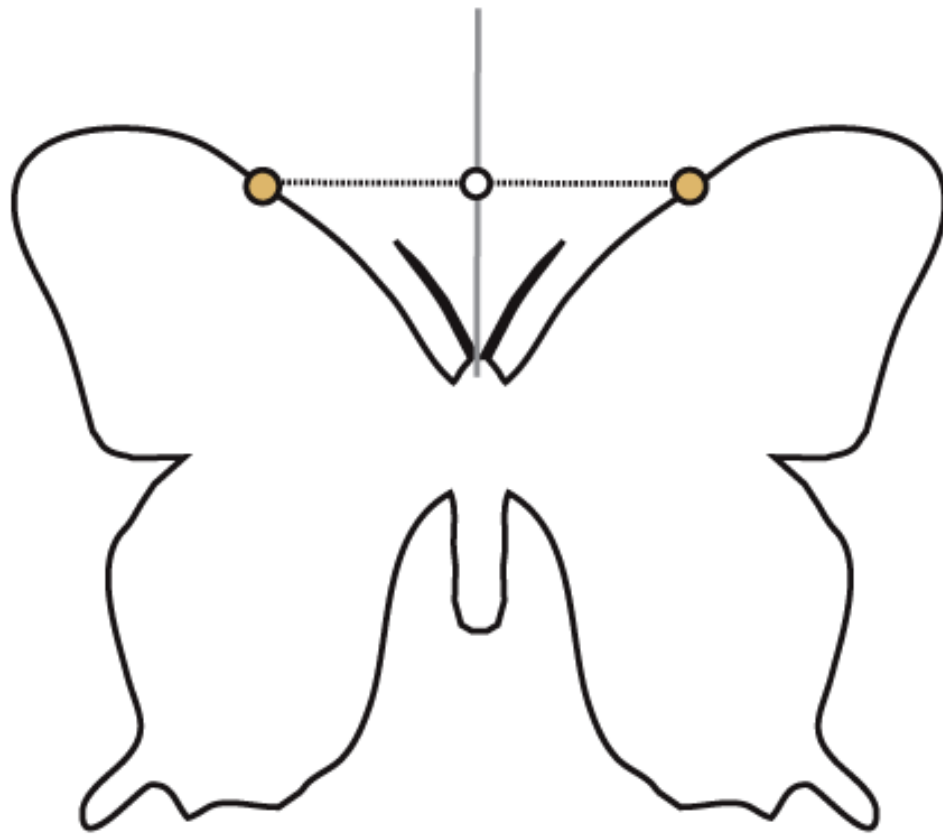


# Reflective Symmetry: A Pair Votes

---

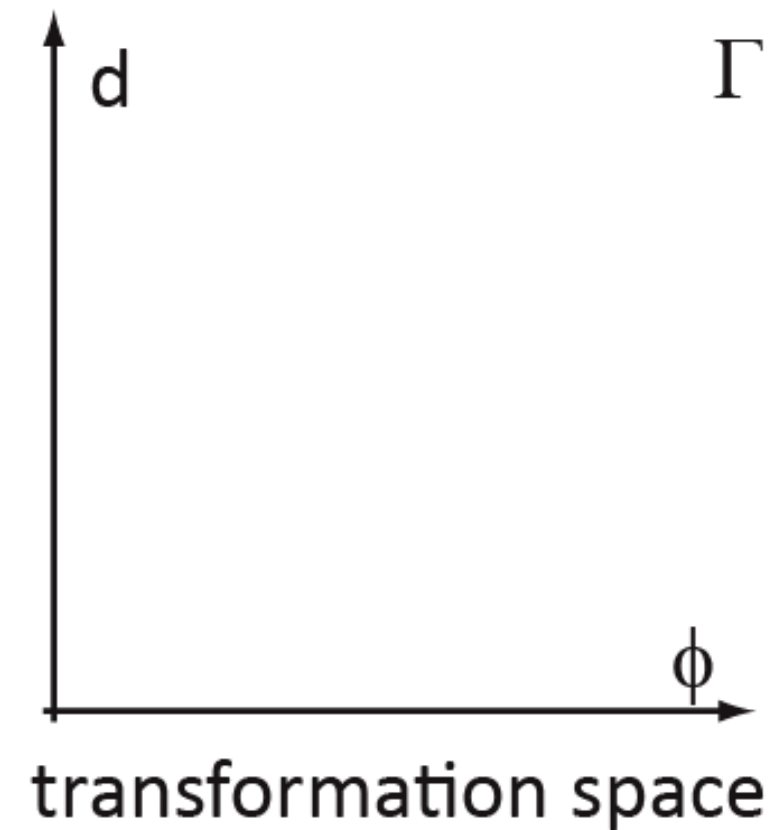
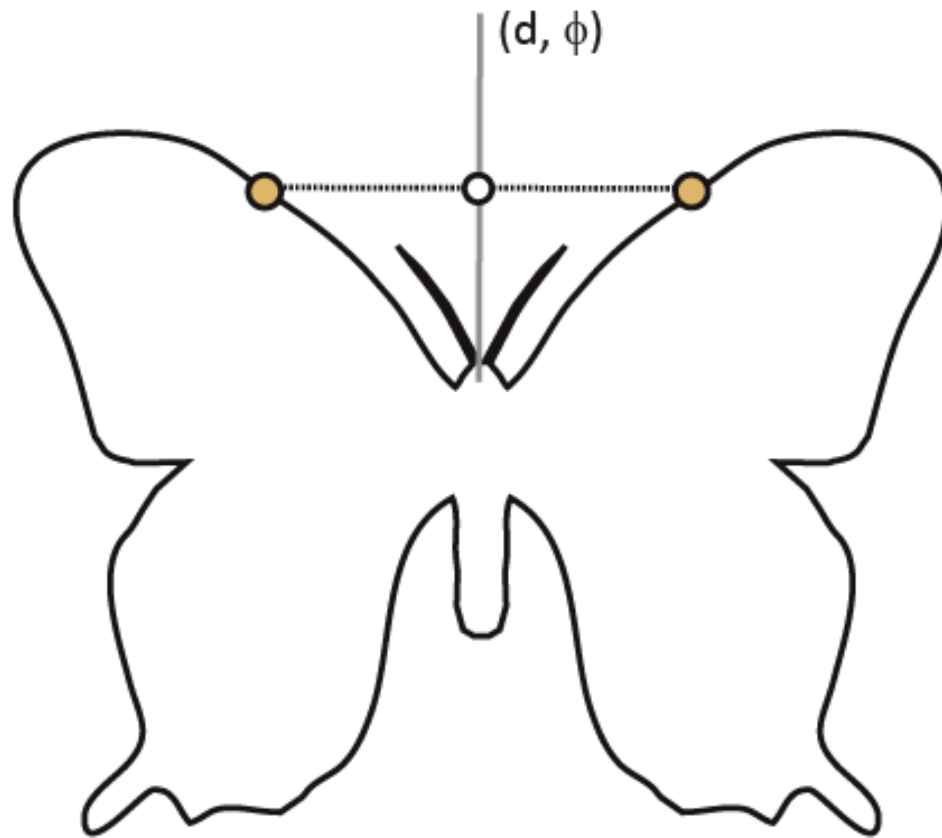


# Reflective Symmetry: A Pair Votes

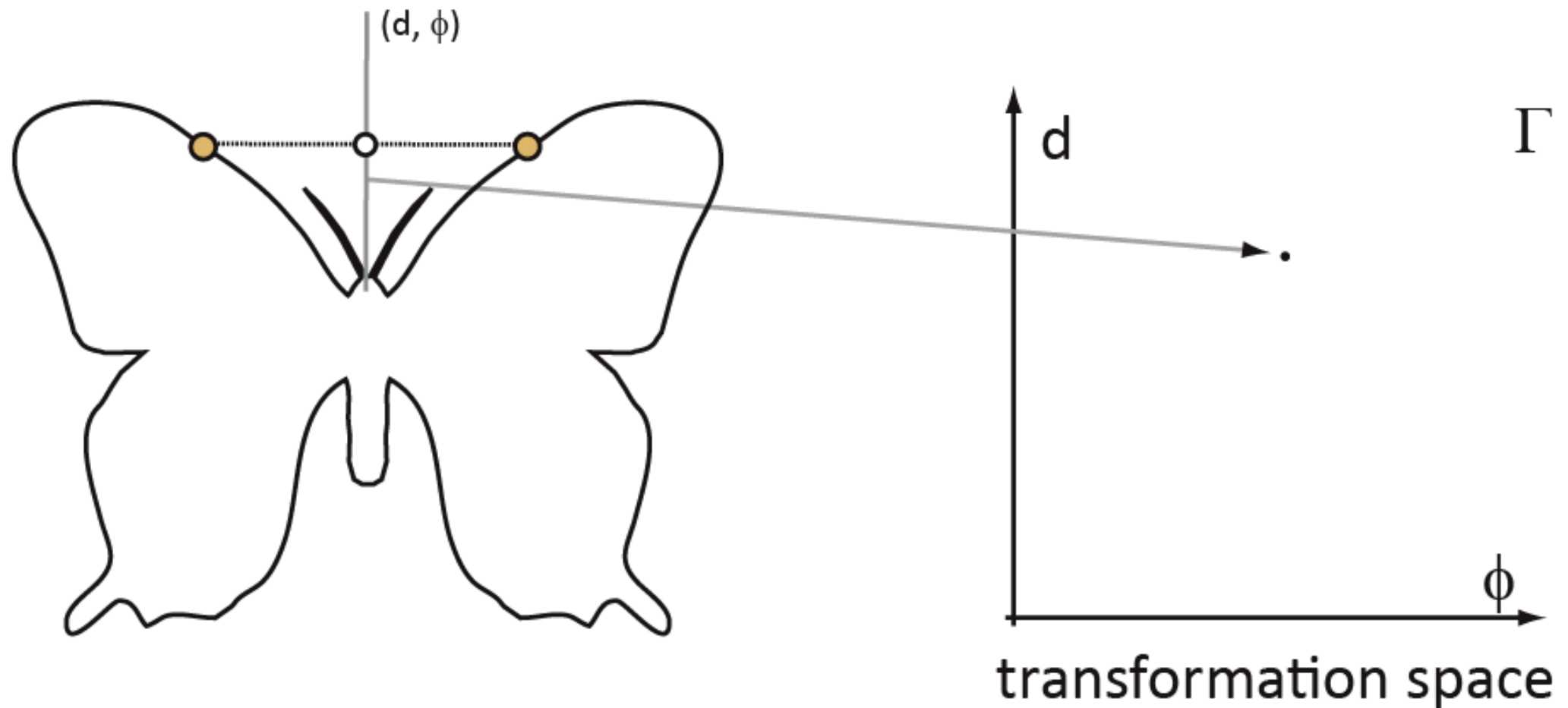




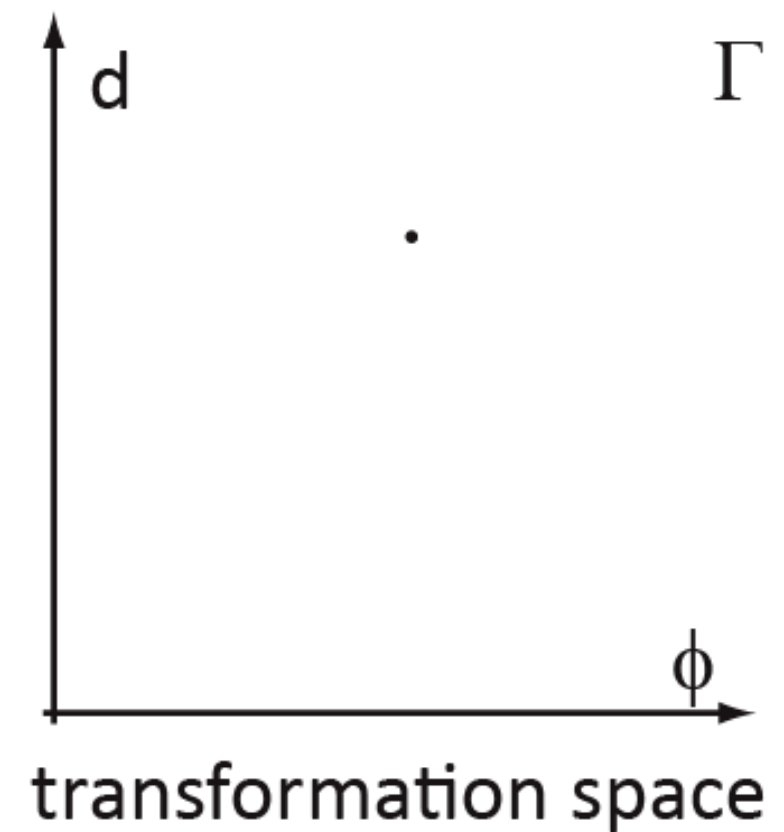
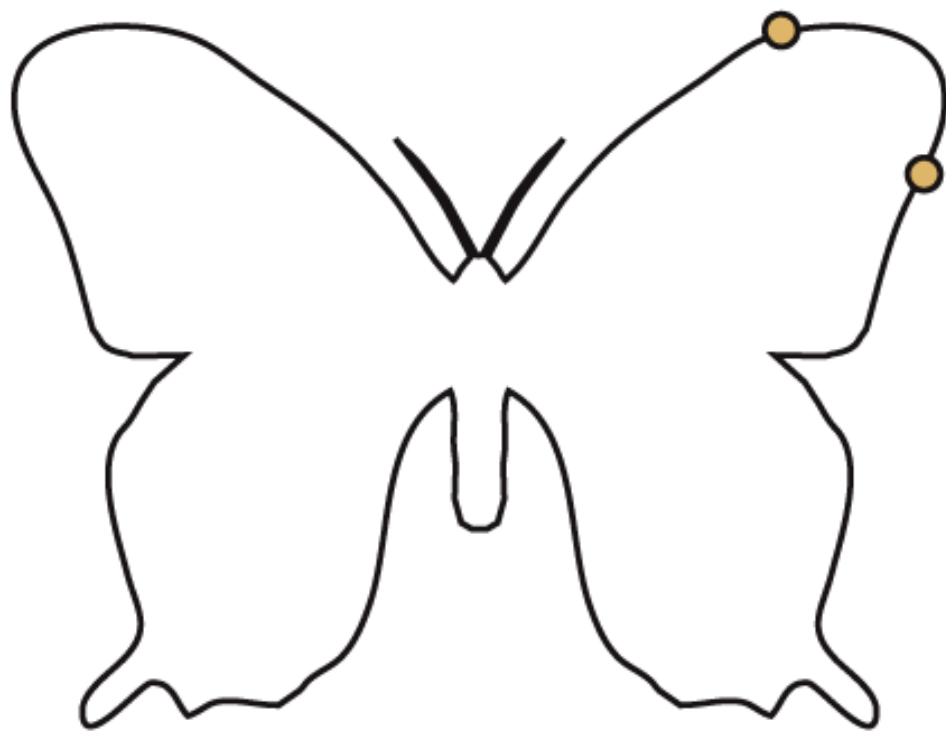
# Reflective Symmetry: A Pair Votes



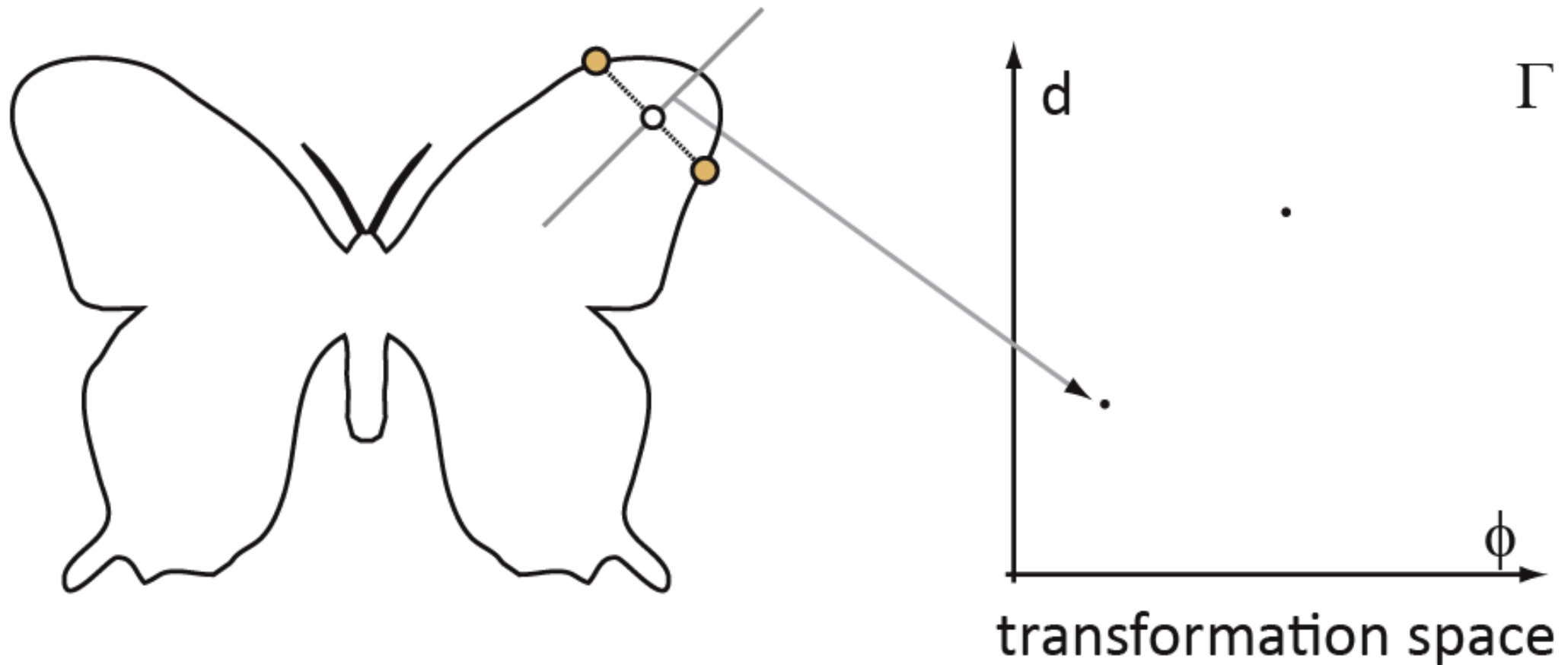
# Reflective Symmetry: A Pair Votes



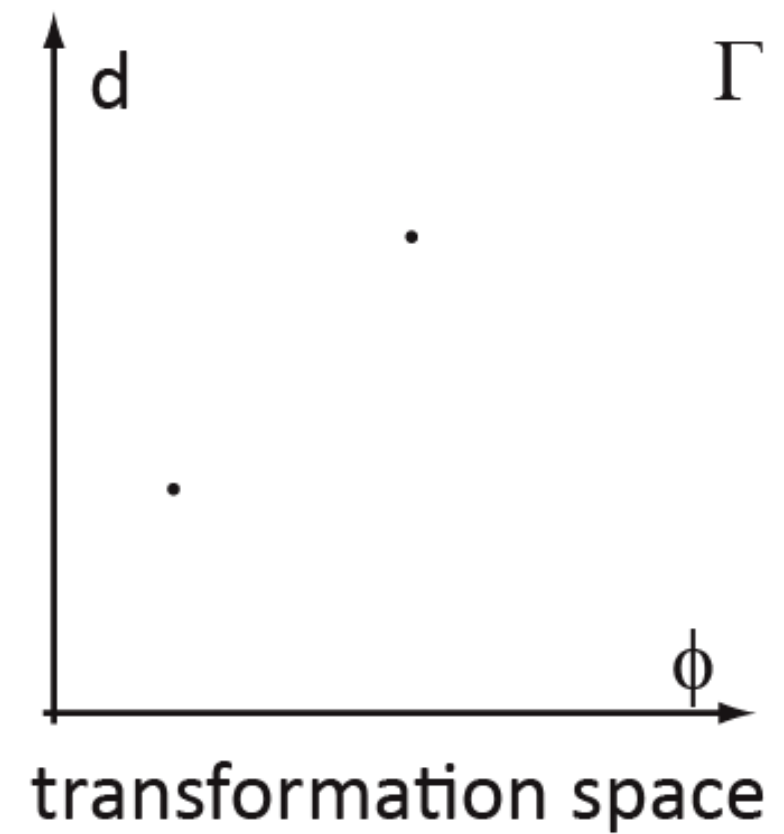
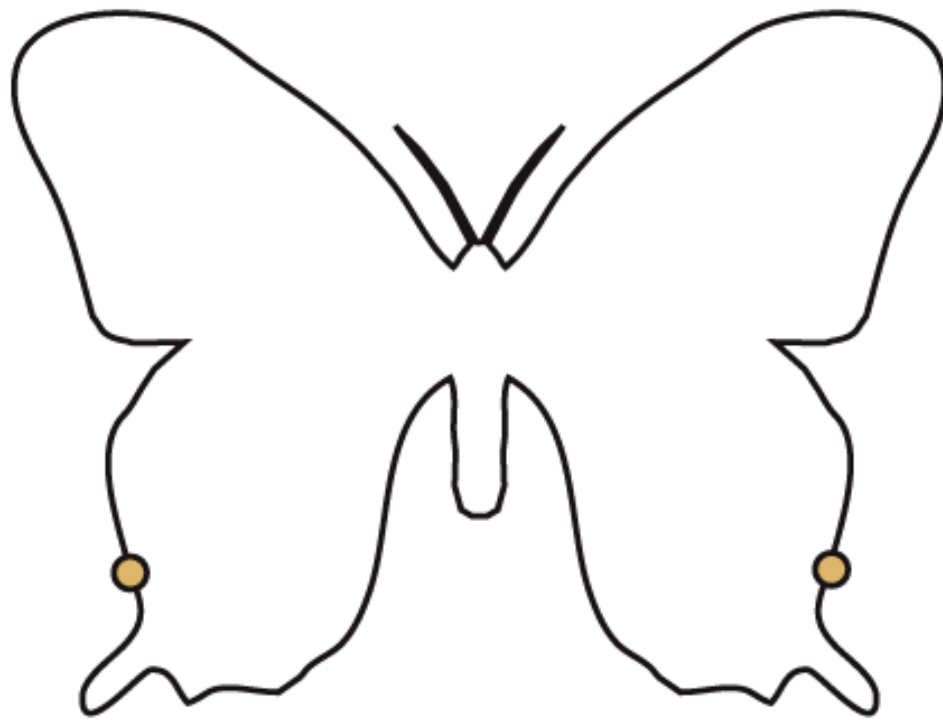
# Reflective Symmetry: Voting Continues



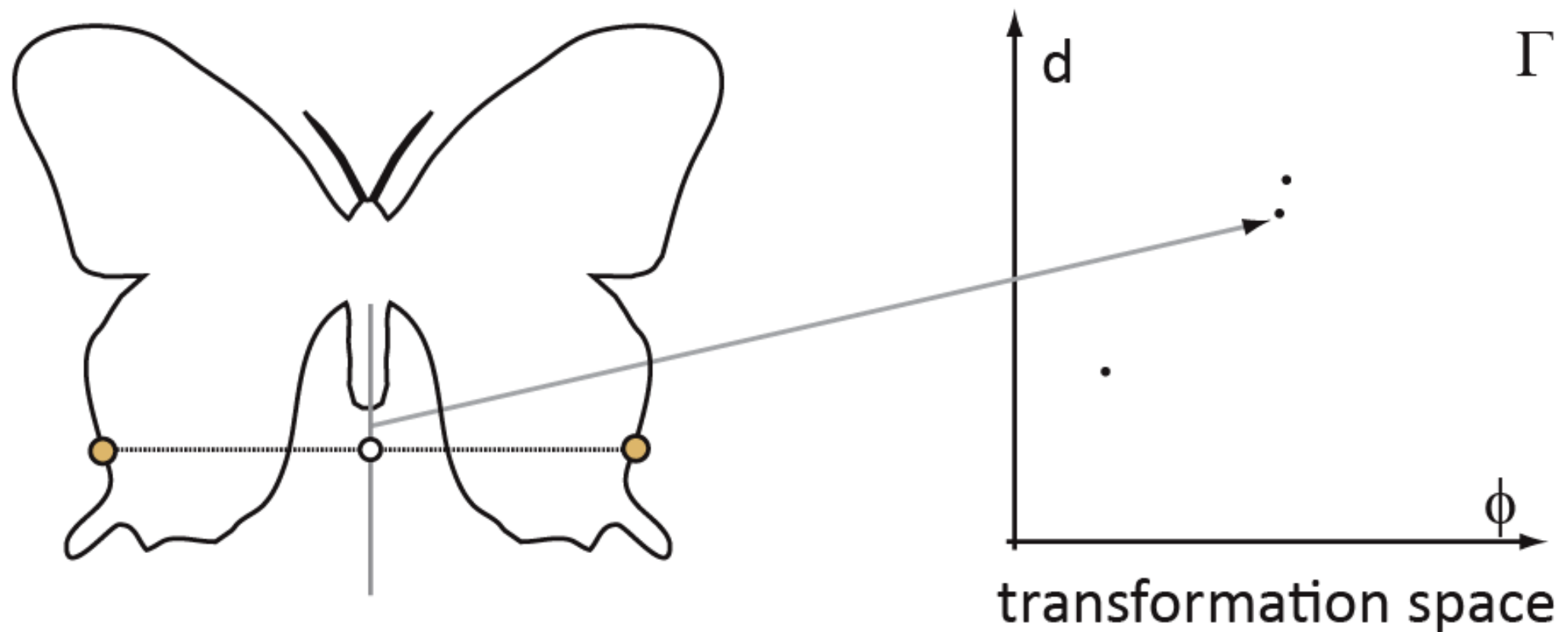
# Reflective Symmetry: Voting Continues



# Reflective Symmetry: Voting Continues

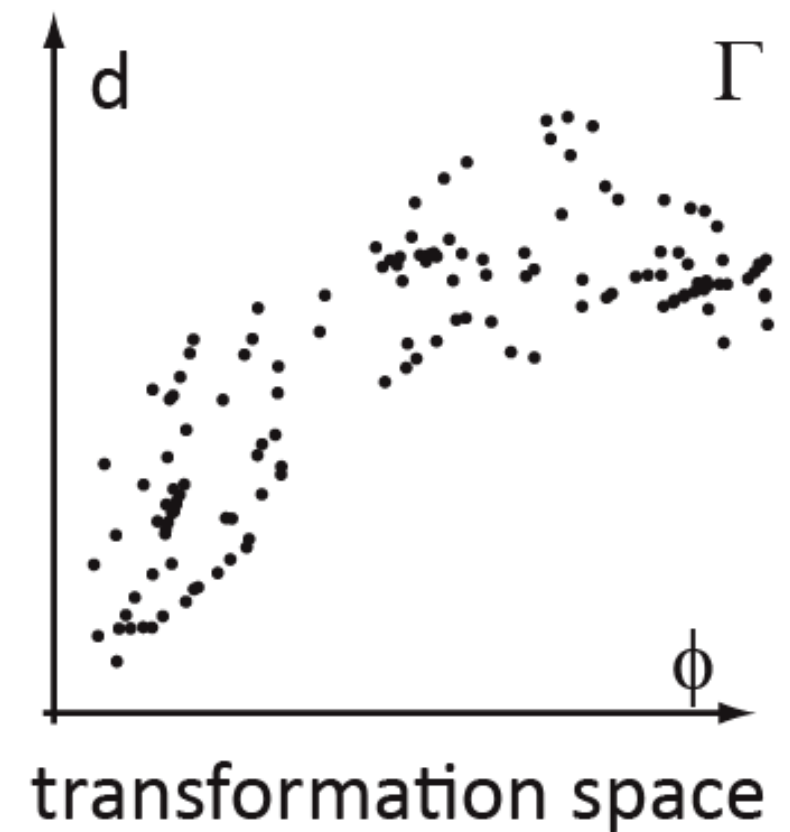
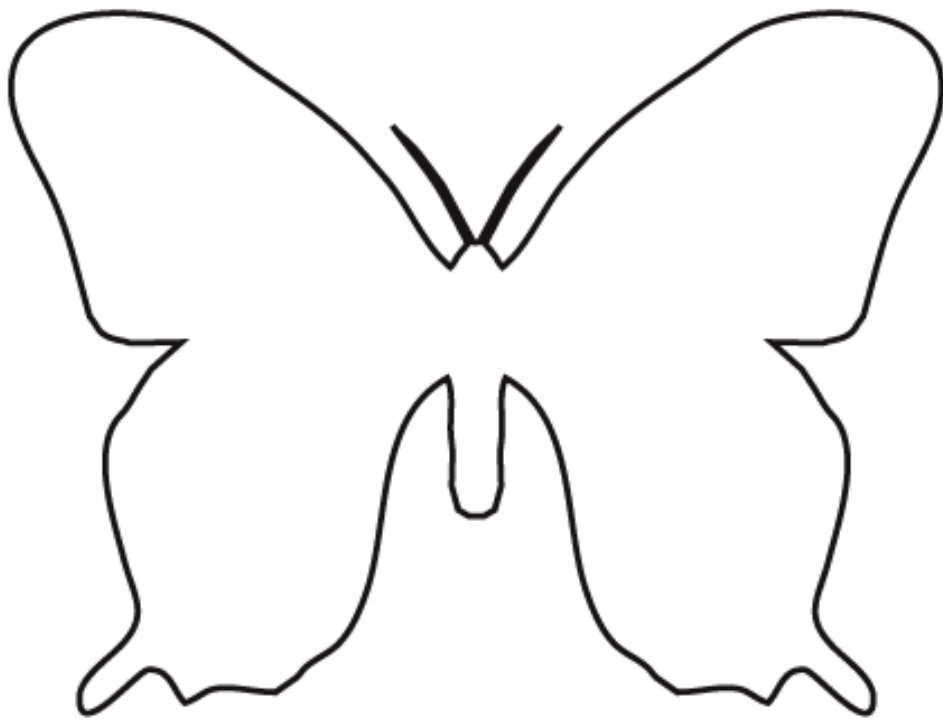


# Reflective Symmetry: Voting Continues



# Reflective Symmetry: Largest Cluster

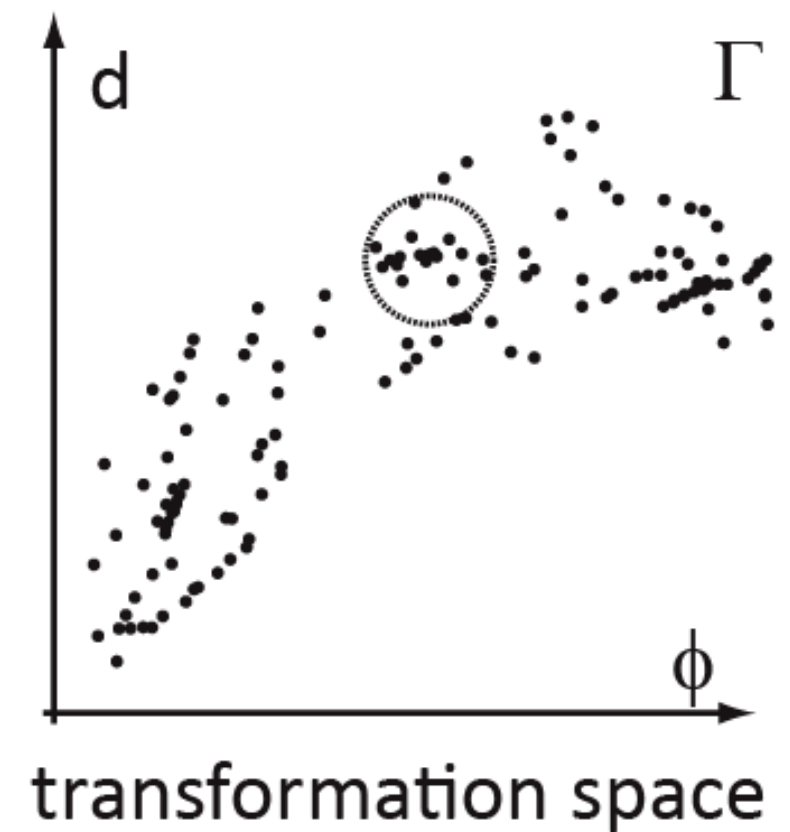
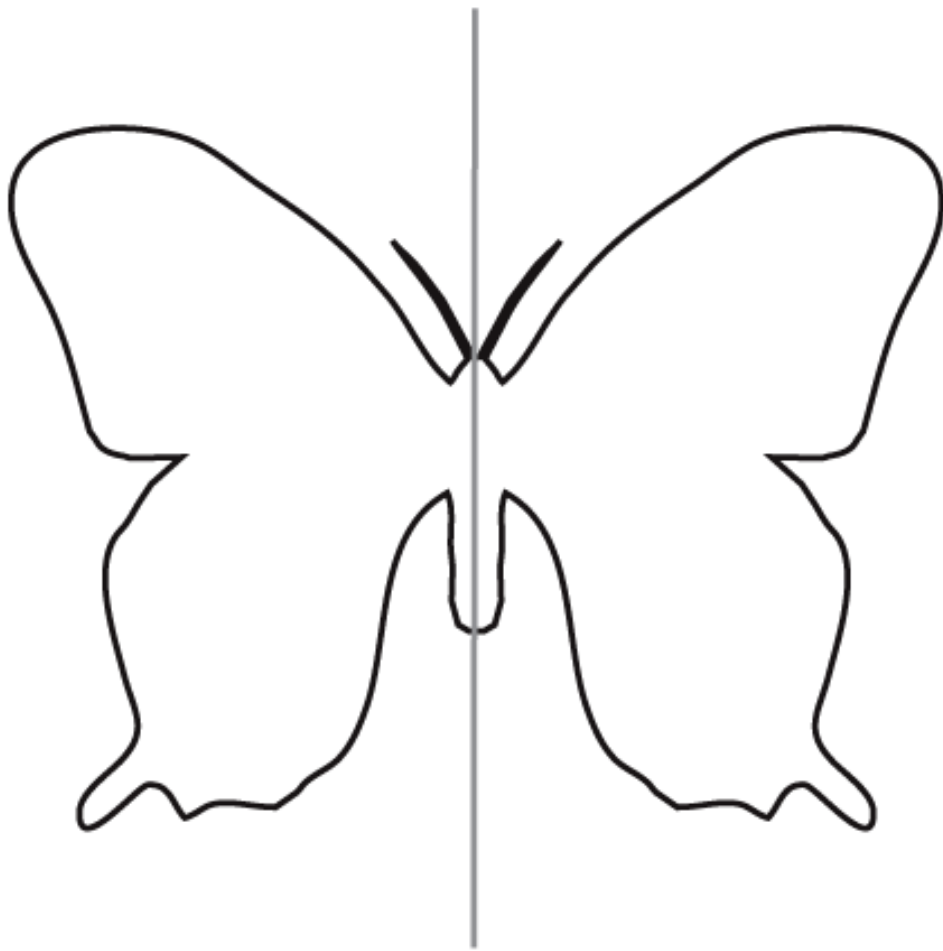
---



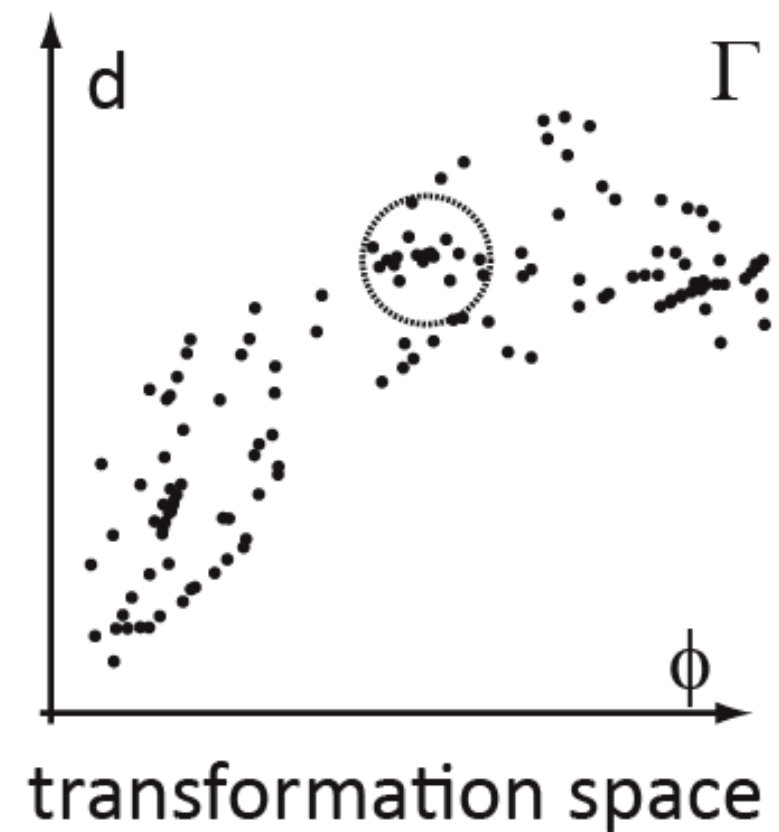
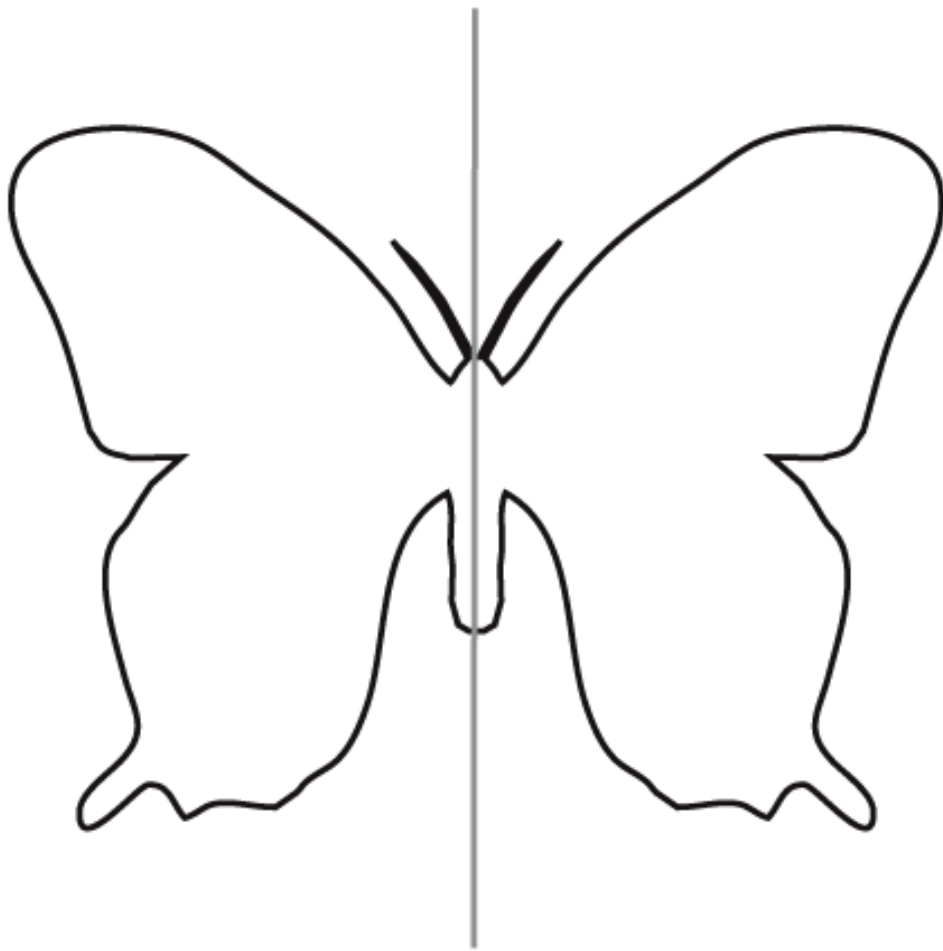


# Reflective Symmetry: Largest Cluster

---



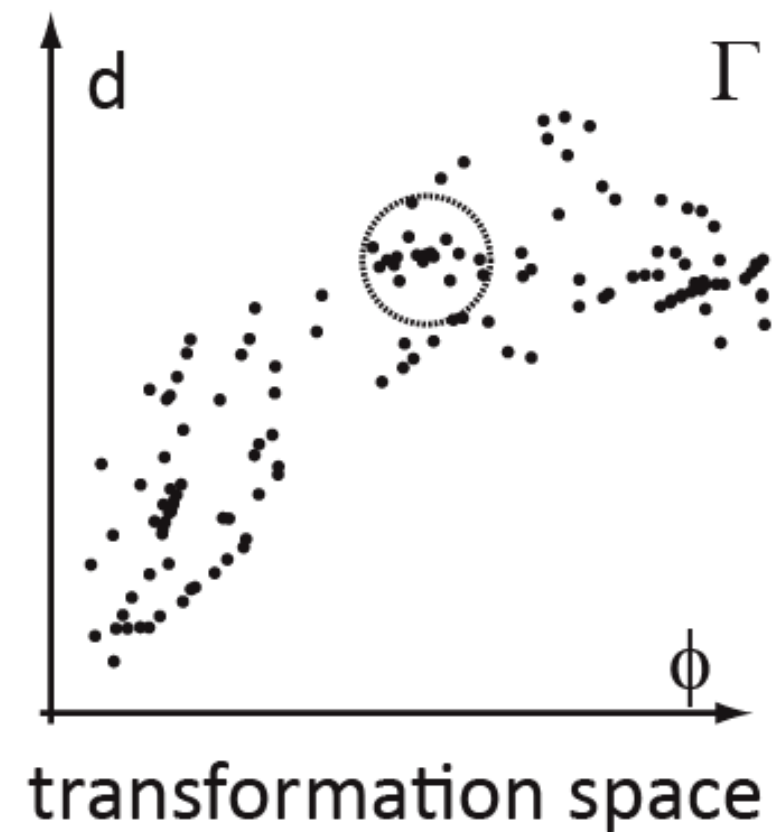
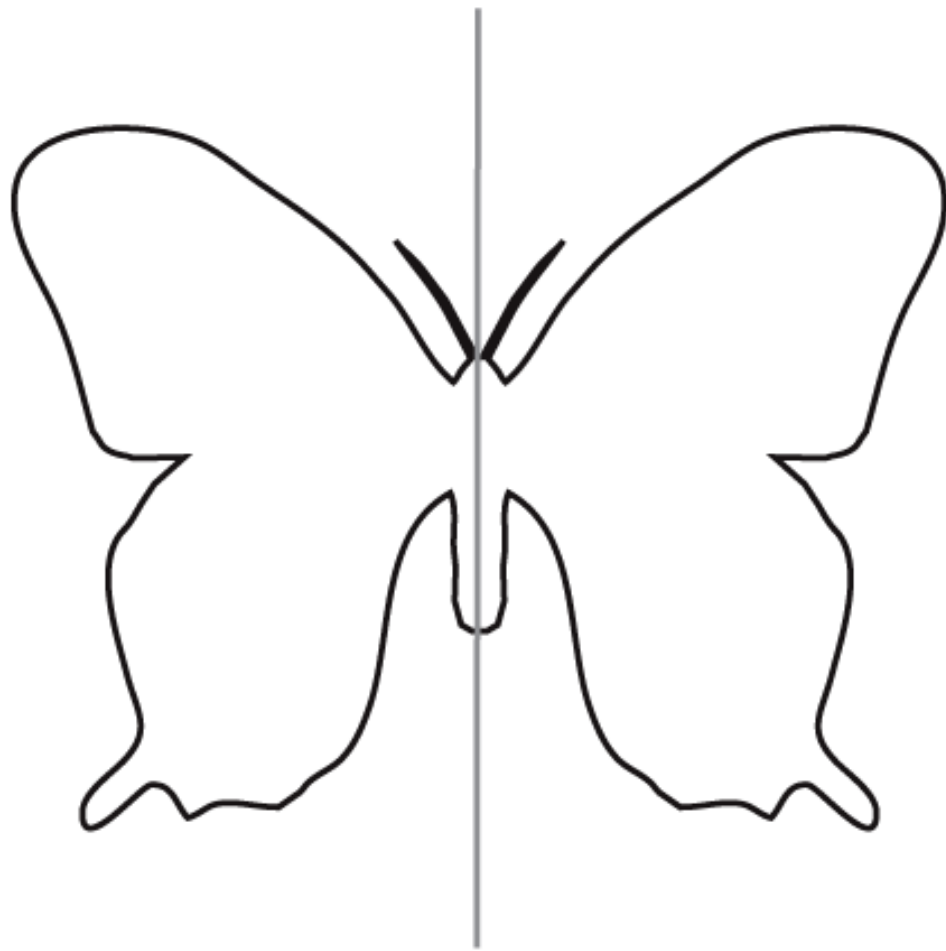
# Reflective Symmetry: Largest Cluster



- Height of cluster  $\longrightarrow$  size of patch

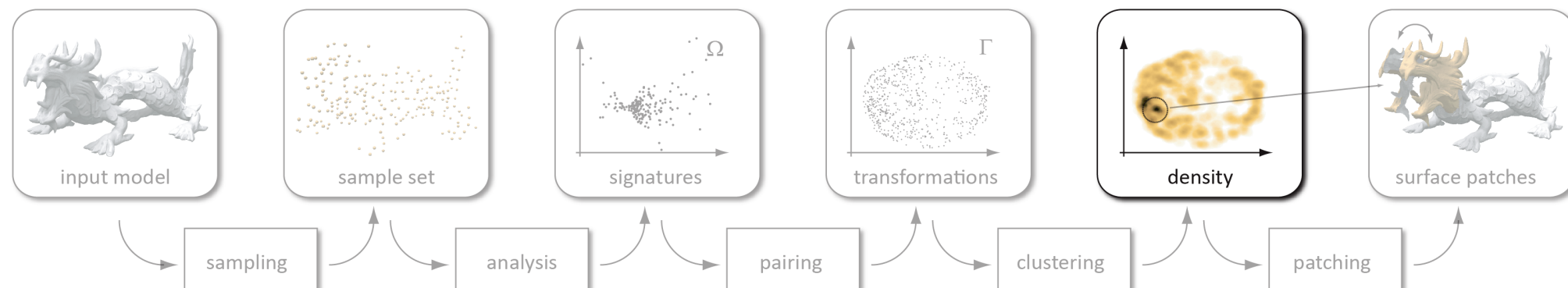
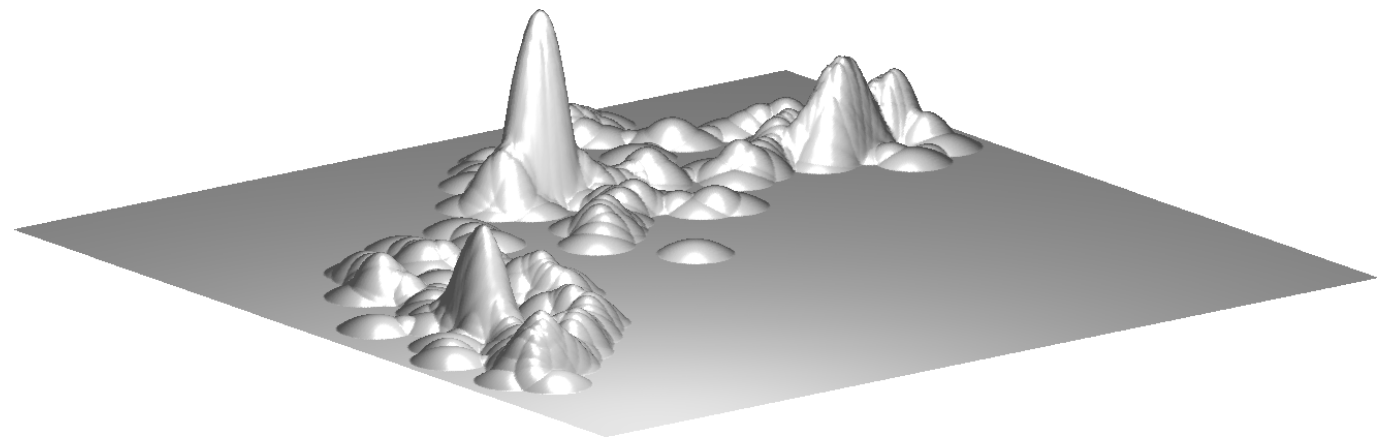
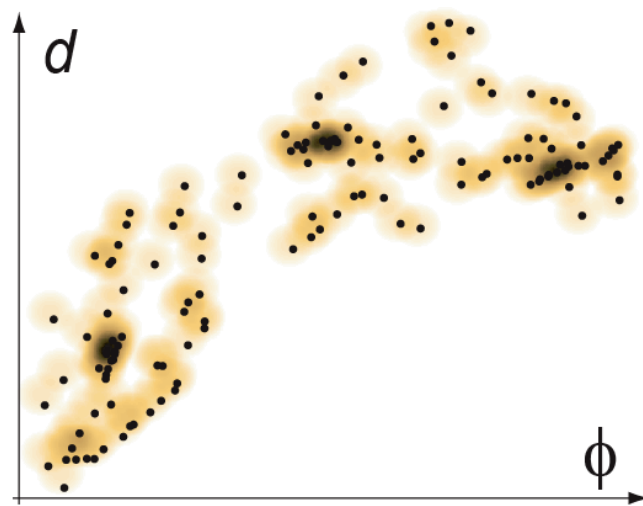
# Reflective Symmetry: Largest Cluster

---



- Height of cluster  $\longrightarrow$  size of patch
- Spread of cluster  $\longrightarrow$  level of approximation

# Mean-Shift Clustering



# Detection Results: Dragon



# Detection Results: Dragon



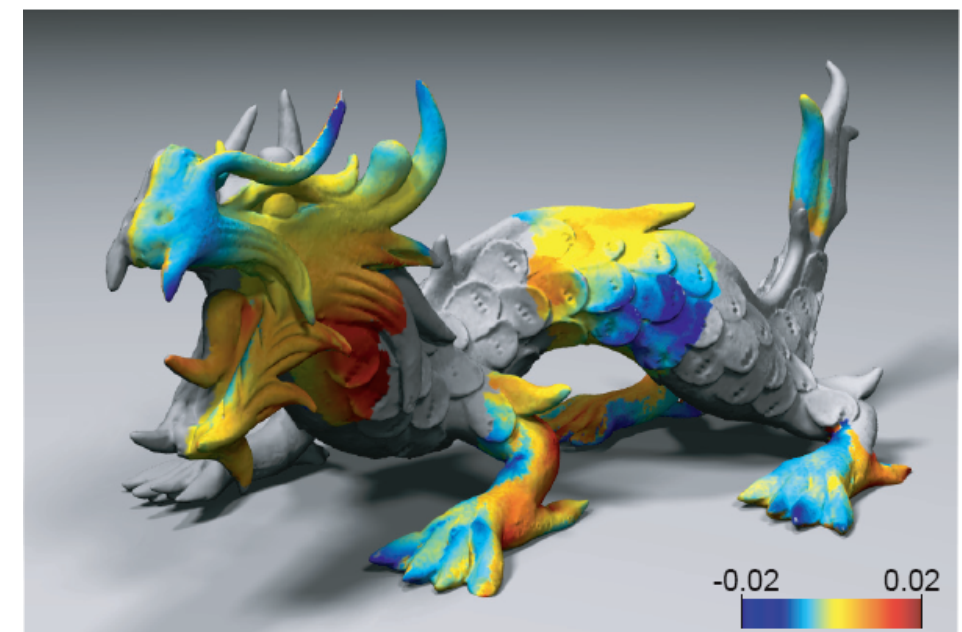
detected symmetries



# Detection Results: Dragon



detected symmetries



correction field



# Insight: Global to Local Problem

---



(Euclidean) symmetry in spatial domain



cluster(s) in transform domain

# Enhancing Symmetry



Cluster  
Contraction



Transformation Space

[Mitra et al., Siggraph 2007]

# Graph-based Symmetries

---



[Bokeloh et al., Eurographics 2009]

# Graph-based Symmetries

- ***Features:*** slippage analysis



[Bokeloh et al., Eurographics 2009]

# Graph-based Symmetries

- ***Features:*** slippage analysis



[Bokeloh et al., Eurographics 2009]

# Graph-based Symmetries

- **Features:** slippage analysis
- **Aggregation:** locally coherent line arrangements



[Bokeloh et al., Eurographics 2009]

# Graph-based Symmetries

- **Features:** slippage analysis
- **Aggregation:** locally coherent line arrangements



[Bokeloh et al., Eurographics 2009]



# Graph-based Symmetries

- **Features:** slippage analysis
- **Aggregation:** locally coherent line arrangements
- **Extraction:** simultaneous refinement



[Bokeloh et al., Eurographics 2009]

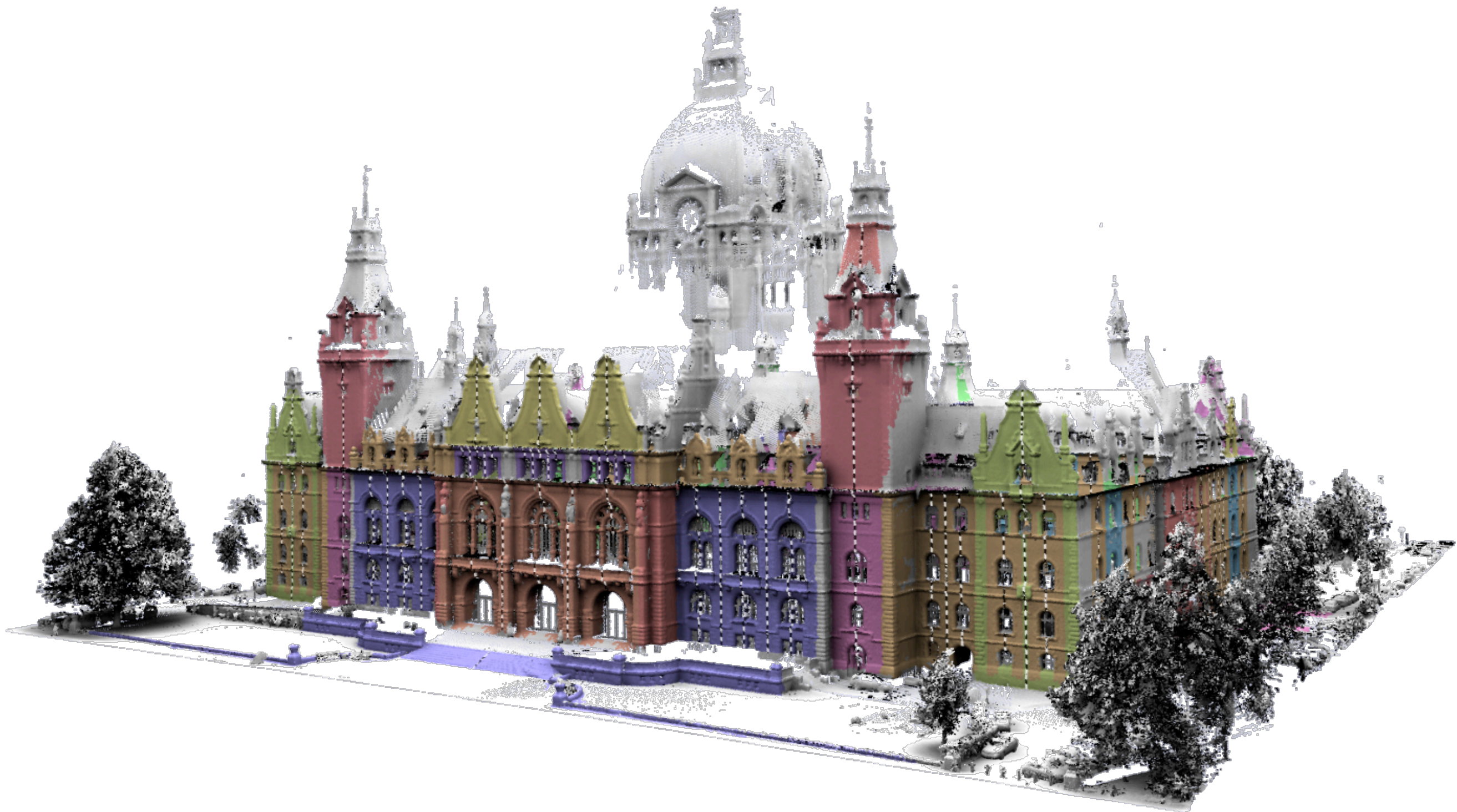
# Symmetry Detection



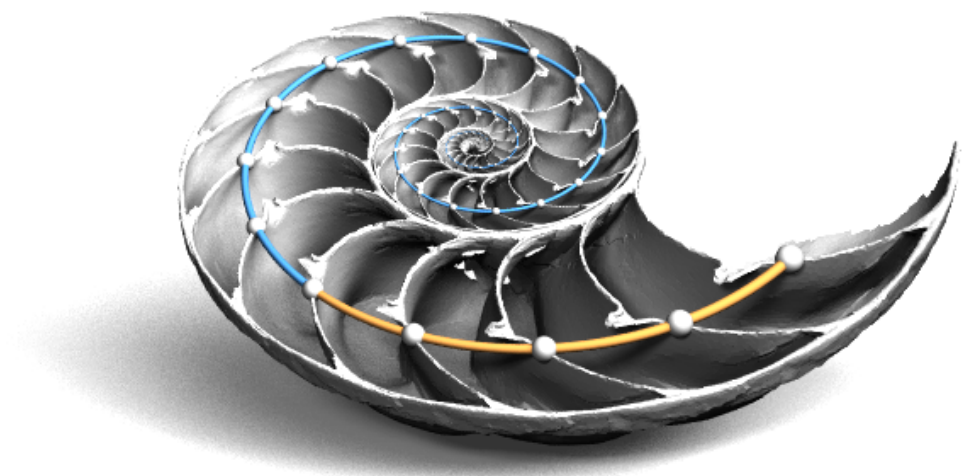
[data set: C. Brenner, IKG Univ. Hannover]



# Symmetry Detection



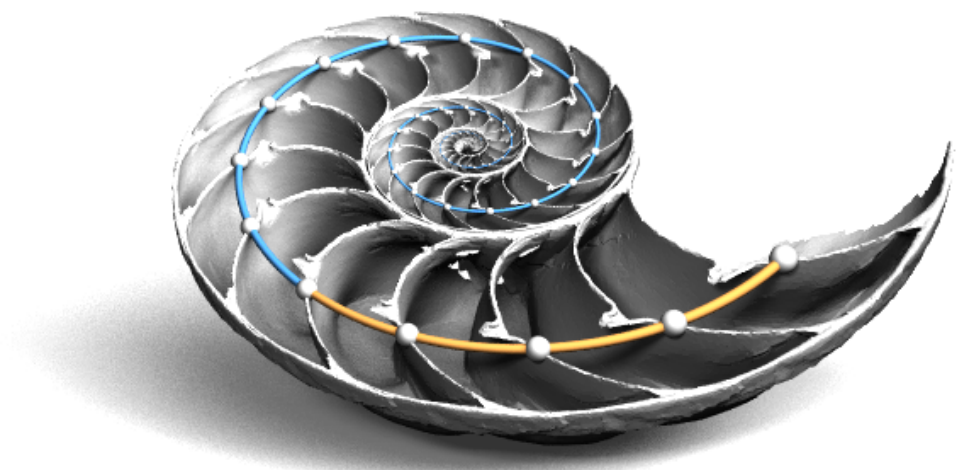
# Symmetry of Symmetries



[Pauly et al., Siggraph 2008]

# Symmetry of Symmetries

- ***Features:*** curvatures

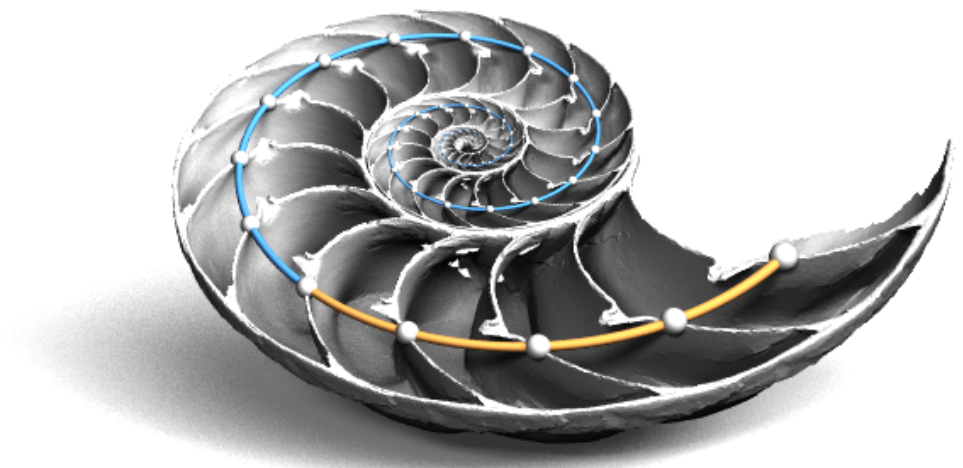


[Pauly et al., Siggraph 2008]



# Symmetry of Symmetries

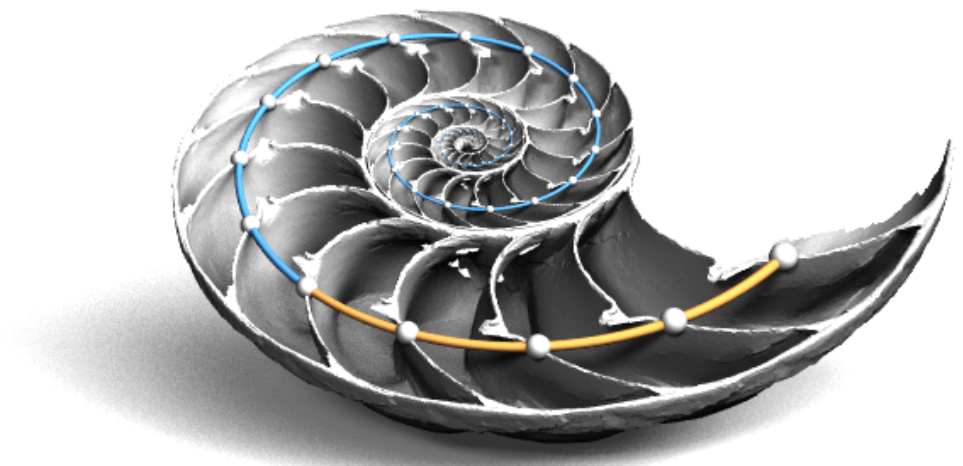
- ***Features:*** curvatures



[Pauly et al., Siggraph 2008]

# Symmetry of Symmetries

- **Features:** curvatures
- **Aggregation:** transform domain model extraction

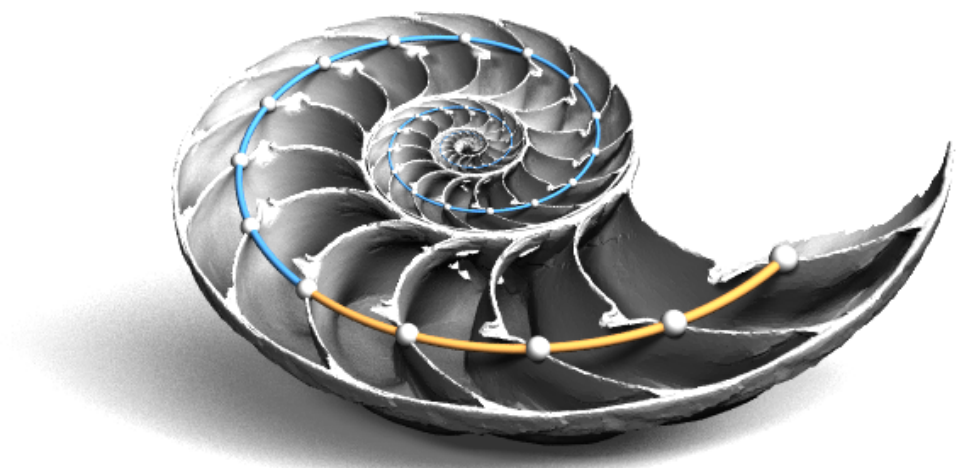


[Pauly et al., Siggraph 2008]



# Symmetry of Symmetries

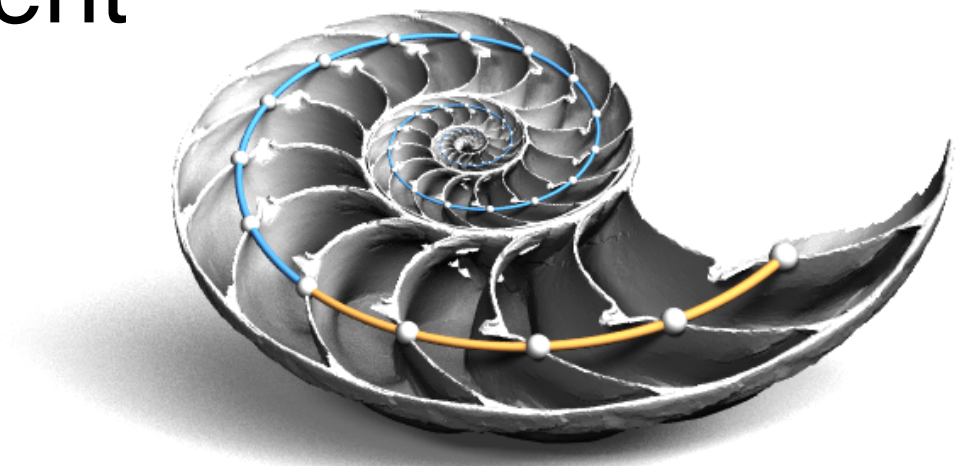
- **Features:** curvatures
- **Aggregation:** transform domain model extraction



[Pauly et al., Siggraph 2008]

# Symmetry of Symmetries

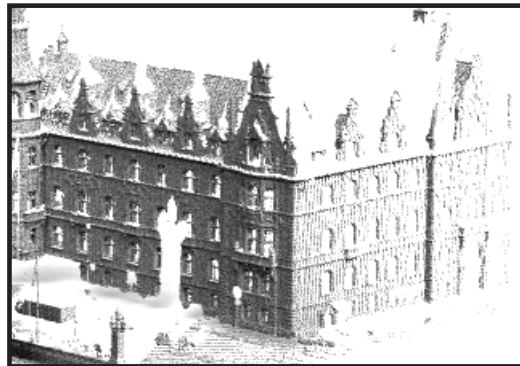
- **Features:** curvatures
- **Aggregation:** transform domain model extraction
- **Extraction:** simultaneous refinement



[Pauly et al., Siggraph 2008]

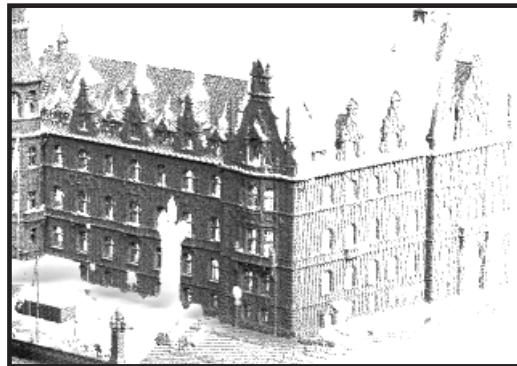
# Structure Discovery

---



Input Model

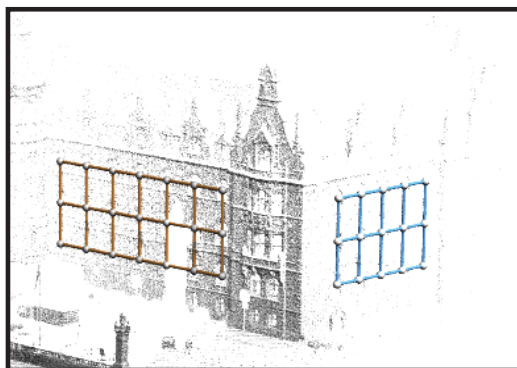
# Structure Discovery



Input Model

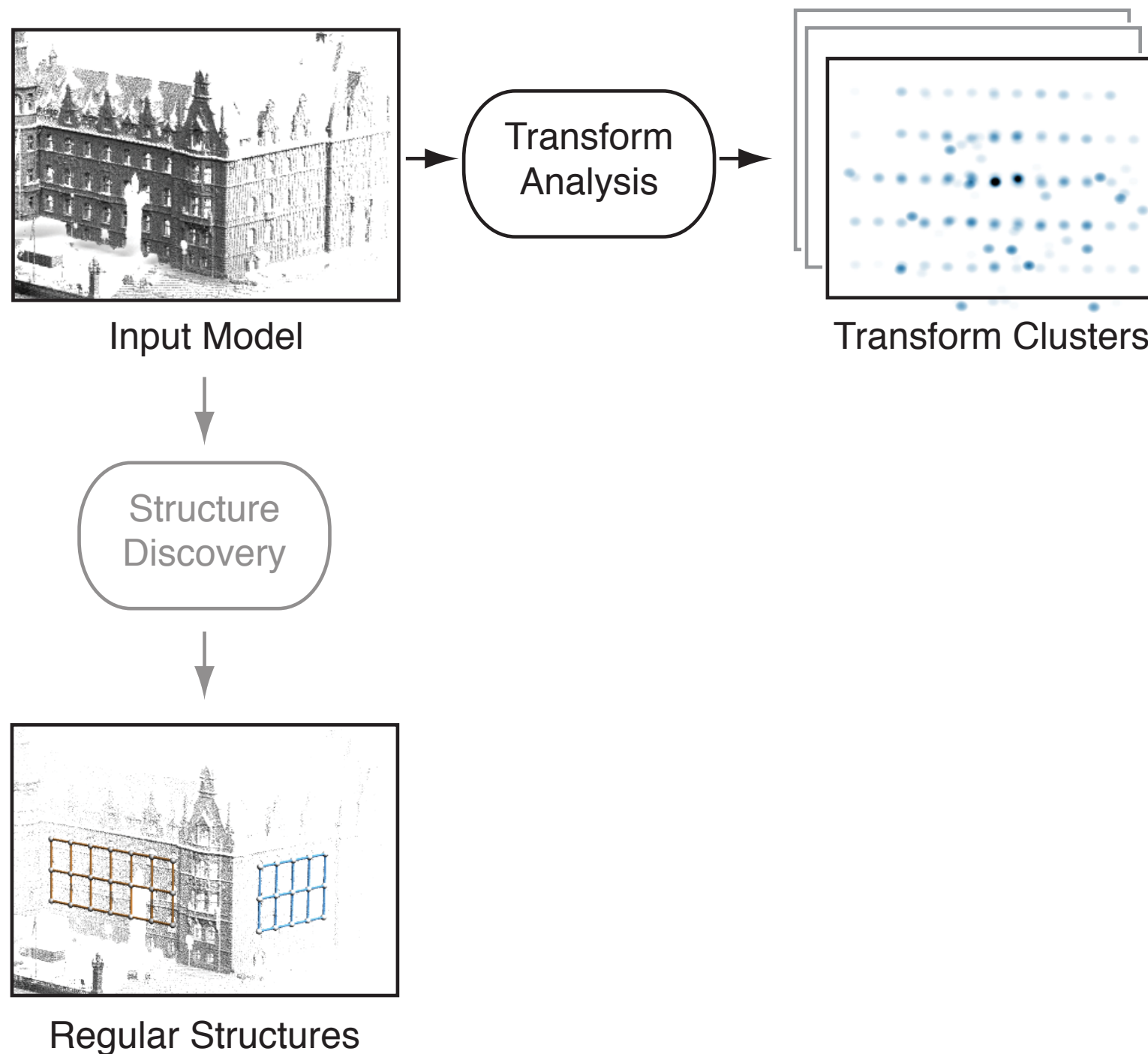


Structure  
Discovery

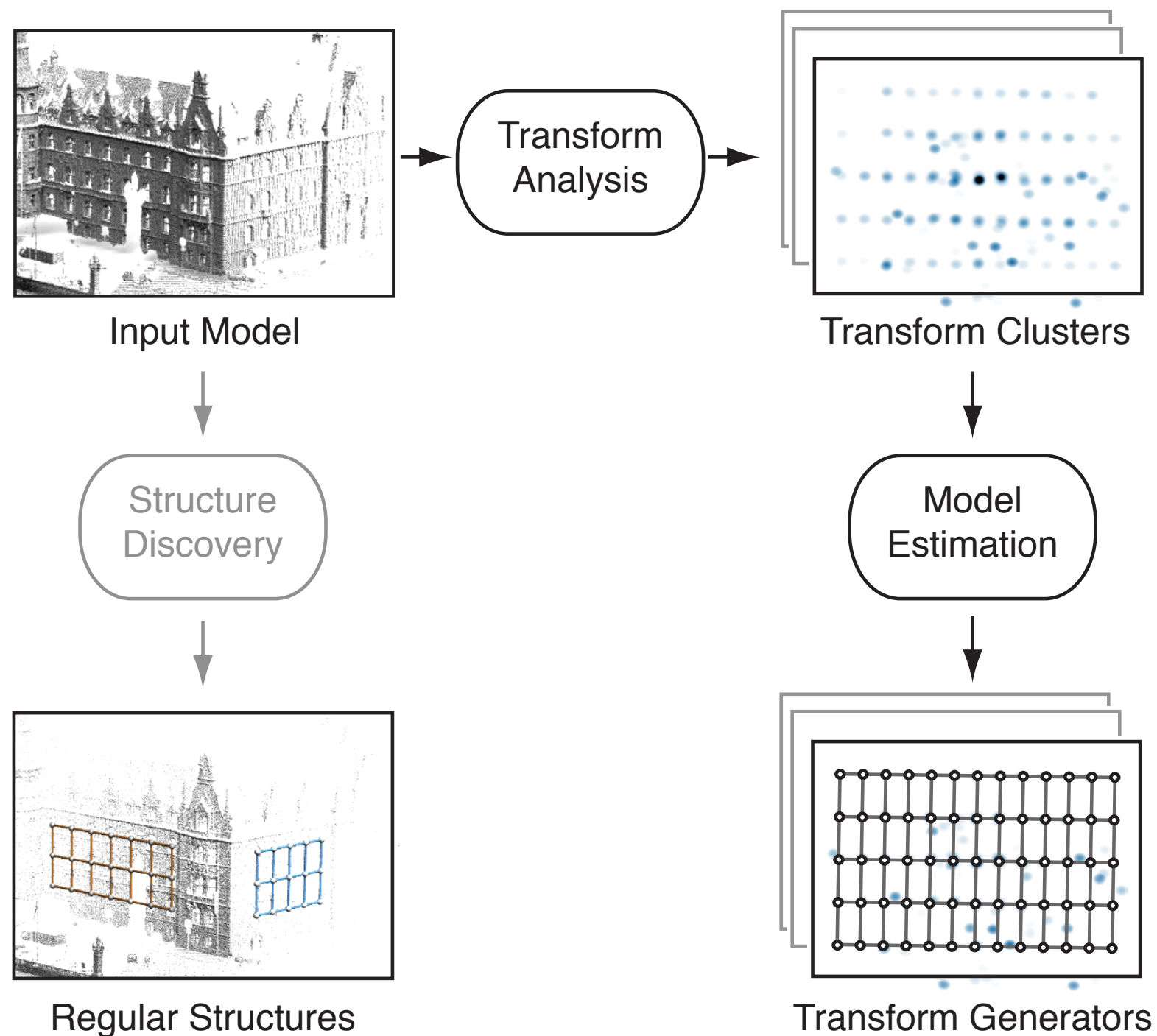


Regular Structures

# Structure Discovery

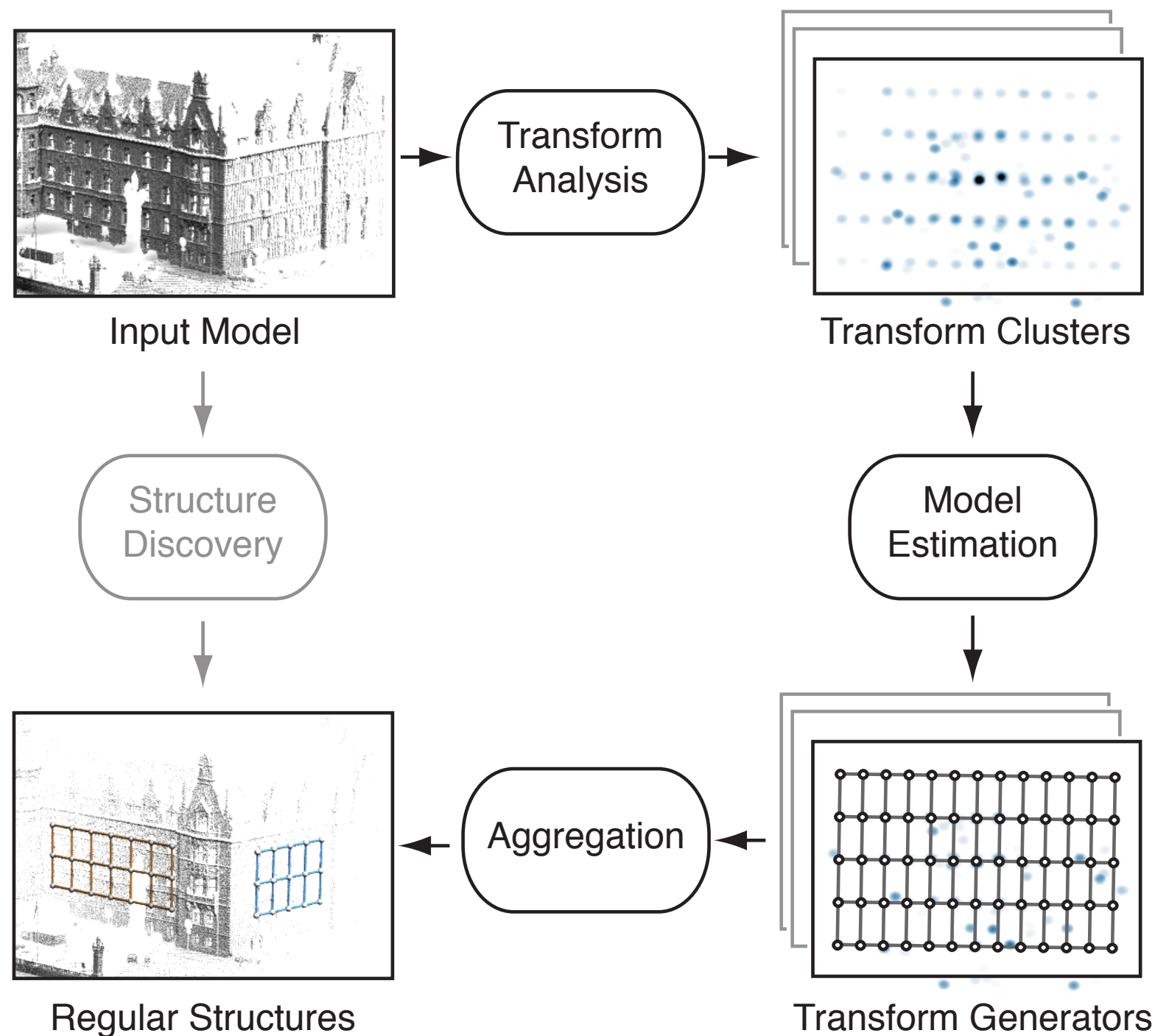


# Structure Discovery



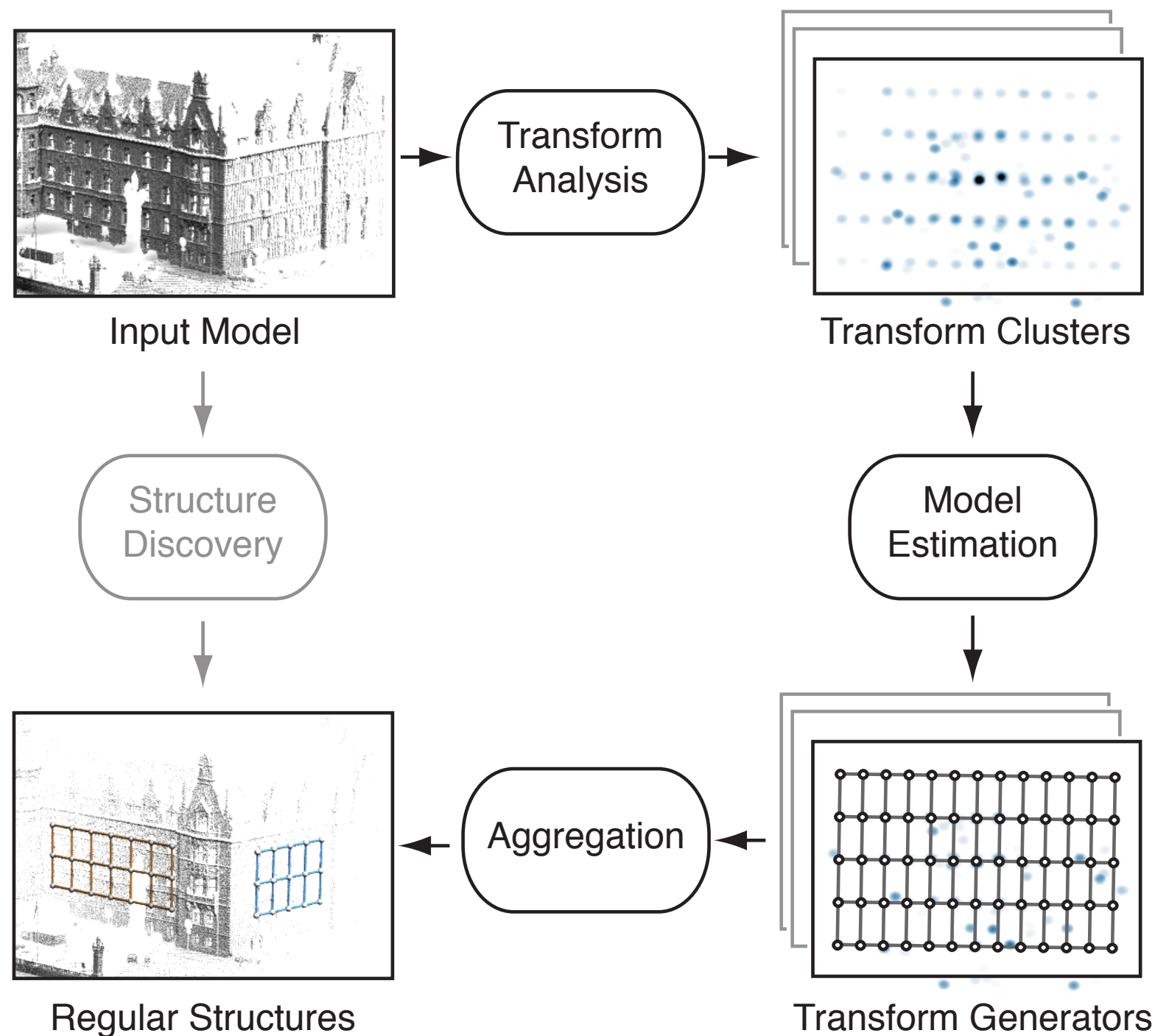


# Structure Discovery

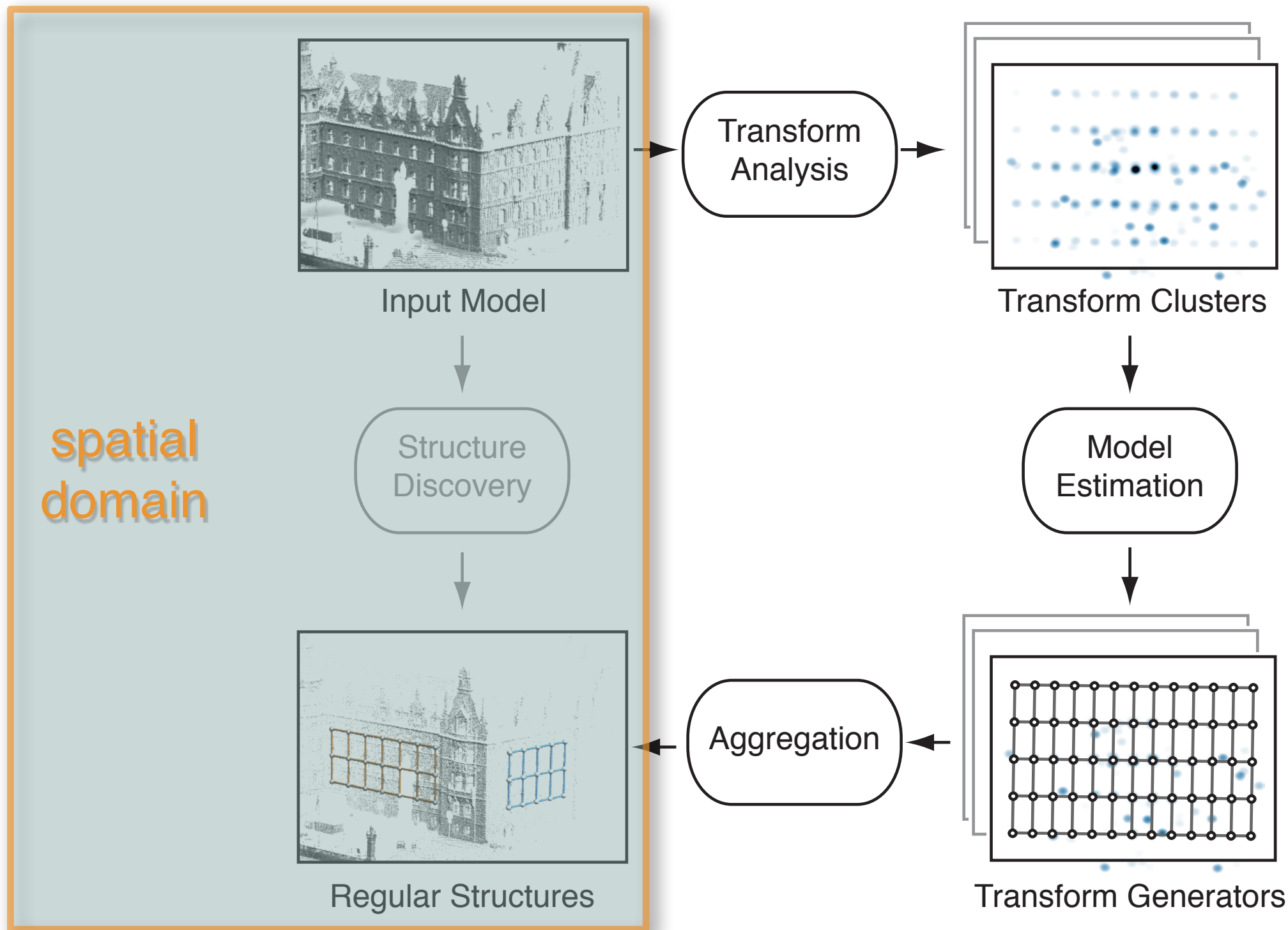




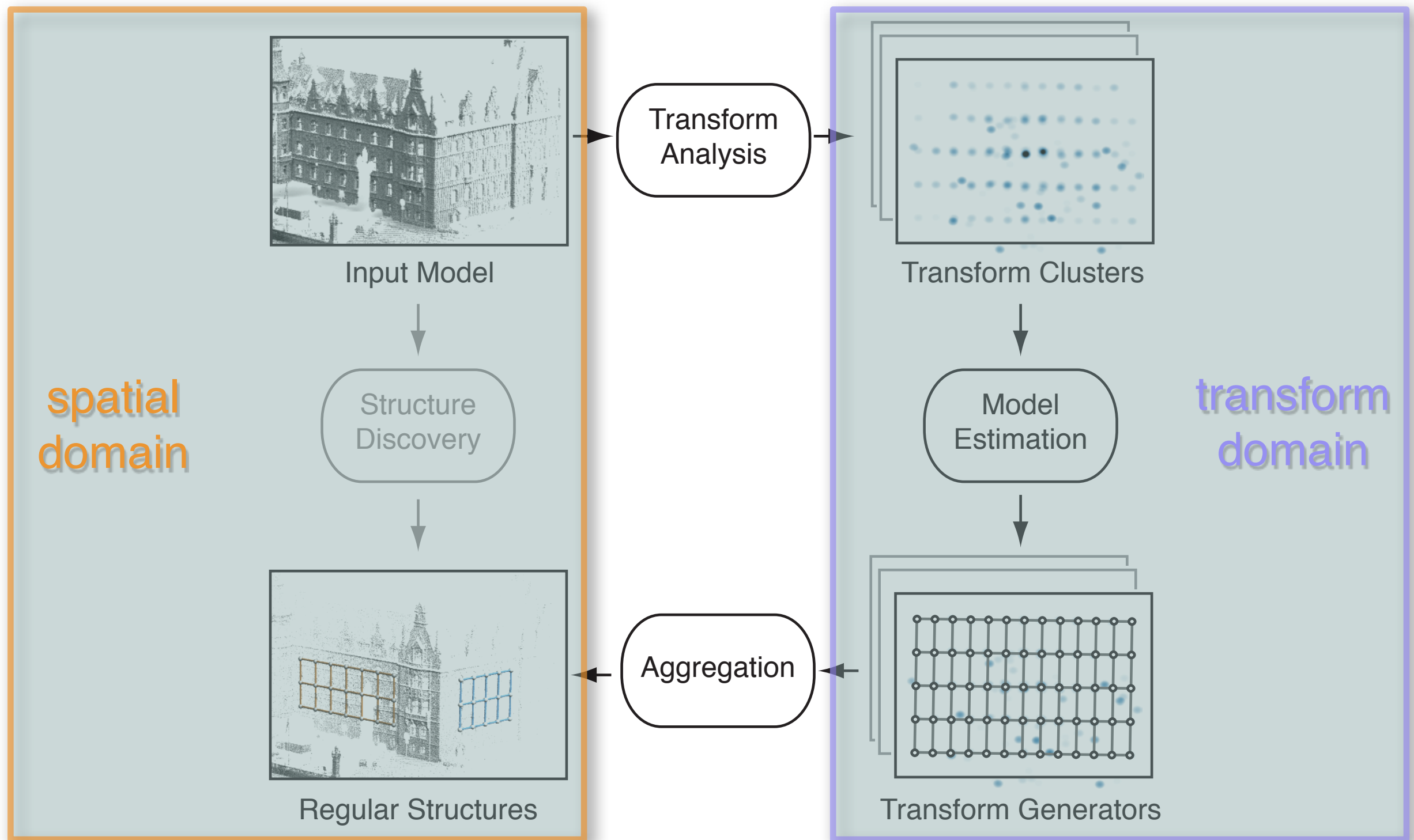
# Structure Discovery



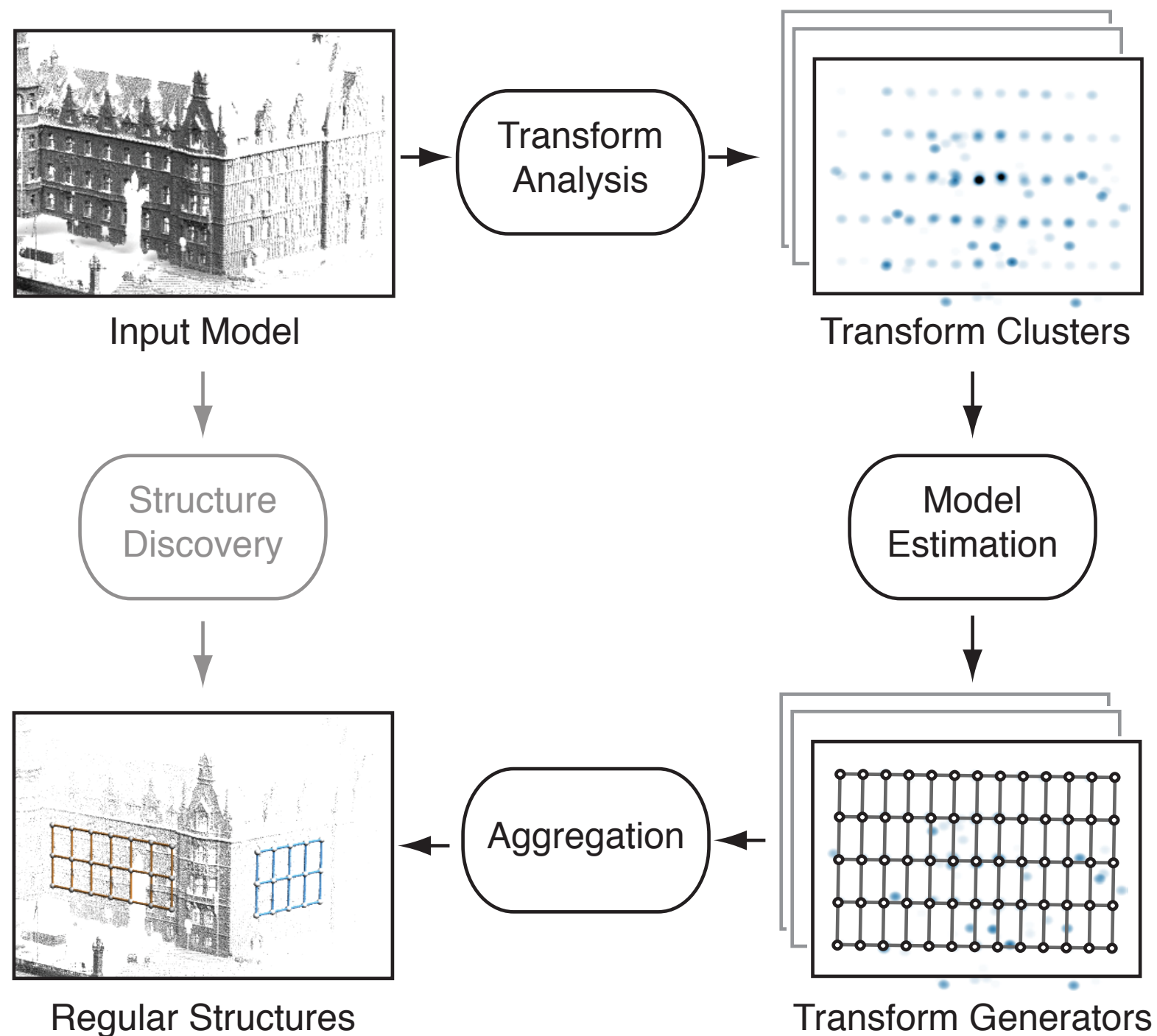
# Structure Discovery



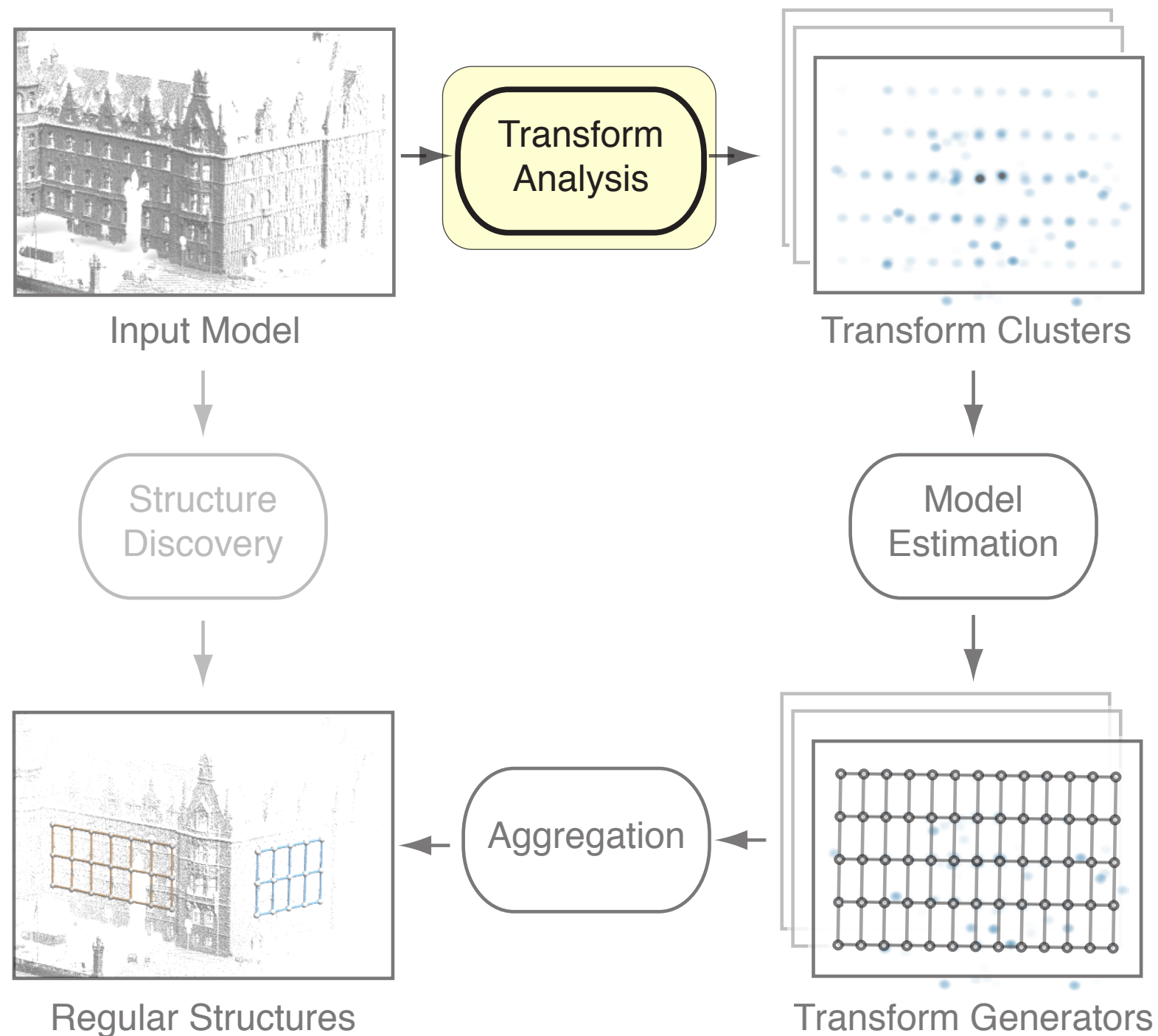
# Structure Discovery



# Structure Discovery



# Structure Discovery



# Transformations

---



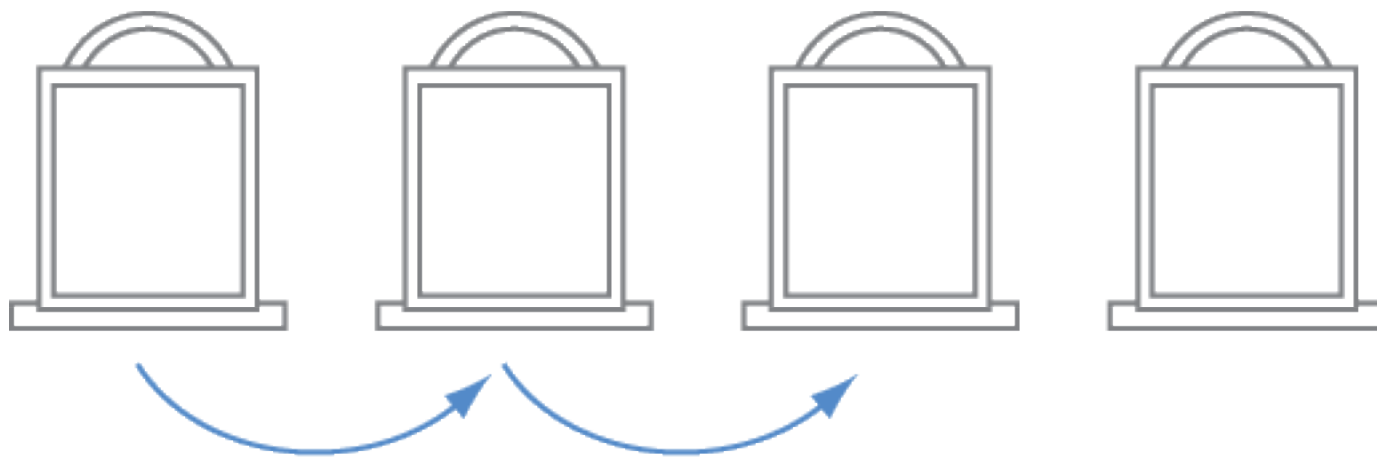


# Transformations

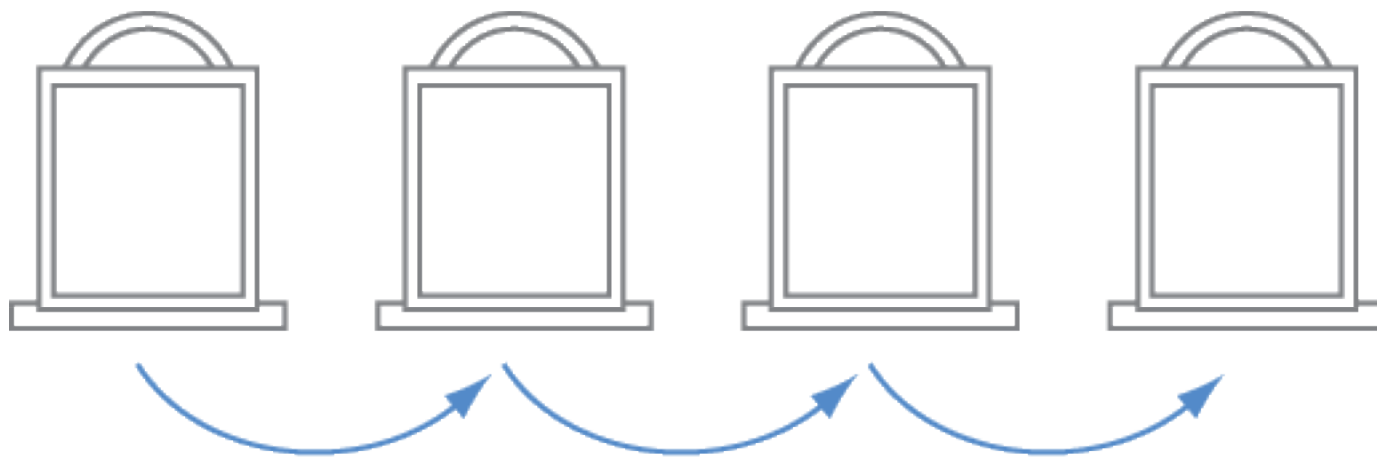




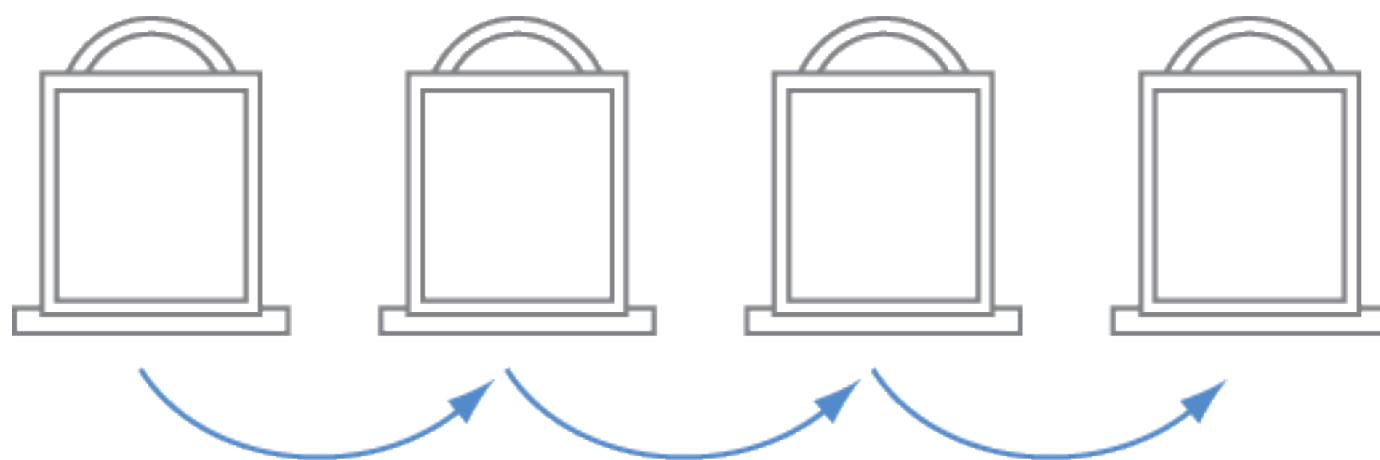
# Transformations



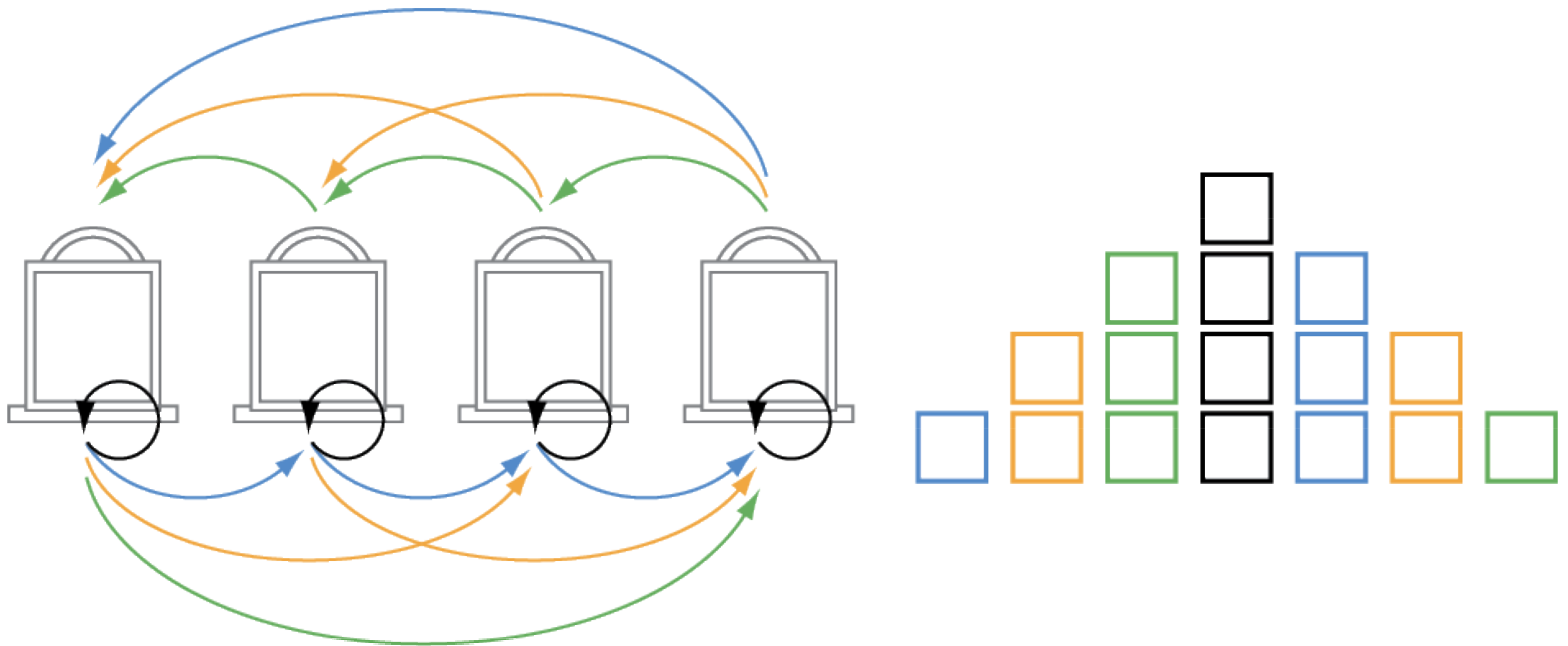
# Transformations



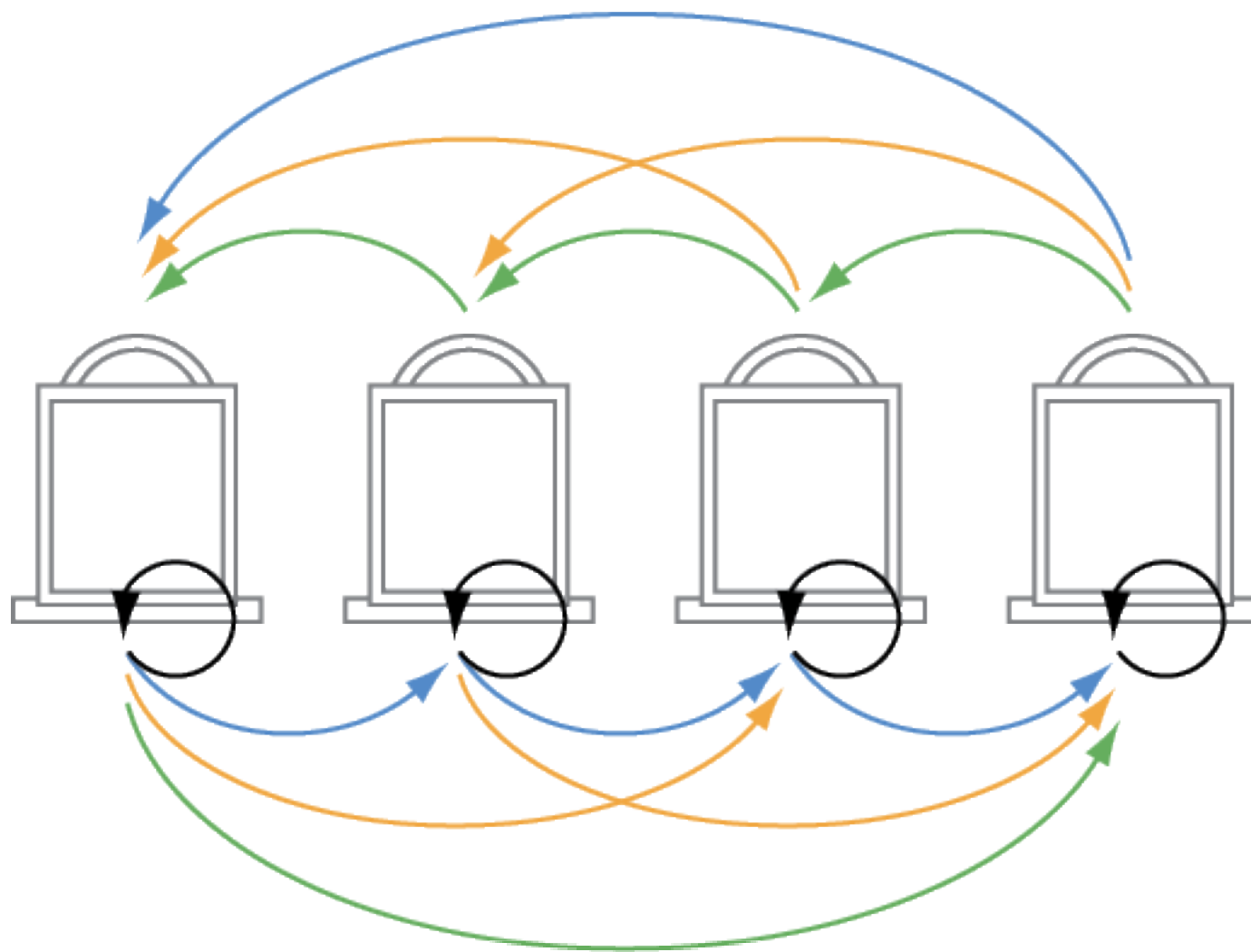
# Transformations



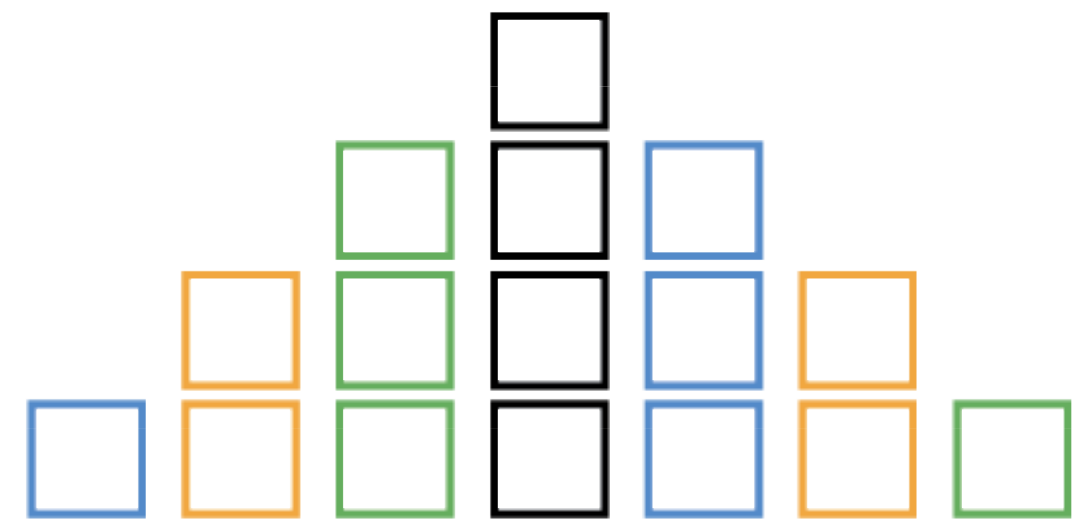
# Transformations



# Transformations

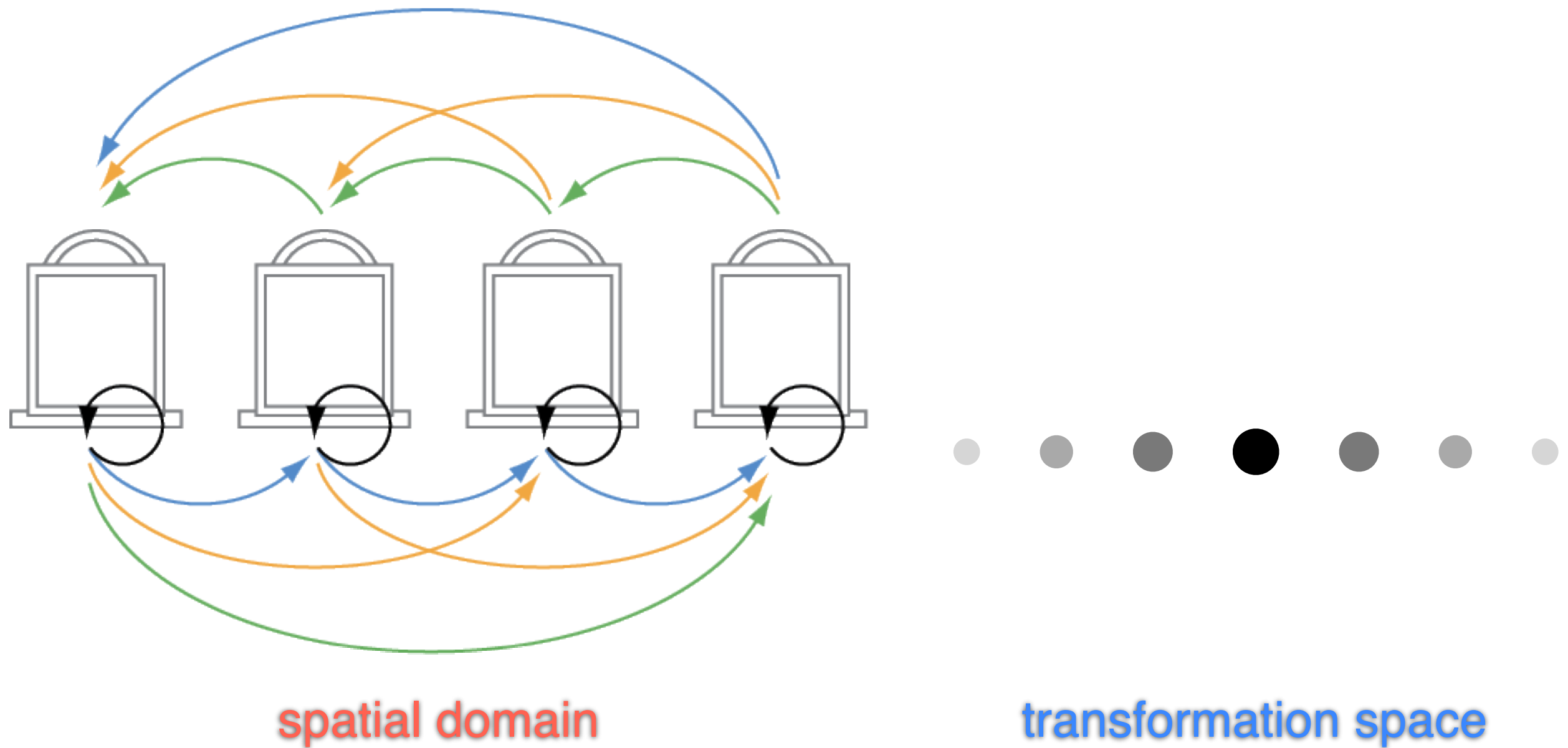


spatial domain

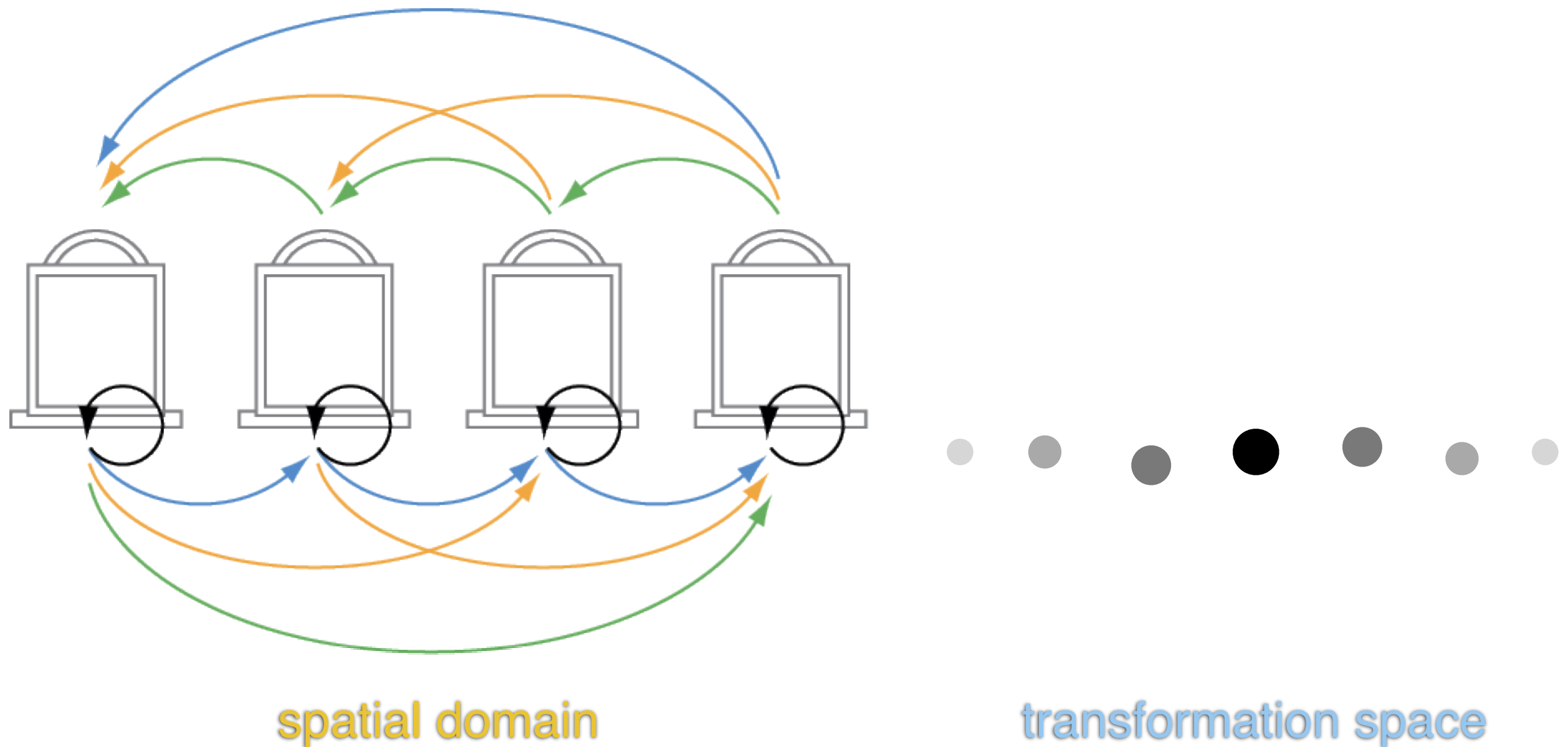


transformation space

# Transformations

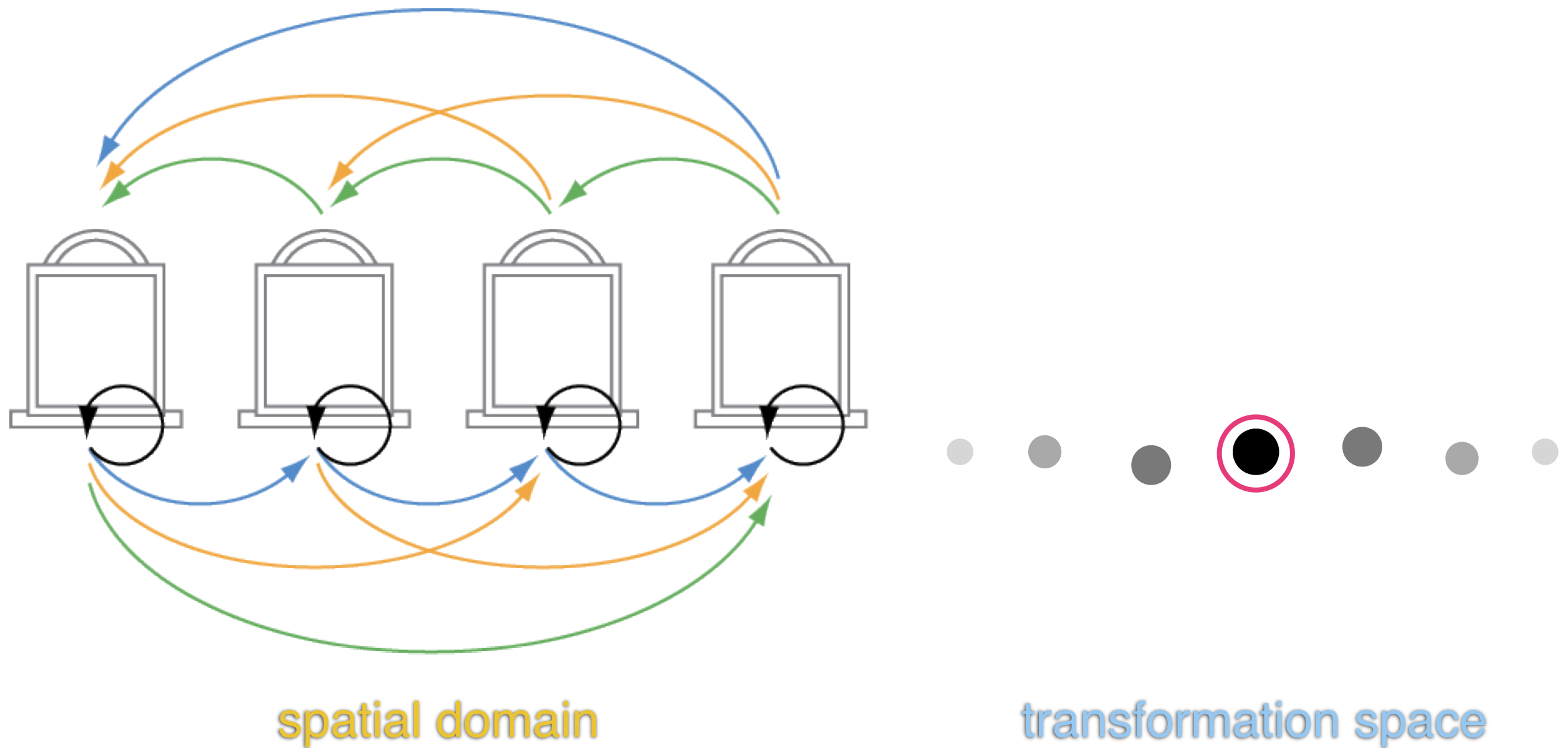


# Transformations

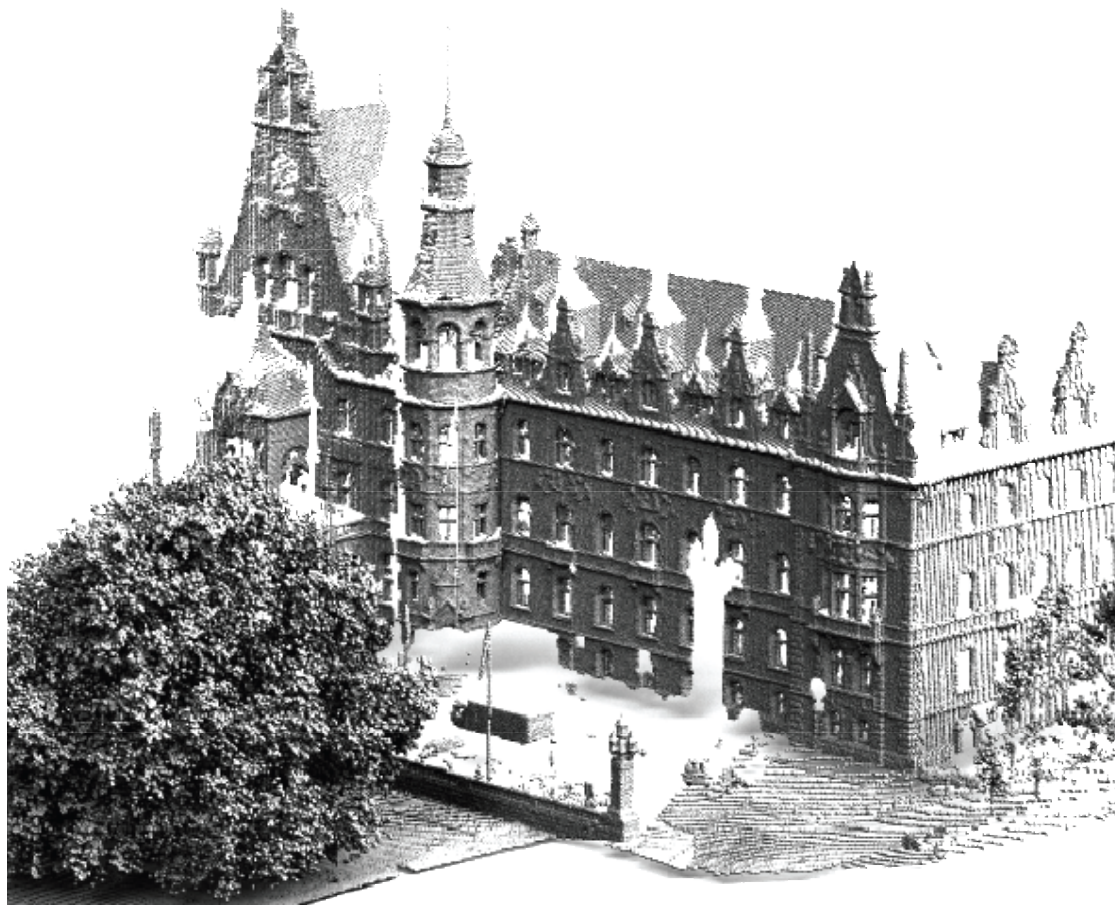




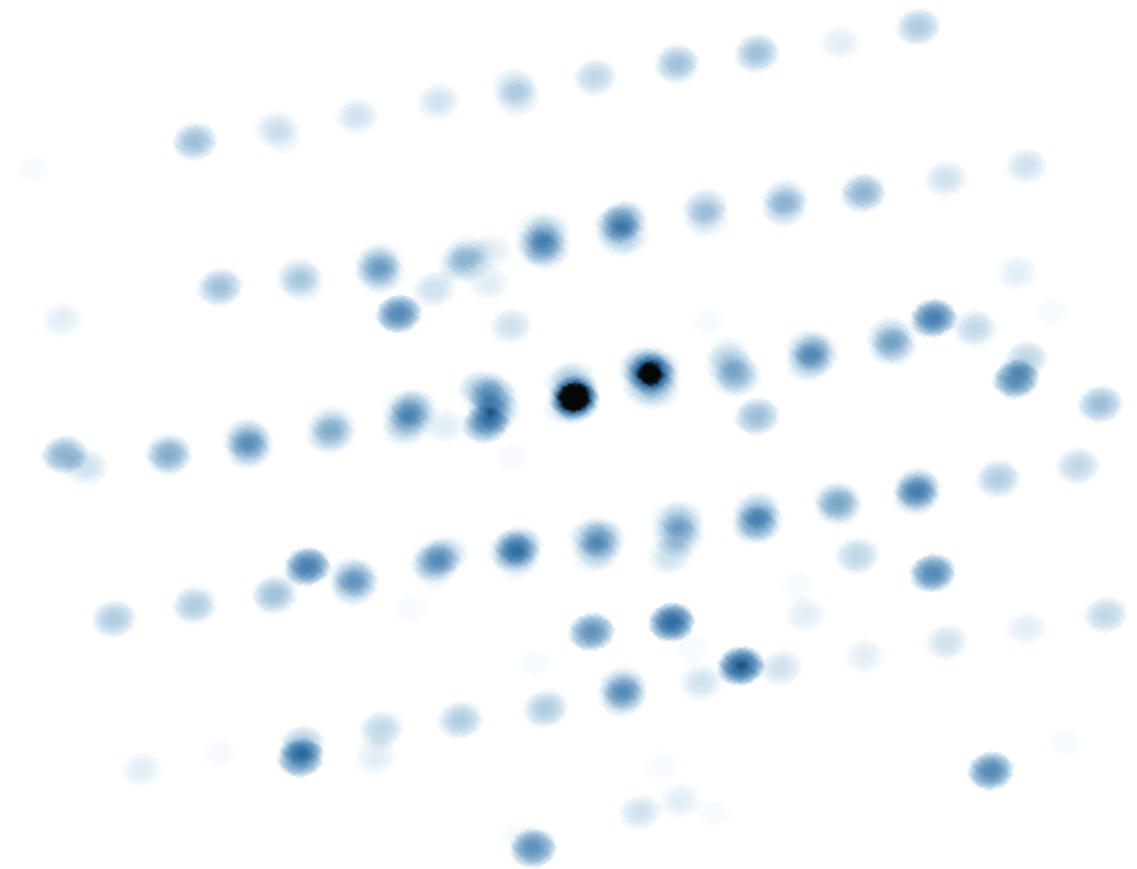
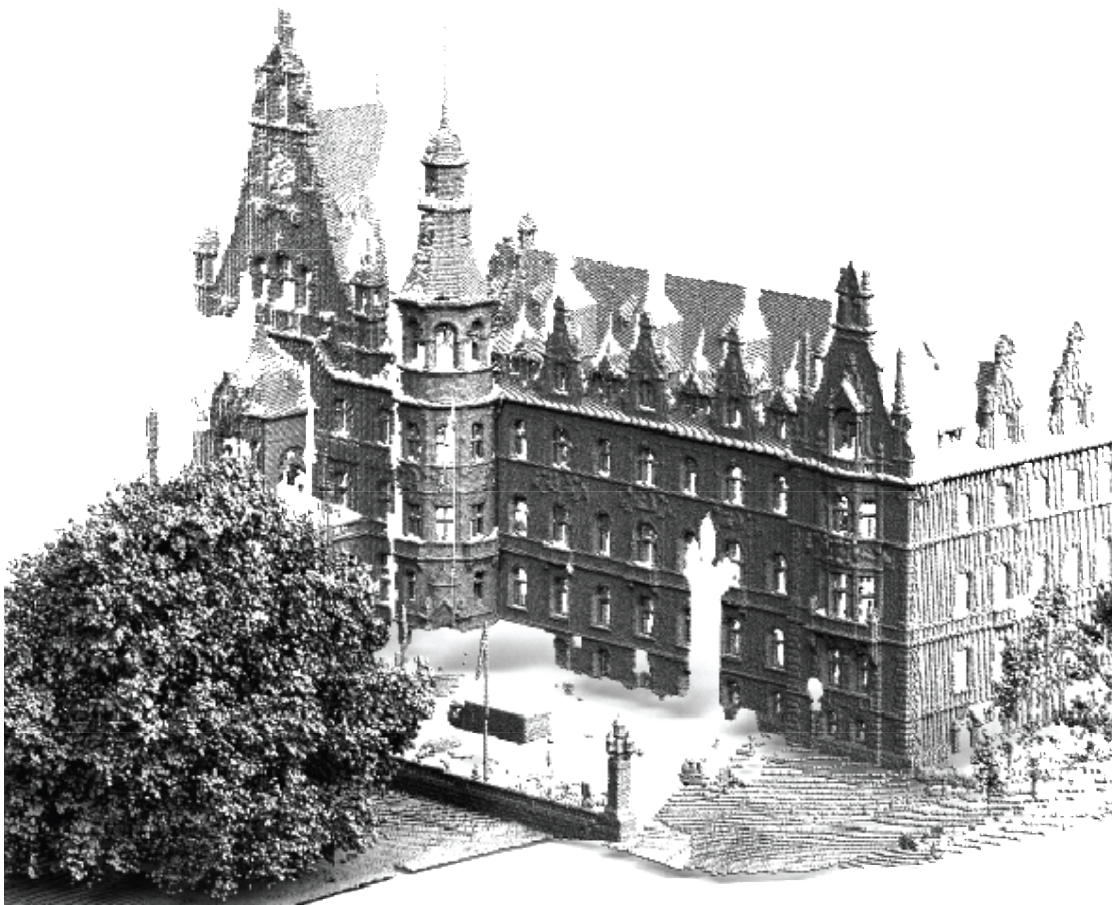
# Transformations



# Model Estimation



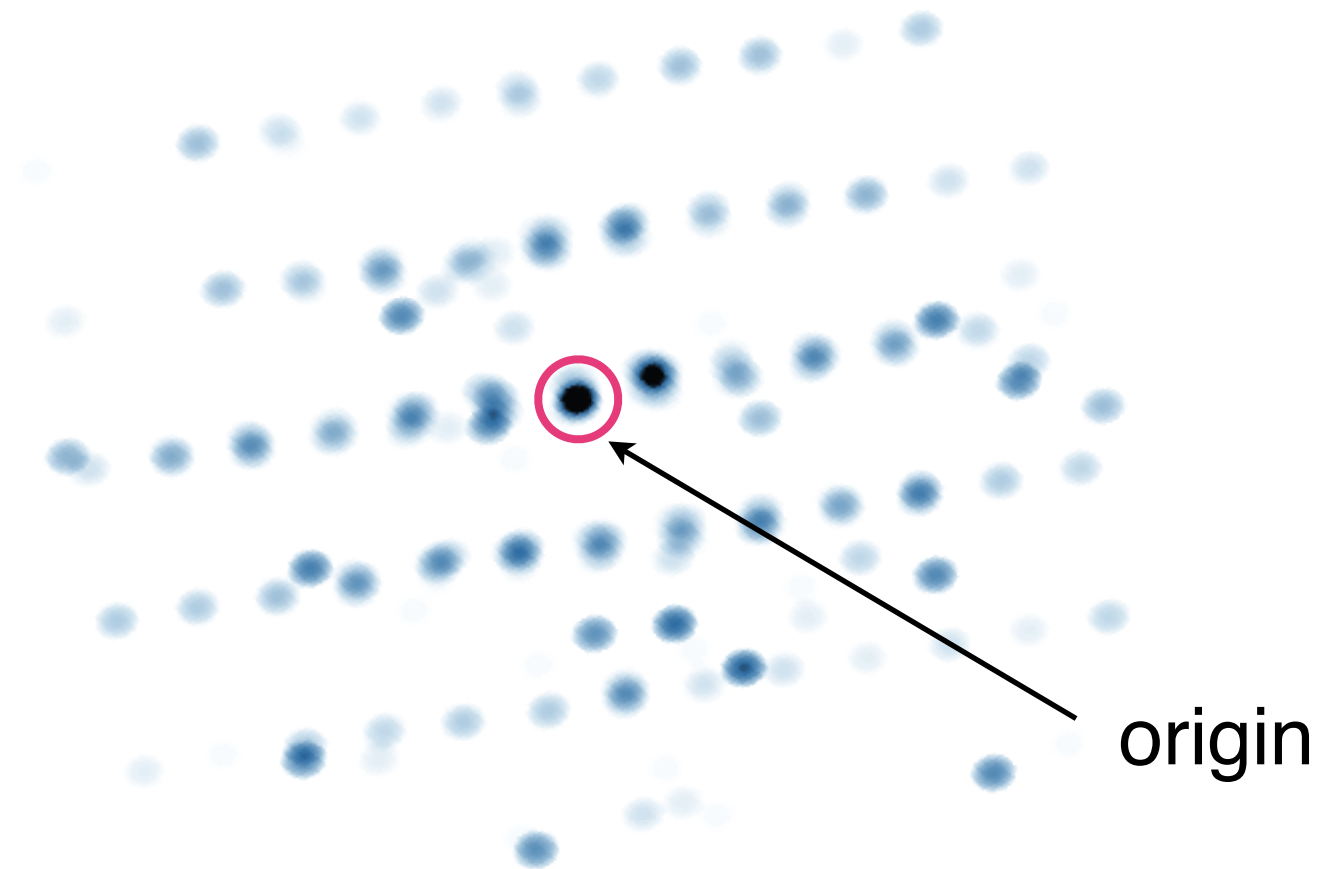
# Model Estimation



density plot of  
pair-wise transformations

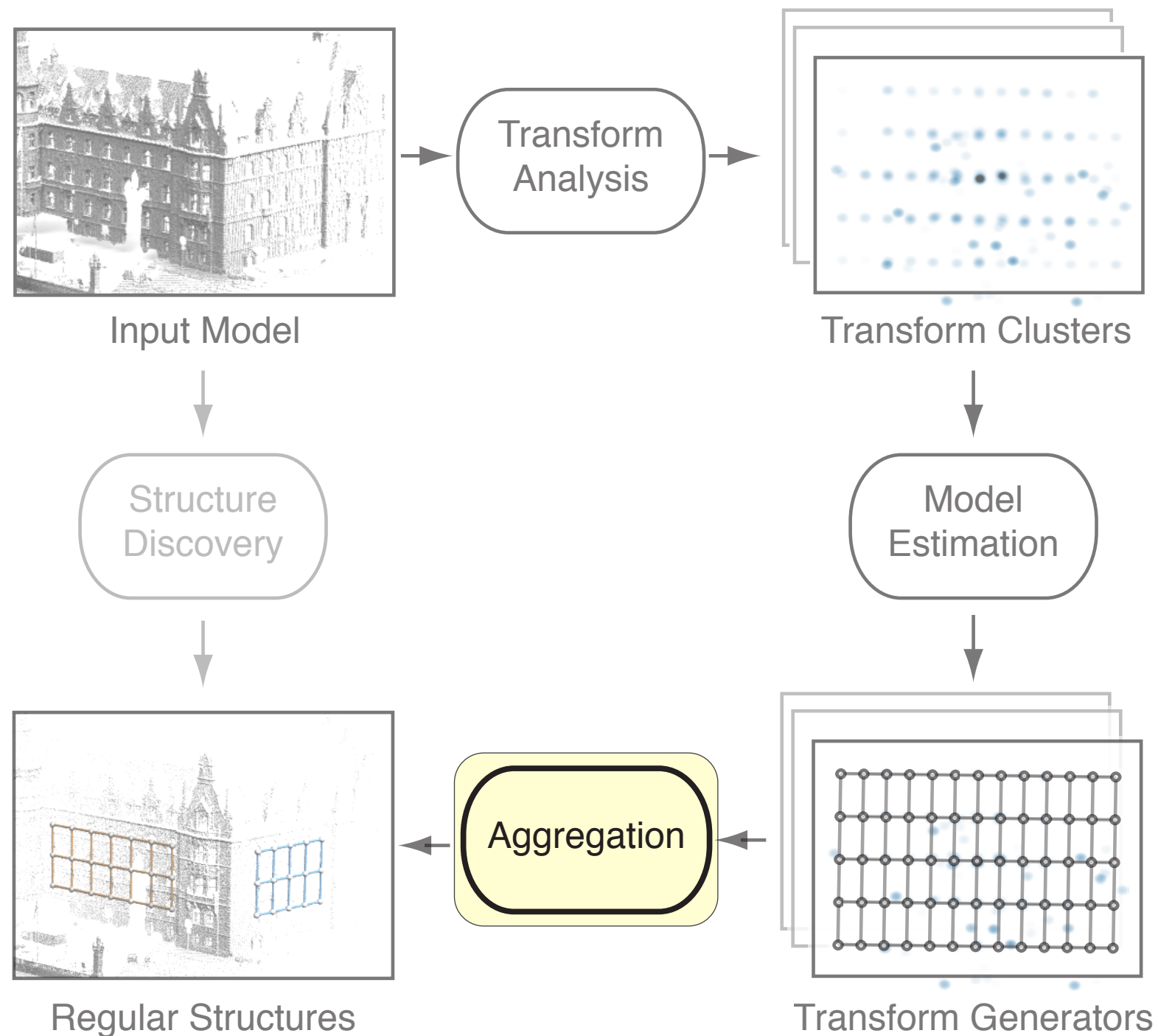


# Model Estimation

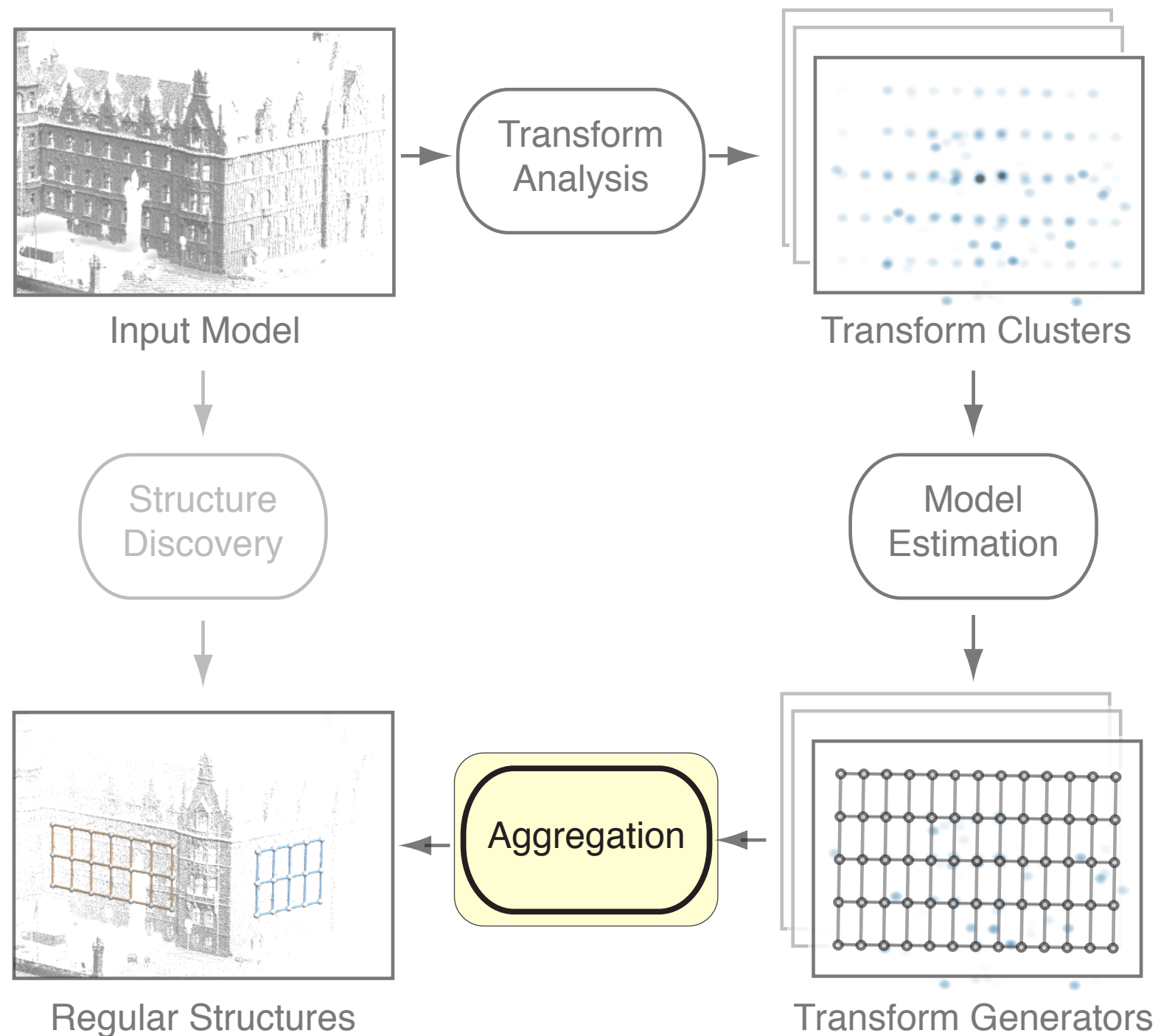


density plot of  
pair-wise transformations

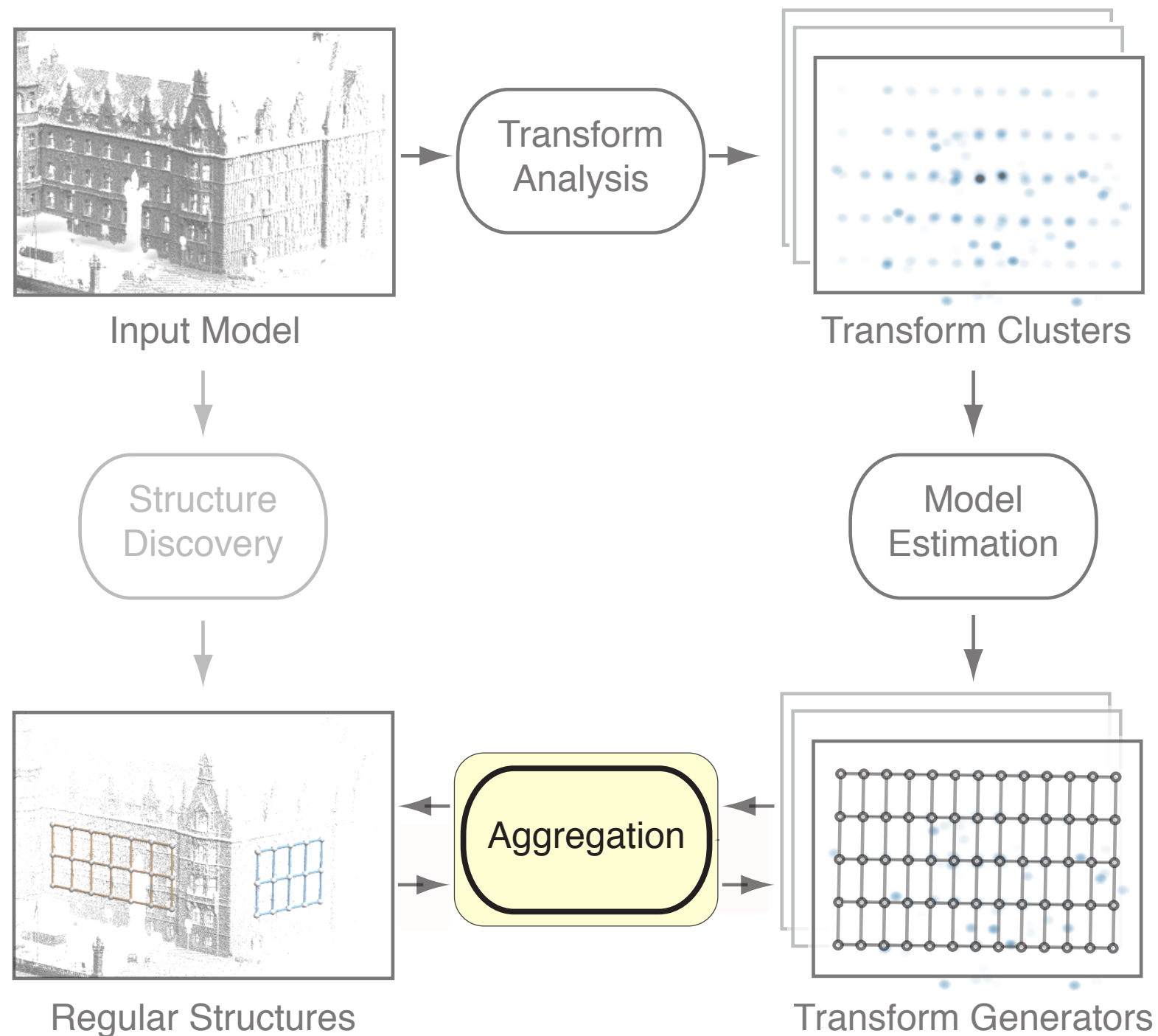
# Structure Discovery



# Structure Discovery

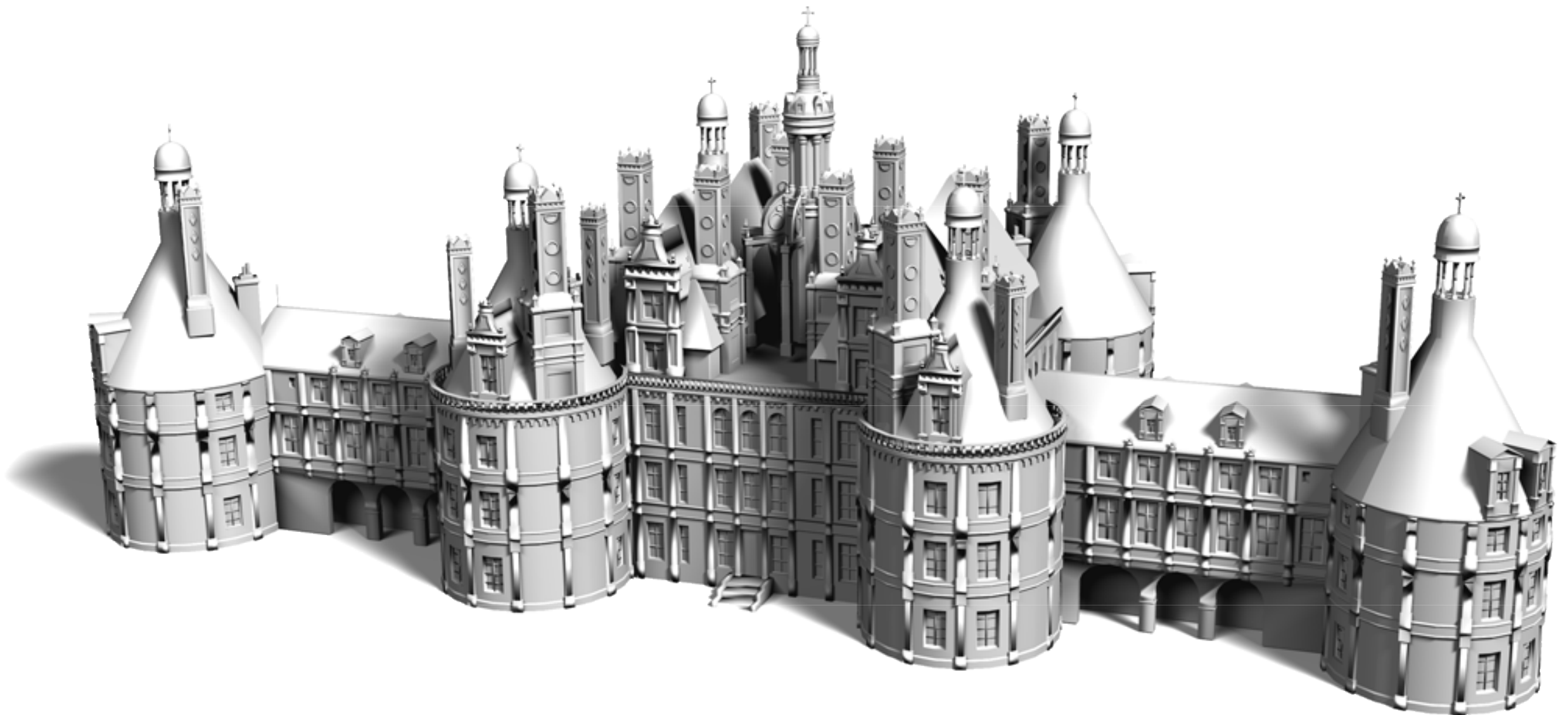


# Structure Discovery

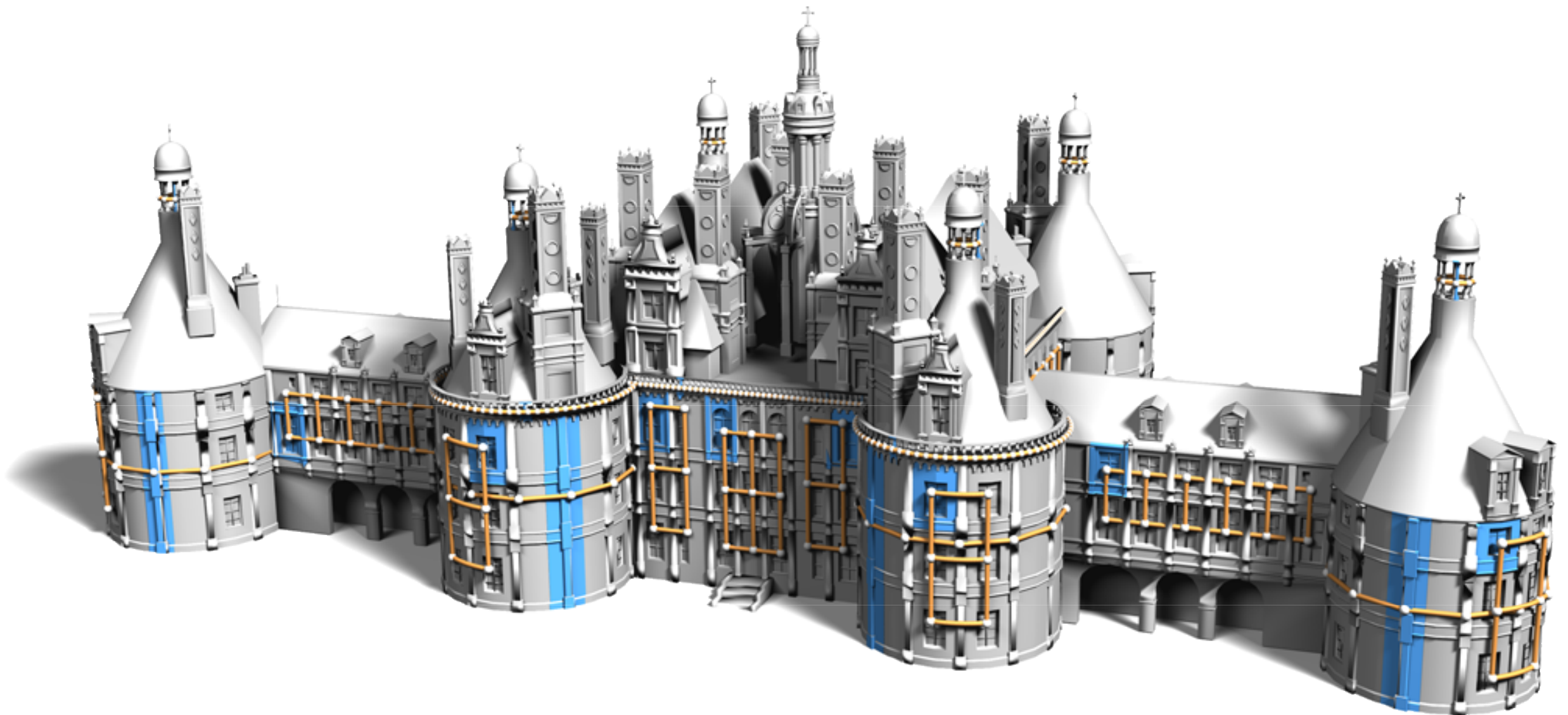




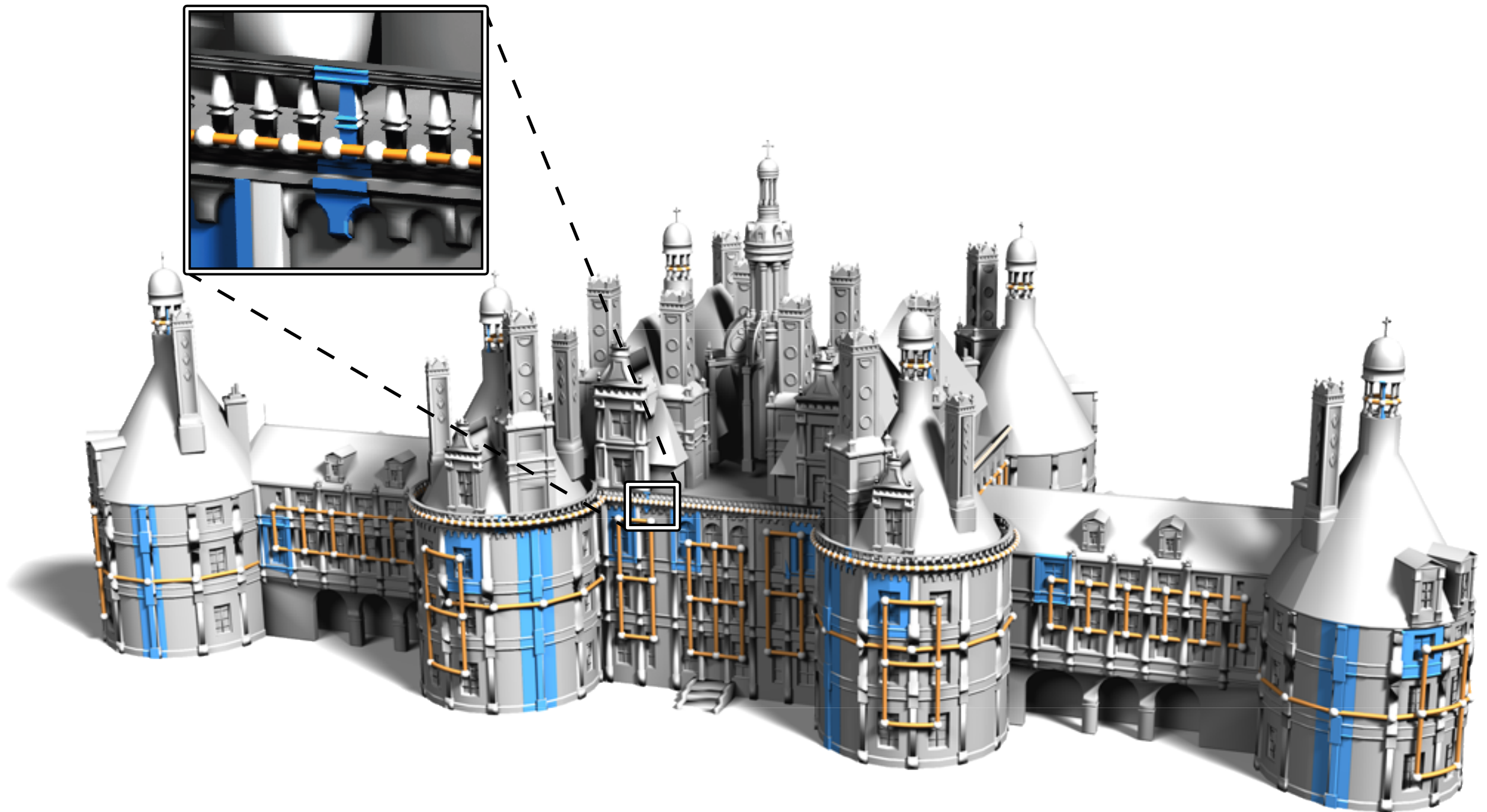
# Chambord Castle



# Chambord Castle

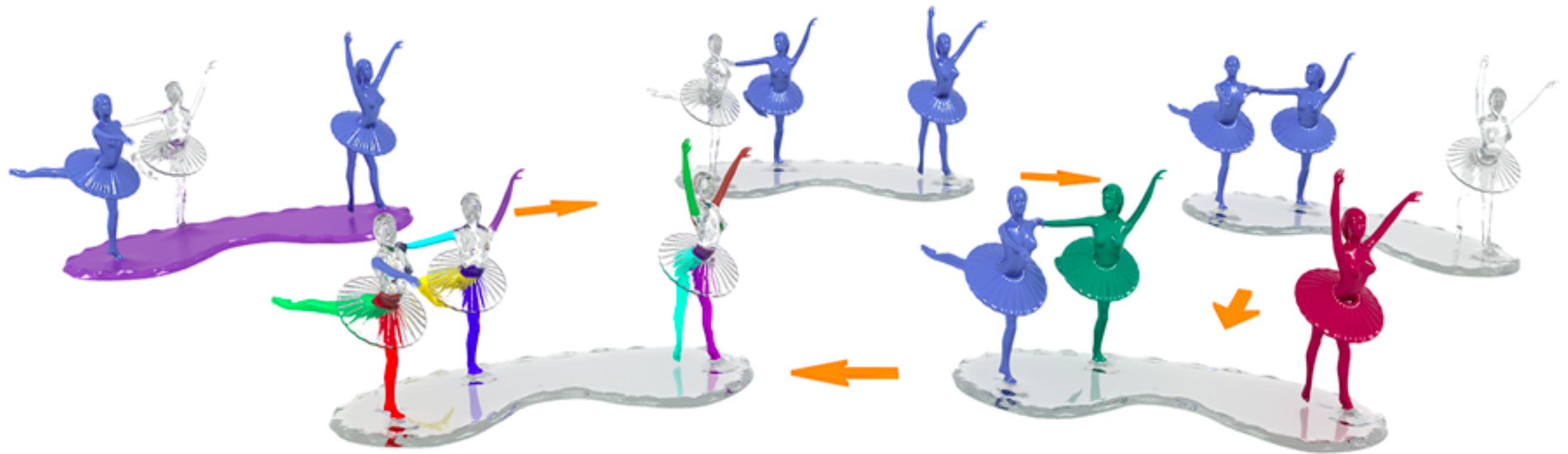


# Chambord Castle





# Multi-scale Symmetry Detection



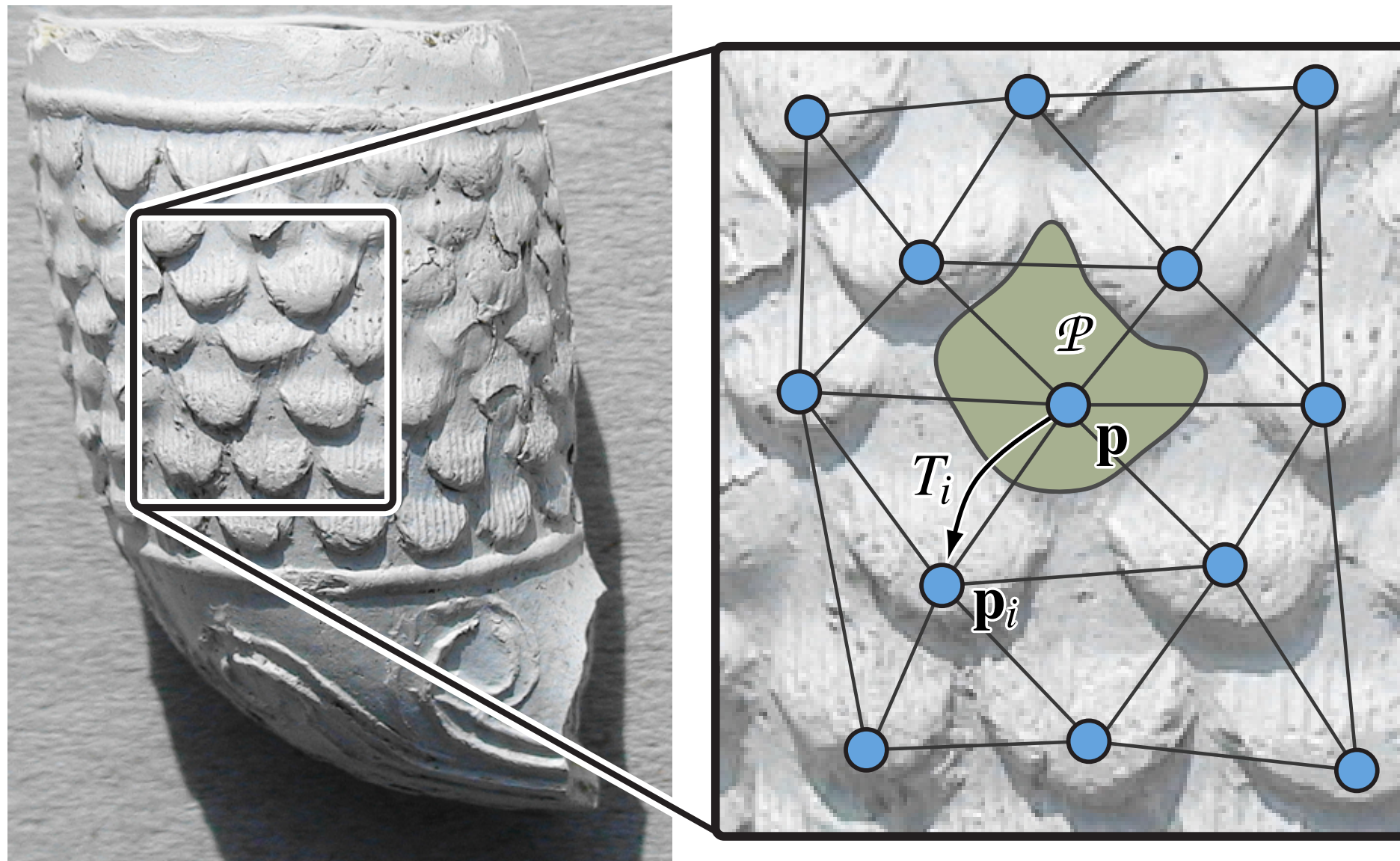
[Xu et al., Siggraph Asia 2012]

# Near Regular Structures



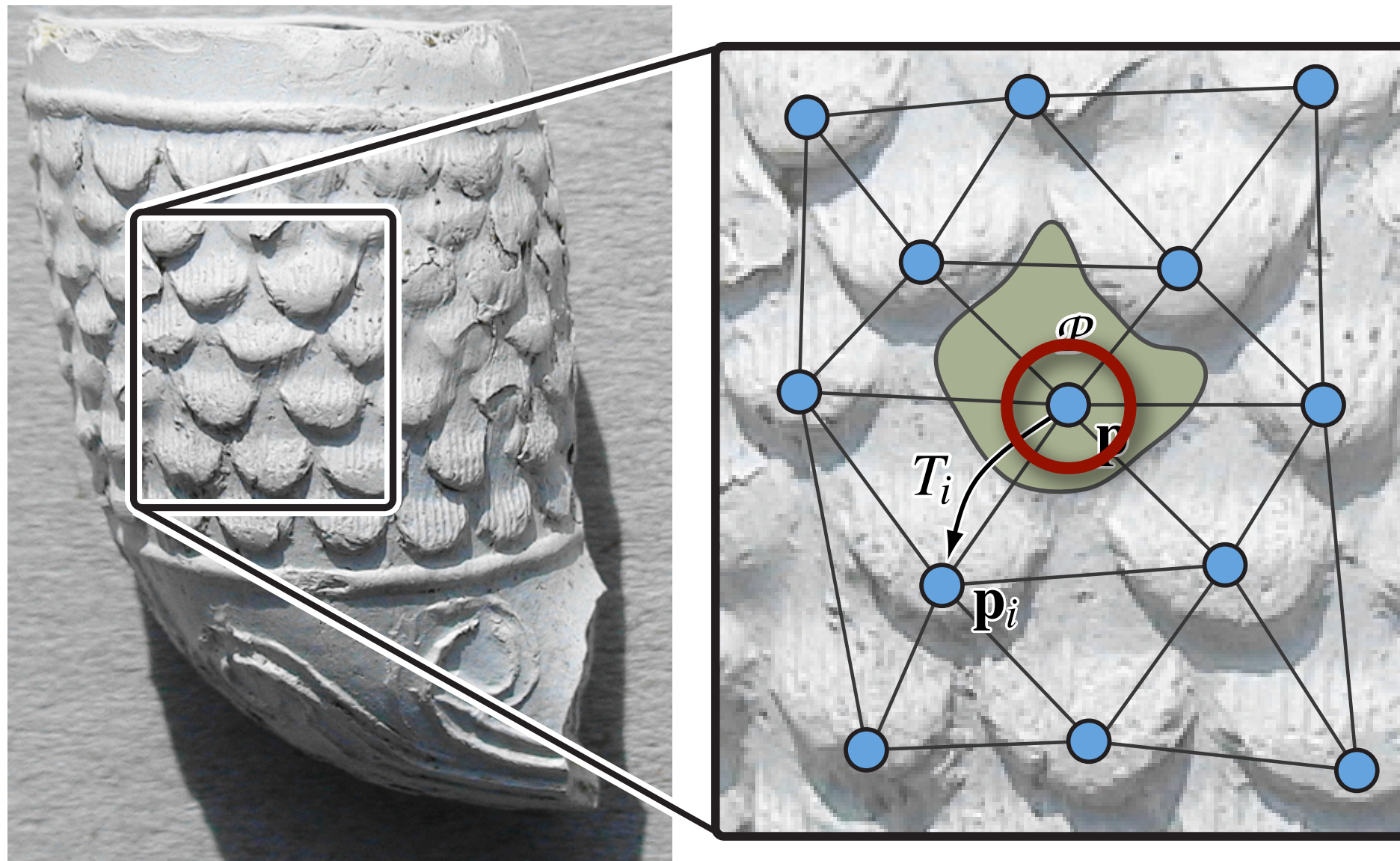
[Huang et al., TOG 2013]

# Near Regular Structures





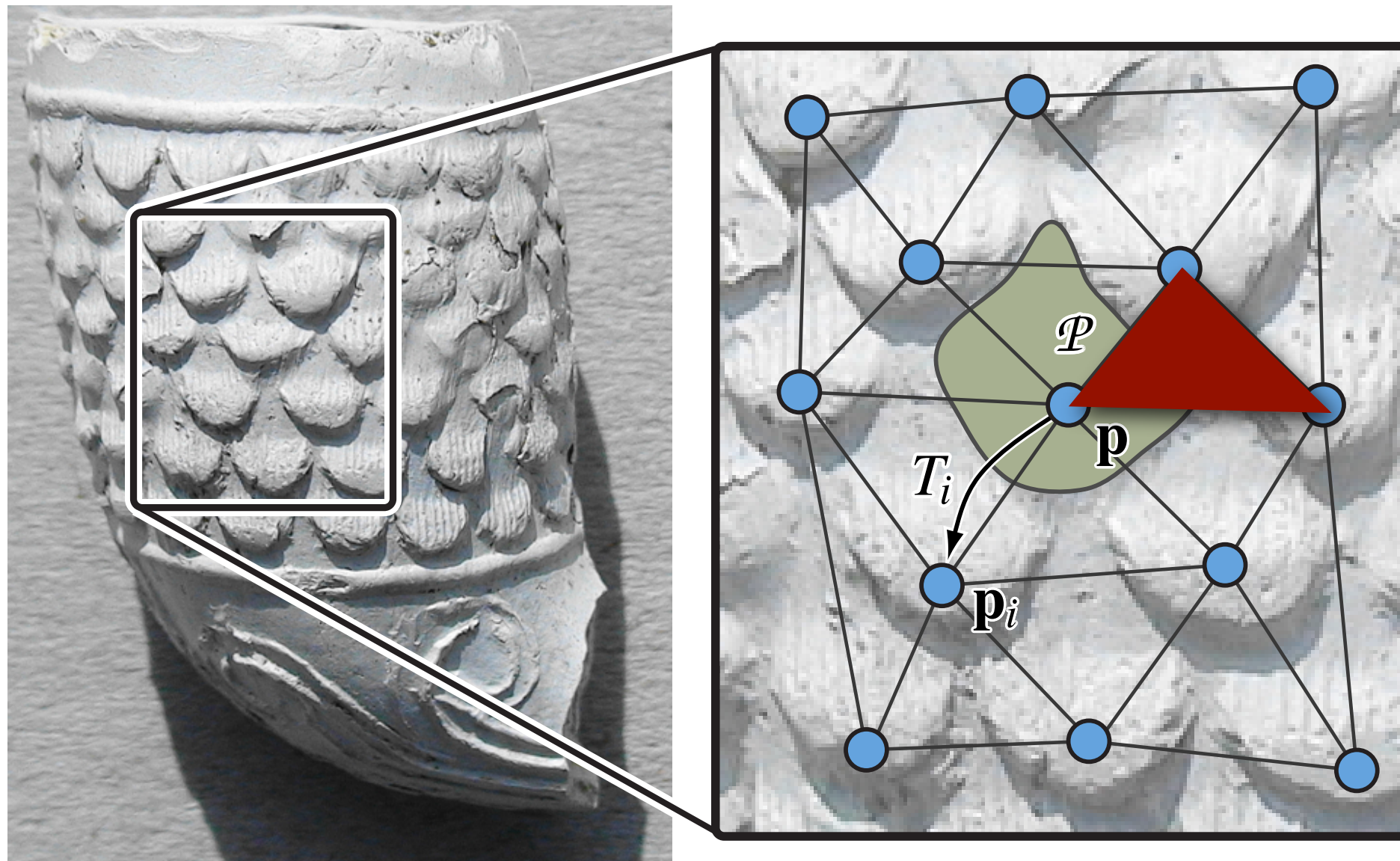
# Near Regular Structures



vertex regularity



# Near Regular Structures



face regularity

# Integer Programming Formulation

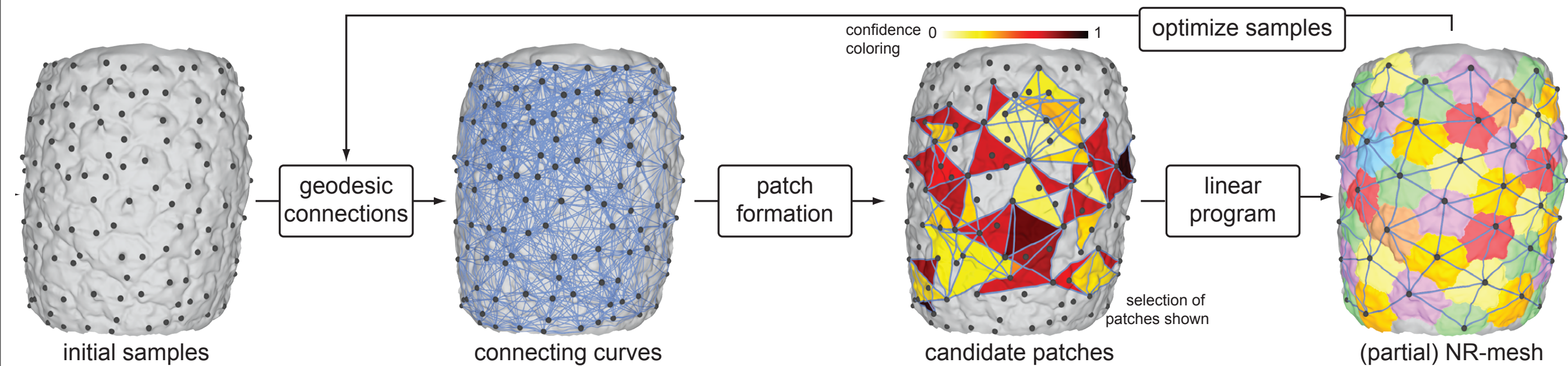
---

$$\begin{array}{ll}\min & \\z_P, x_v \in \{0,1\} & \\ \text{s.t.} & \end{array}$$

Regularity Energy

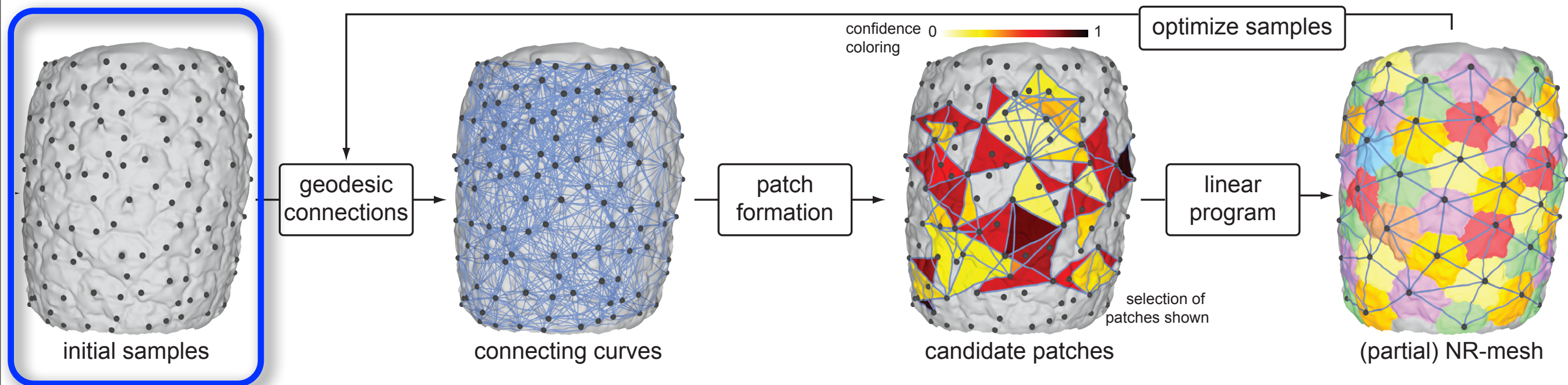
patch collection  
results in a manifold mesh

# Basic Approach

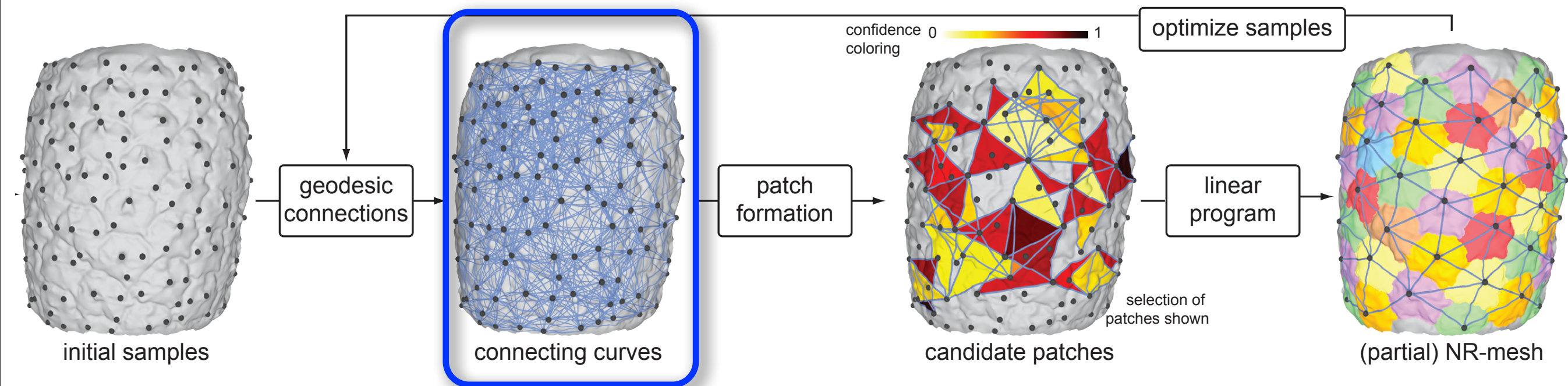




# Basic Approach

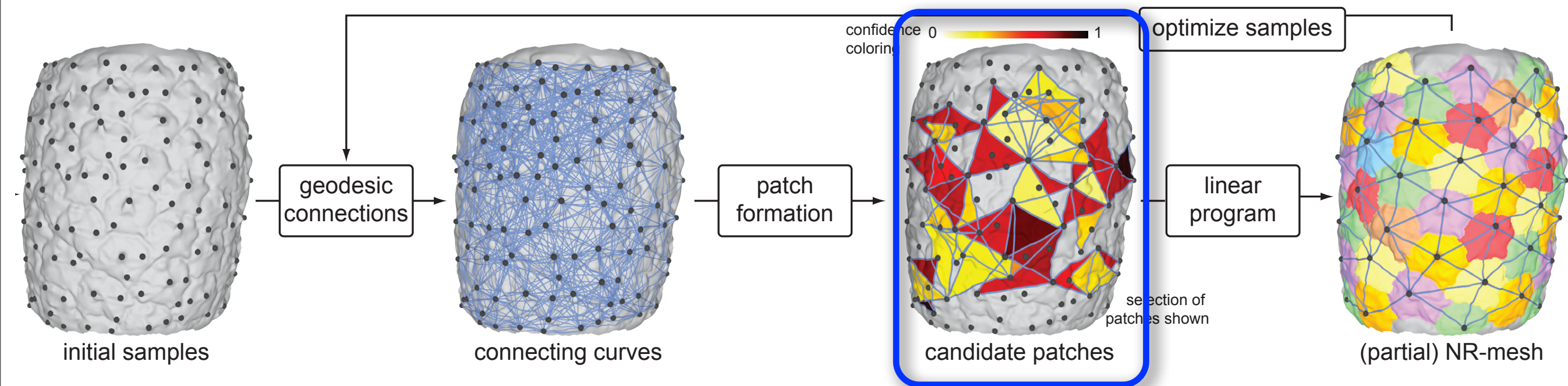


# Basic Approach



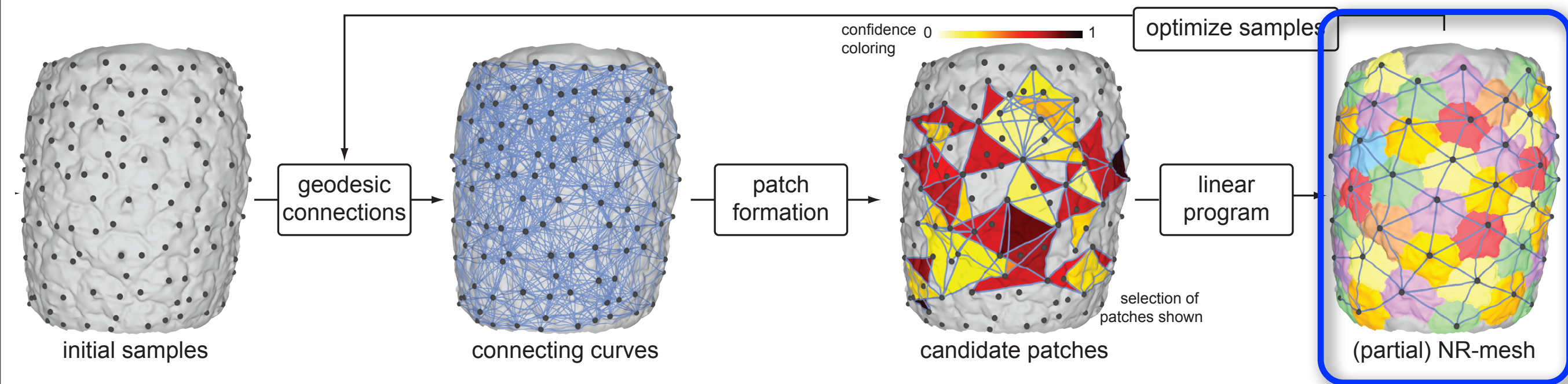


# Basic Approach





# Basic Approach



# Candidate Curves/Patches

---

# Candidate Curves/Patches

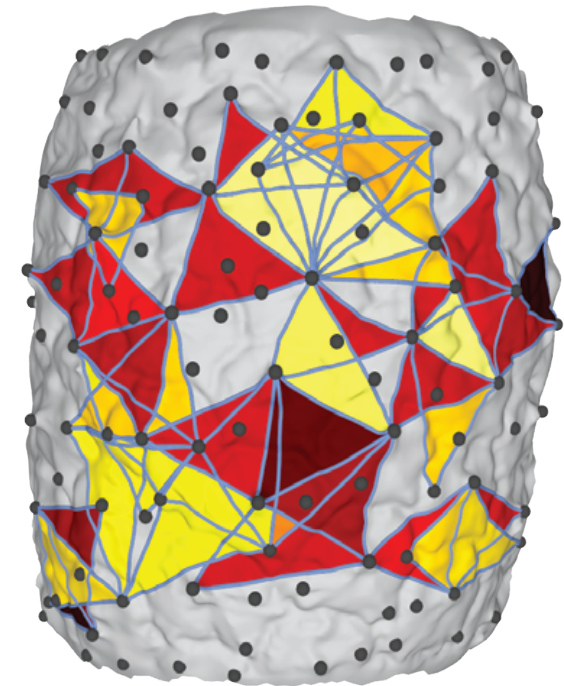
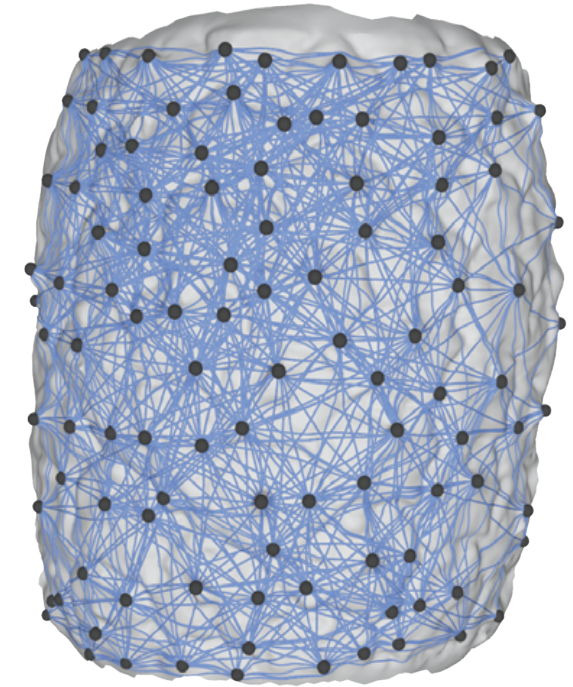
---

- **candidate curves:**
  - geodesic curves connecting samples

# Candidate Curves/Patches



- **candidate curves:**
  - geodesic curves connecting samples
- **candidate patches:**
  - self-intersection free loops of 3/4 curves



# Regularity Energy

---

# Regularity Energy

---

$z_p = 1$ , if patch  $p$  is selected.



# Regularity Energy

$z_p = 1$ , if patch  $p$  is selected.

**face regularity**

$$\sum_p w_p z_p$$

# Regularity Energy

$z_p = 1$ , if patch  $p$  is selected.

**face regularity**  $\sum_p w_p z_p$

$x_v = 1$ , if vertex  $v$  is selected.

# Regularity Energy

$z_p = 1$ , if patch  $p$  is selected.

**face regularity**

$$\sum_p w_p z_p$$

$x_v = 1$ , if vertex  $v$  is selected.

**vertex regularity**

$$\sum_v w_v \left| \sum_{p \in N(v)} z_p - \eta x_v \right|$$

# Integer Programming Formulation

$$\begin{array}{ll}\min & \\z_P, x_v \in \{0,1\} & \\ \text{s.t.} & \end{array}$$

Regularity Energy

patch collection  
results in a manifold mesh

# Integer Programming Formulation

$$\min_{z_P, x_v \in \{0,1\}} \quad - \sum_{P \in \mathcal{P}} w_P z_P + \lambda_v \sum_{v \in \mathcal{S}} w_v \left| \sum_{P \in \mathcal{A}_P(v)} z_P - \eta x_v \right|$$

s . t .

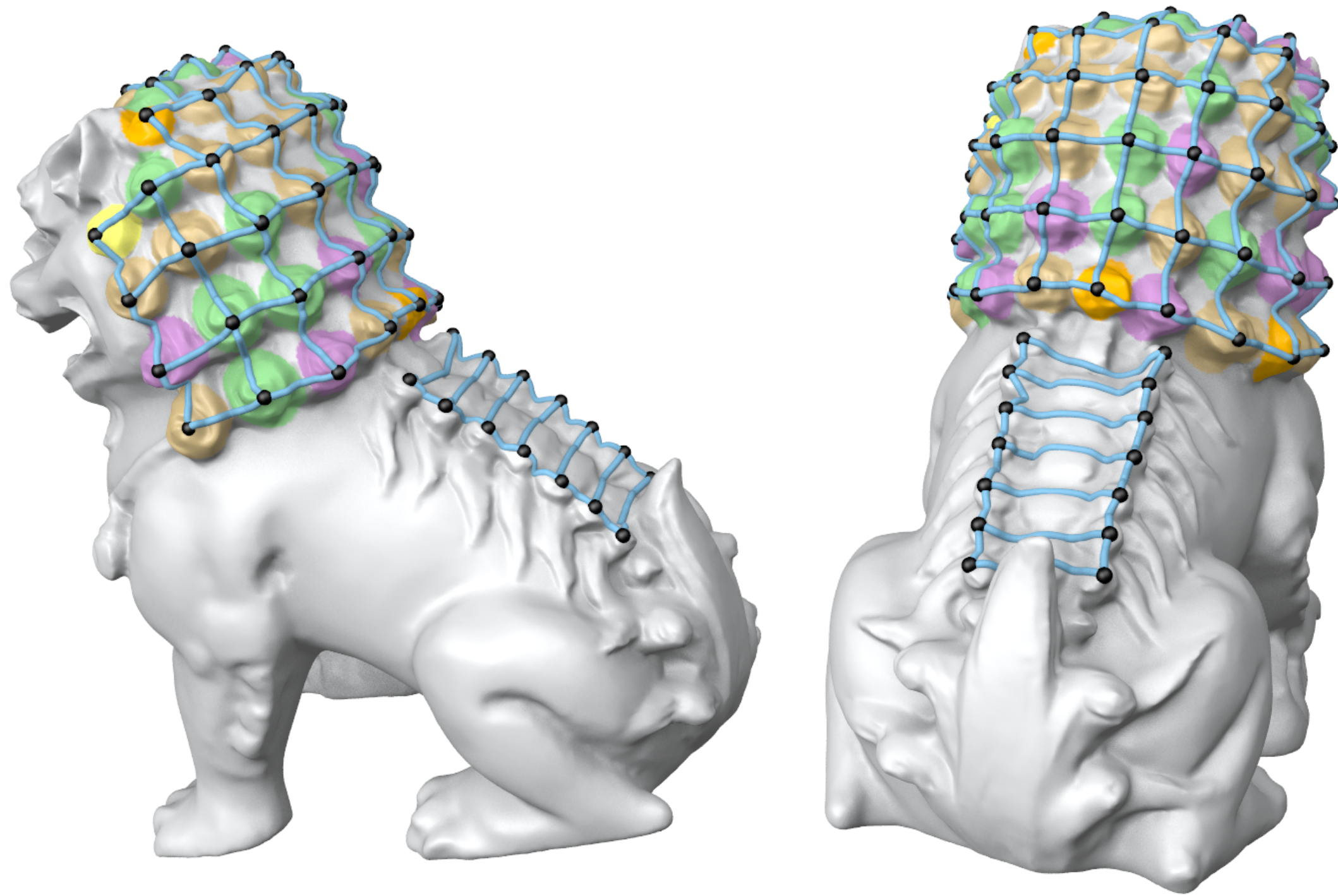
patch collection  
results in a manifold mesh

# Integer Programming Formulation

$$\begin{aligned} \min_{z_P, x_v \in \{0,1\}} \quad & - \sum_{P \in \mathcal{P}} w_P z_P + \lambda_v \sum_{v \in \mathcal{S}} w_v \left| \sum_{P \in \mathcal{A}_P(v)} z_P - \eta x_v \right| \\ \text{s.t.} \quad & z_P \leq x_v, \quad \forall v \in \mathcal{P}, \quad \forall P \in \mathcal{P}, \\ & \sum_{P \in \mathcal{A}_L(c)} z_P = \sum_{P \in \mathcal{A}_R(c)} z_P, \quad c \in \mathcal{C}_I, \\ & \sum_{P \in \mathcal{C}(f)} z_P = 1. \end{aligned}$$



# Asian Lion

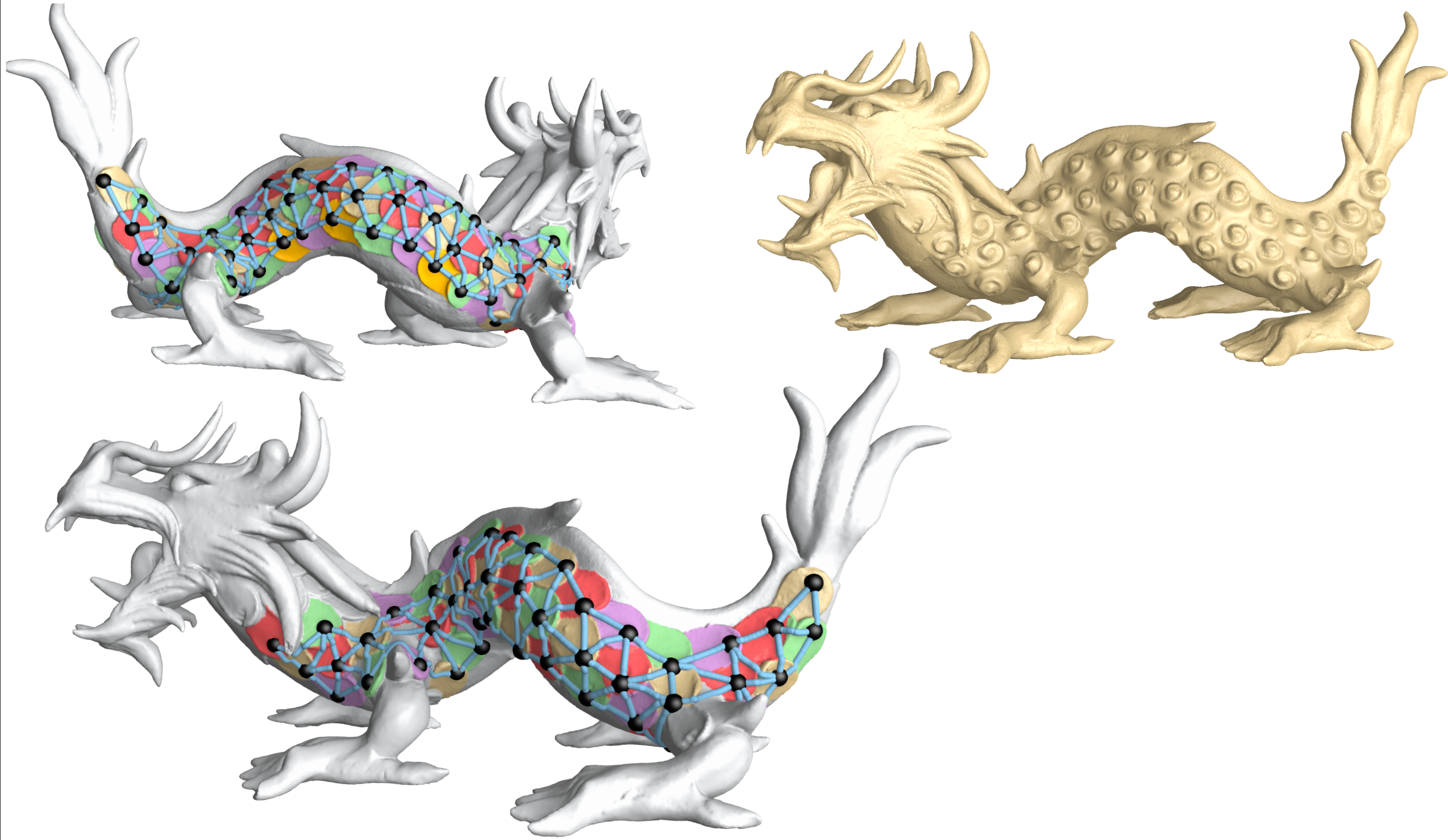


# Near-regular Structures





# Near-regular Structures



an unified view

# Constrained Shape Space

---



- **Given:**  
single *constrained* shape (primitives + relations)

[Yang et al., SiggraphA 2011]



# Constrained Shape Space

- **Given:**  
single *constrained* shape (primitives + relations)
- **Goal:**
  - characterize/navigate *neighboring* constrained shapes
  - navigate only the *good* ones

[Yang et al., SiggraphA 2011]



# Constrained Shape Space

---

$$\mathbf{x} = (v_1, \dots, v_n) \in \mathbb{R}^D$$

# Constrained Shape Space

---

$$\mathbf{x} = (v_1, \dots, v_n) \in \mathbb{R}^D$$

- mesh  $\rightarrow$  point

# Constrained Shape Space

$$\mathbf{x} = (v_1, \dots, v_n) \in \mathbb{R}^D$$

- mesh  $\rightarrow$  point
- combinatorics remain fixed

# Constrained Shape Space

$$\mathbf{x} = (v_1, \dots, v_n) \in \mathbb{R}^D$$

- mesh  $\rightarrow$  point
- combinatorics remain fixed
- starting mesh  $\mathbf{x}_0$  satisfies (nonlinear) constraints

# Intersection Surface

Each (face) constraint/relation

$$\Gamma_i := \{\mathbf{x} \in \mathbb{R}^D : E_i(\mathbf{x}) = 0\} \quad \forall \quad i = 1, \dots, m$$

# Intersection Surface

Each (face) constraint/relation

$$\Gamma_i := \{\mathbf{x} \in \mathbb{R}^D : E_i(\mathbf{x}) = 0\} \quad \forall \quad i = 1, \dots, m$$

$$\mathbf{d} \Rightarrow \mathbf{x}_0 + \mathbf{d}$$



# Osculant Surface

---

# Osculant Surface

$$\mathbf{S}(\mathbf{u}) = \mathbf{x}_0 + \sum_{i=1}^{D-m} u_i \mathbf{e}_i + \frac{1}{2} \sum_{j=1}^m (\mathbf{u}^T \cdot A_j \cdot \mathbf{u}) \mathbf{n}_j$$

# Osculant Surface

$$\mathbf{S}(\mathbf{u}) = \mathbf{x}_0 + \sum_{i=1}^{D-m} u_i \mathbf{e}_i + \frac{1}{2} \sum_{j=1}^m (\mathbf{u}^T \cdot A_j \cdot \mathbf{u}) \mathbf{n}_j$$

$$\begin{aligned} E_i(\mathbf{x}) &= E_i(\mathbf{x}_0) + \nabla E_i^T \cdot (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_0)^T \cdot H_i \cdot (\mathbf{x} - \mathbf{x}_0) \\ &\quad + o(\|\mathbf{x} - \mathbf{x}_0\|^2) \end{aligned}$$

# Osculant Surface

$$\mathbf{S}(\mathbf{u}) = \mathbf{x}_0 + \sum_{i=1}^{D-m} u_i \mathbf{e}_i + \frac{1}{2} \sum_{j=1}^m (\mathbf{u}^T \cdot A_j \cdot \mathbf{u}) \mathbf{n}_j$$

$$\begin{aligned} E_i(\mathbf{x}) &= E_i(\mathbf{x}_0) + \nabla E_i^T \cdot (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_0)^T \cdot H_i \cdot (\mathbf{x} - \mathbf{x}_0) \\ &\quad + o(\|\mathbf{x} - \mathbf{x}_0\|^2) \end{aligned}$$

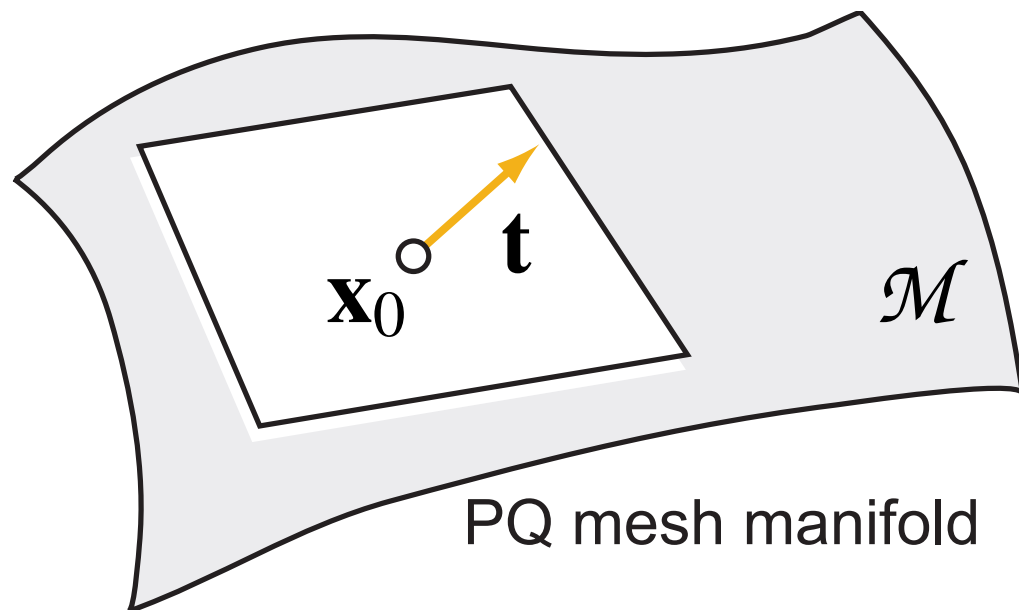
# Osculant Surface

$$\mathbf{S}(\mathbf{u}) = \mathbf{x}_0 + \sum_{i=1}^{D-m} u_i \mathbf{e}_i + \frac{1}{2} \sum_{j=1}^m (\mathbf{u}^T \cdot A_j \cdot \mathbf{u}) \mathbf{n}_j$$

$$\begin{aligned} E_i(\mathbf{x}) &= E_i(\mathbf{x}_0) + \nabla E_i^T \cdot (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_0)^T \cdot H_i \cdot (\mathbf{x} - \mathbf{x}_0) \\ &\quad + o(\|\mathbf{x} - \mathbf{x}_0\|^2) \end{aligned}$$

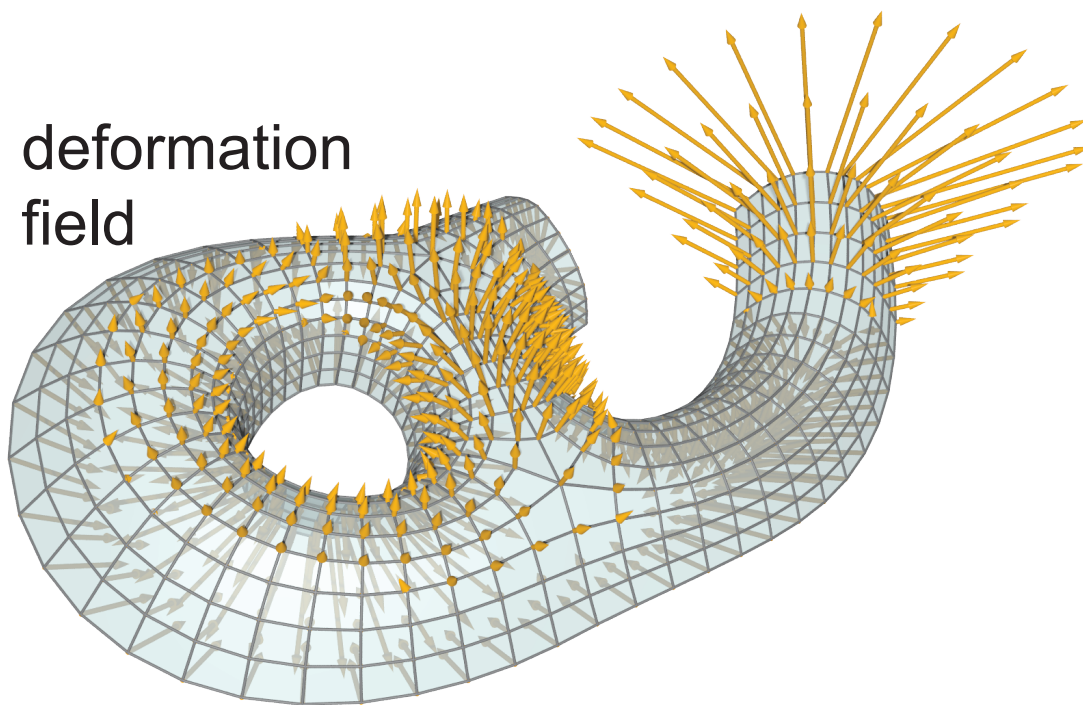
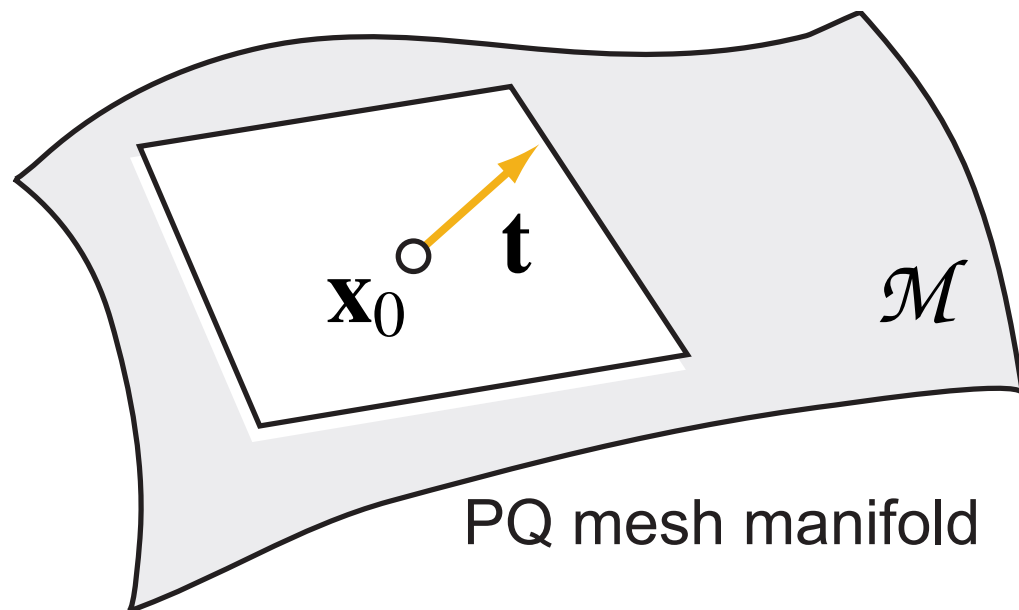
# Walking on the Tangent Space

---

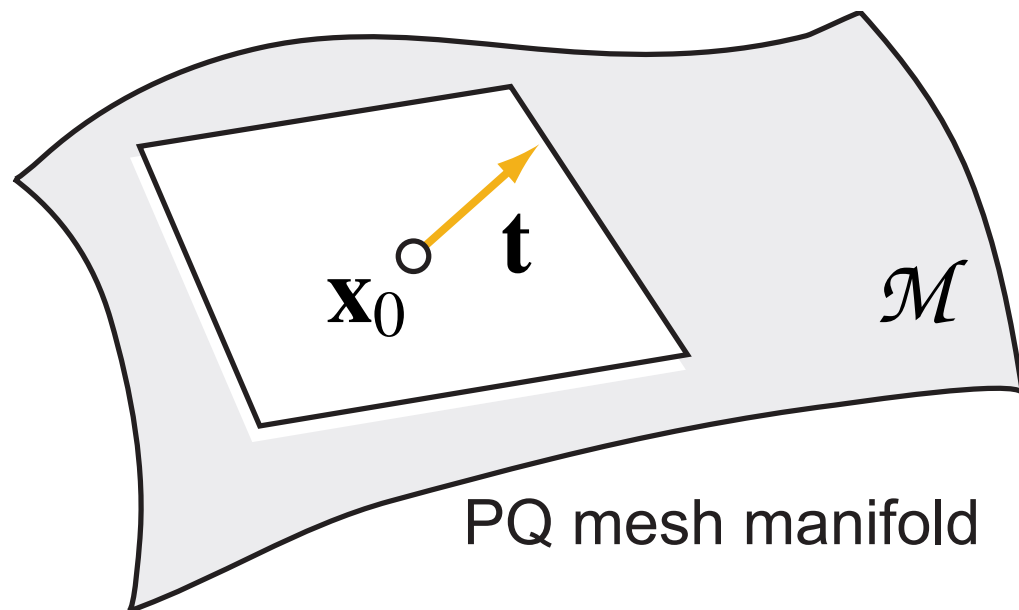




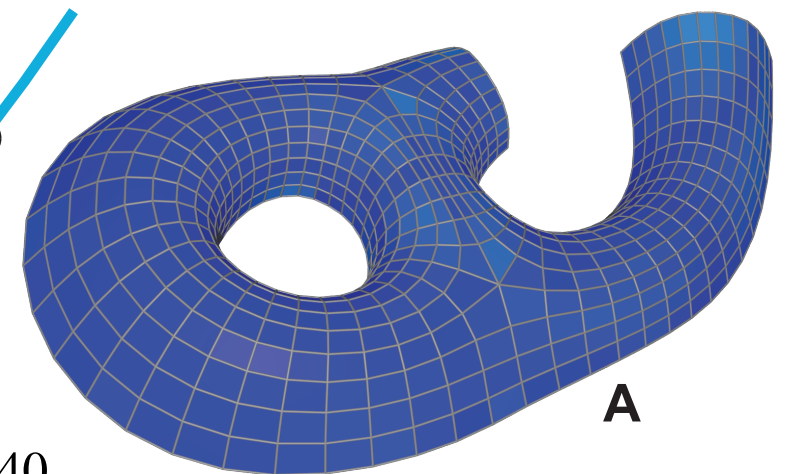
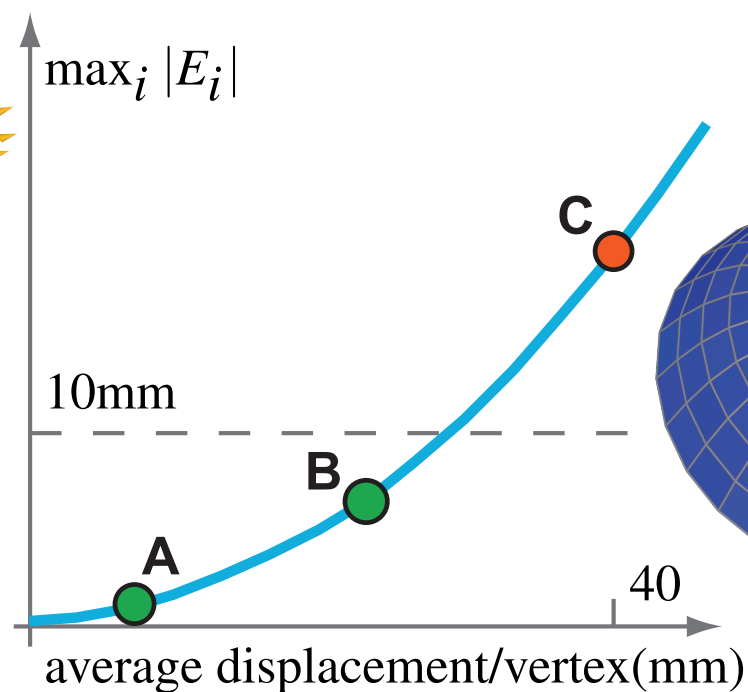
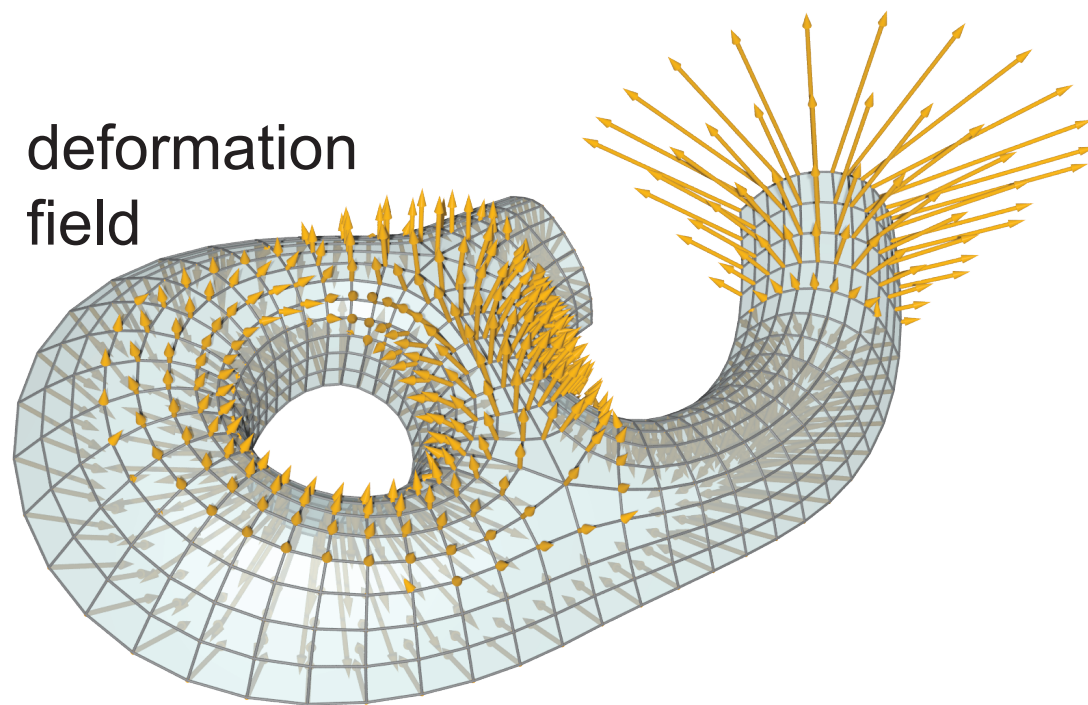
# Walking on the Tangent Space



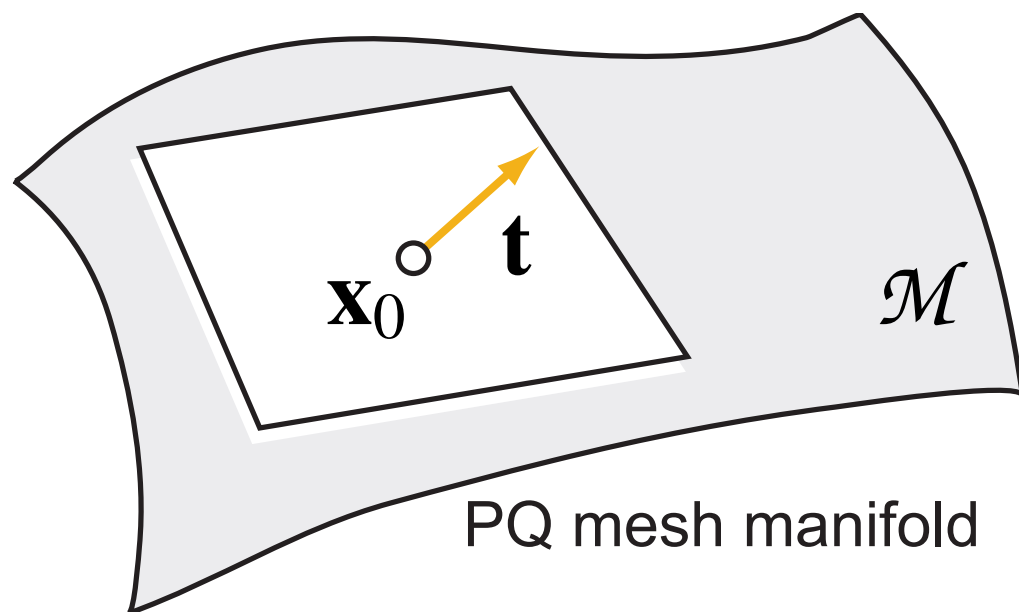
# Walking on the Tangent Space



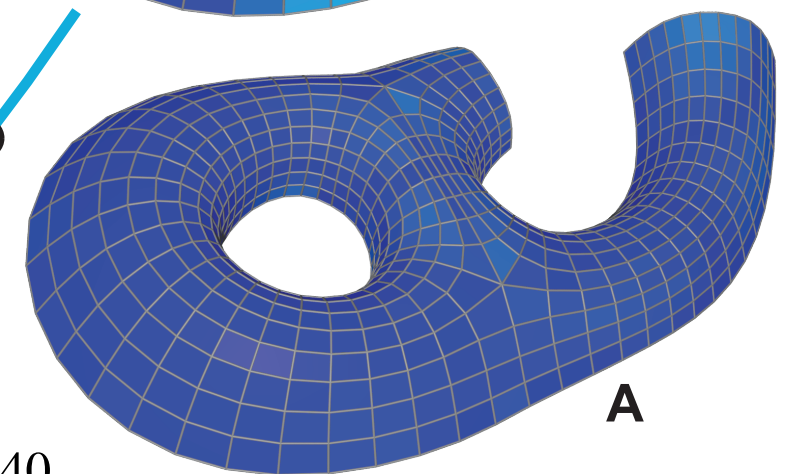
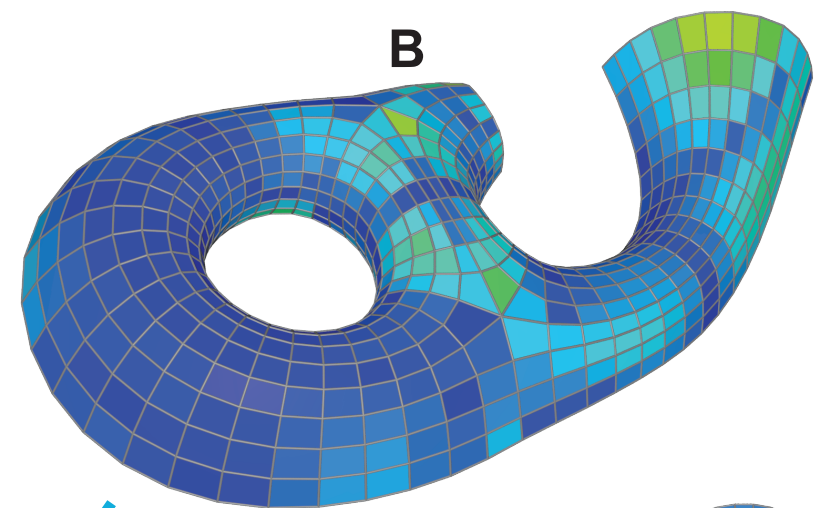
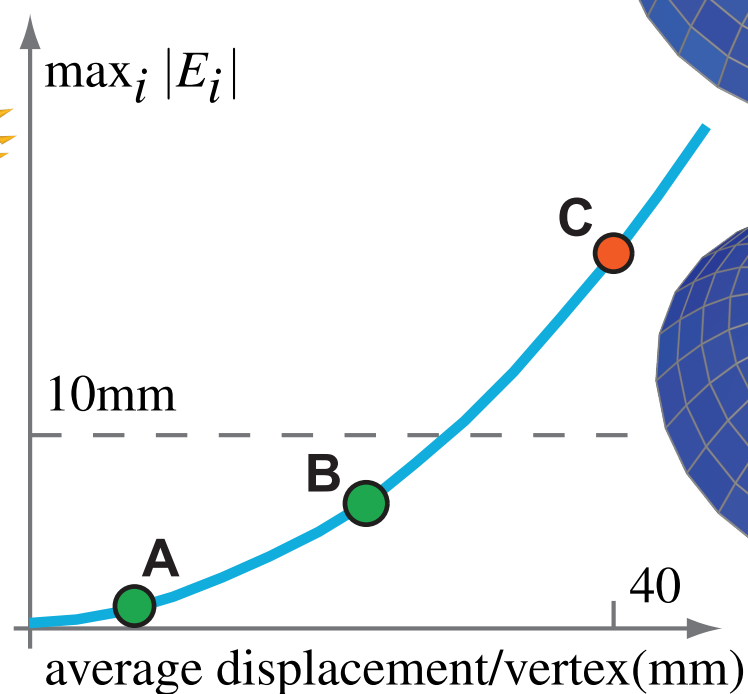
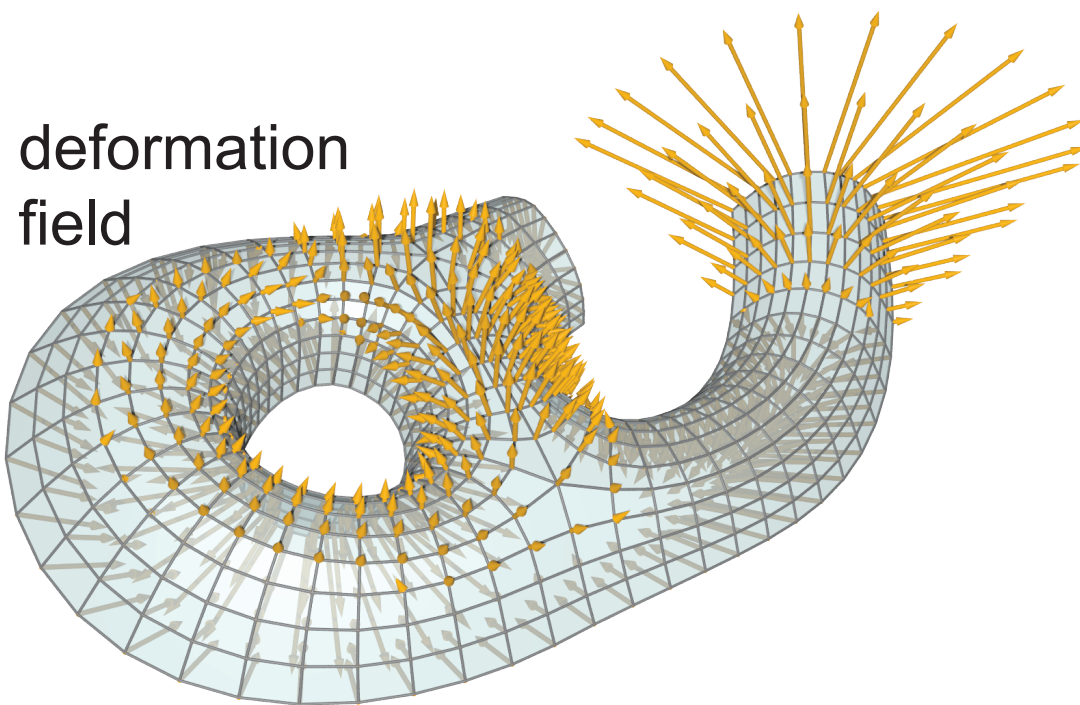
0  10mm



# Walking on the Tangent Space

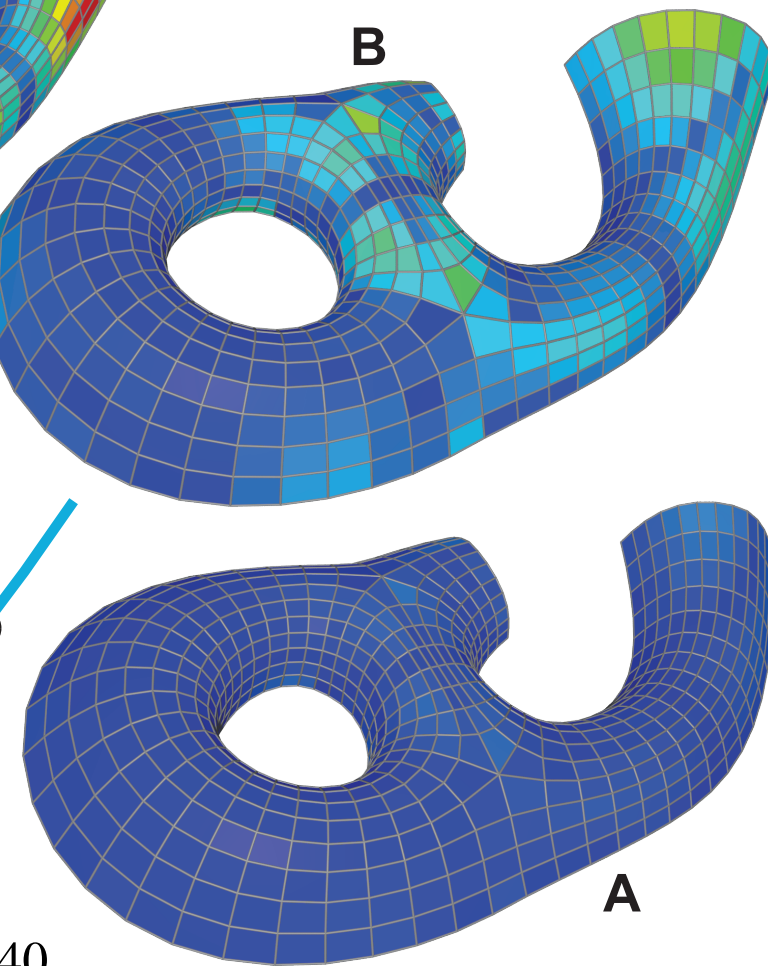
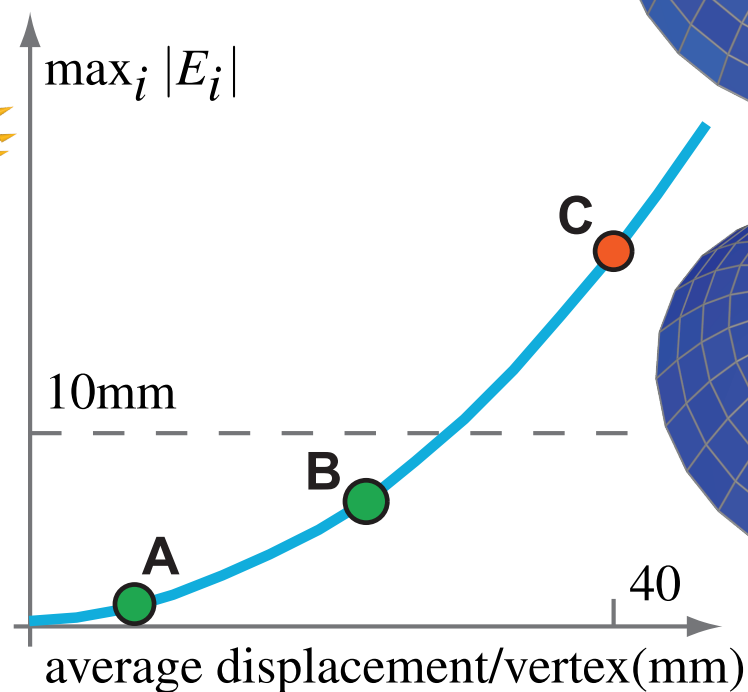
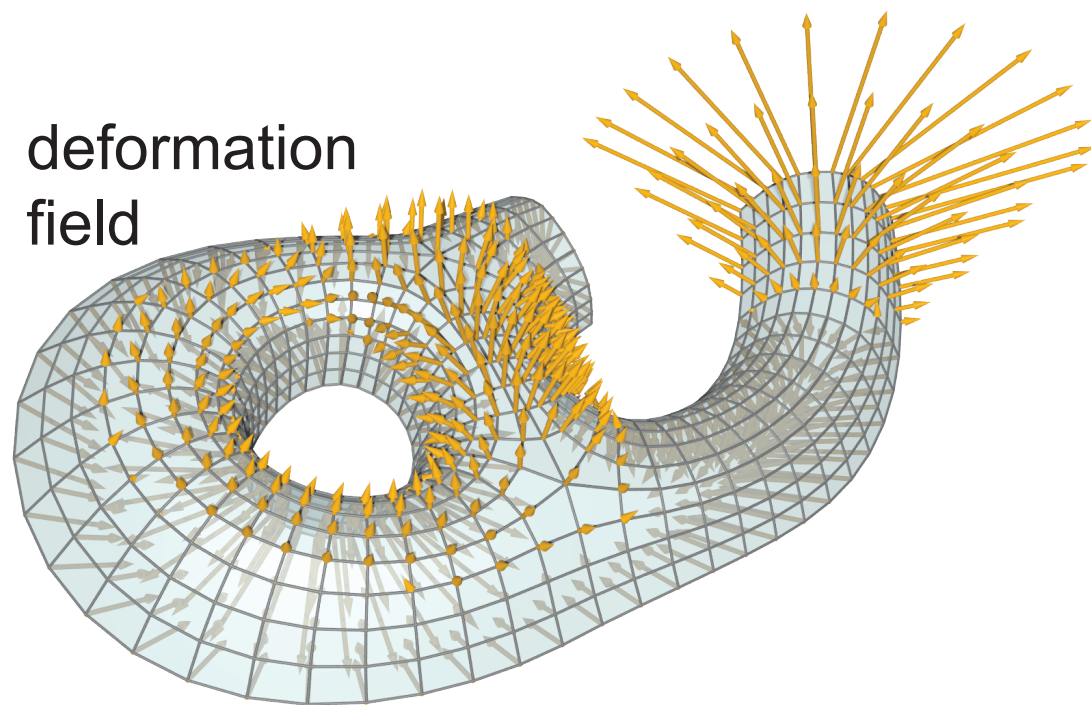
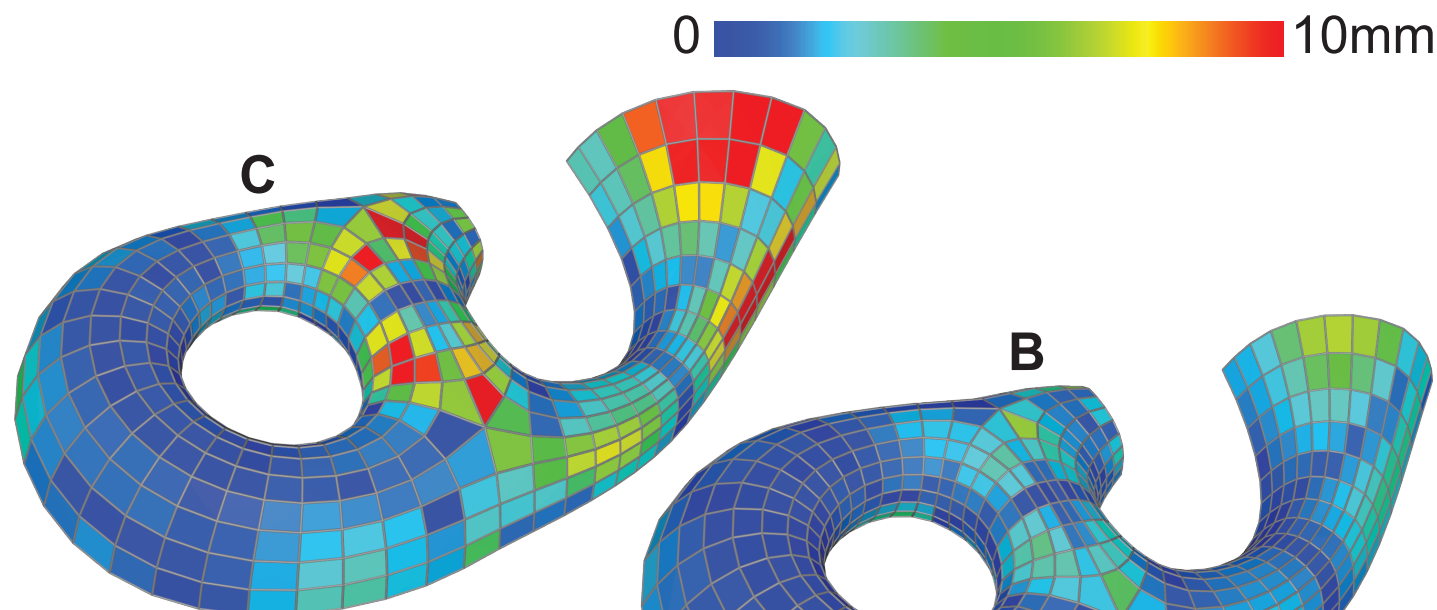
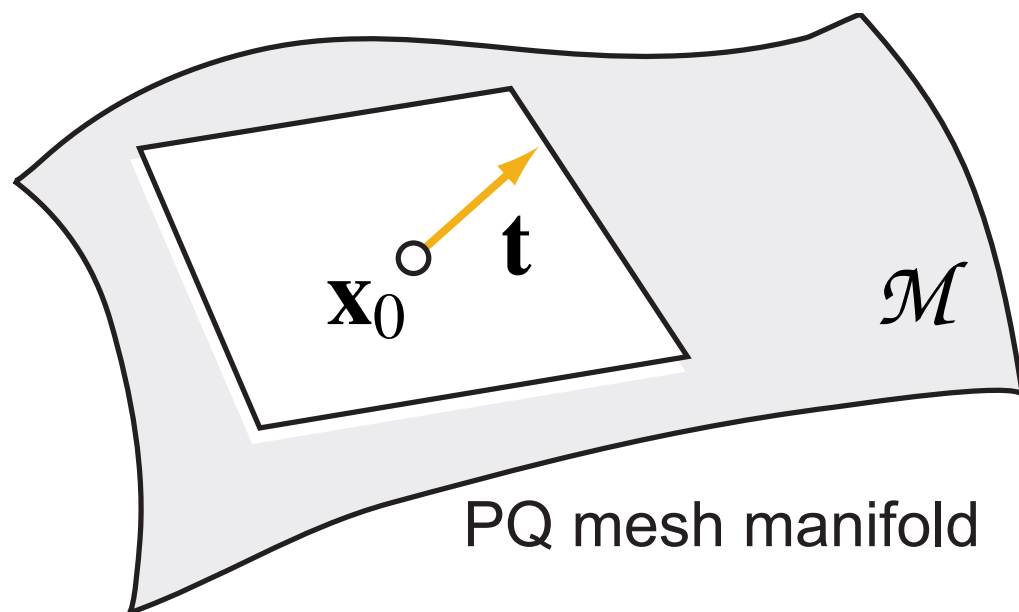


0  10mm

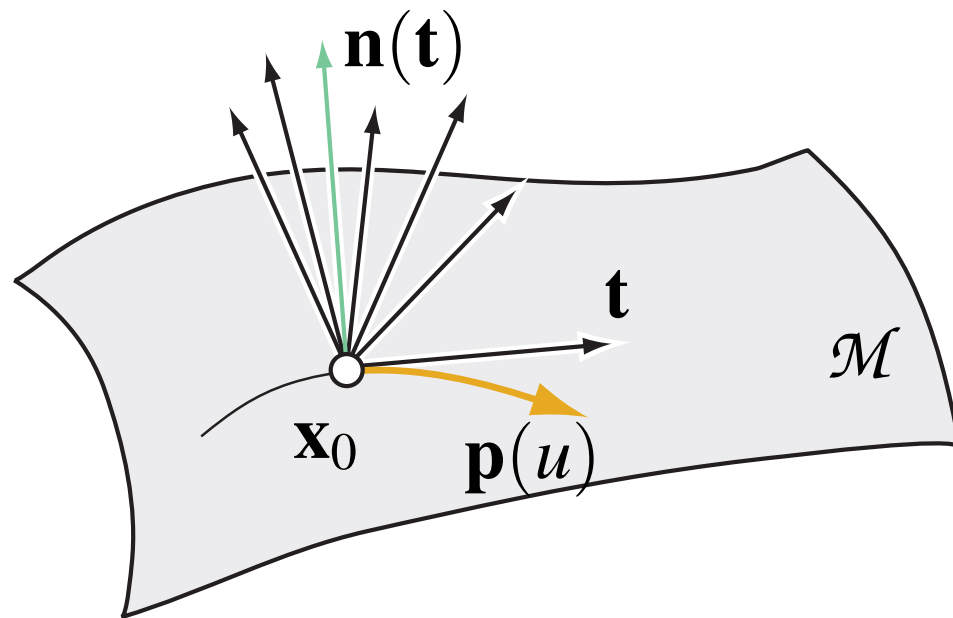




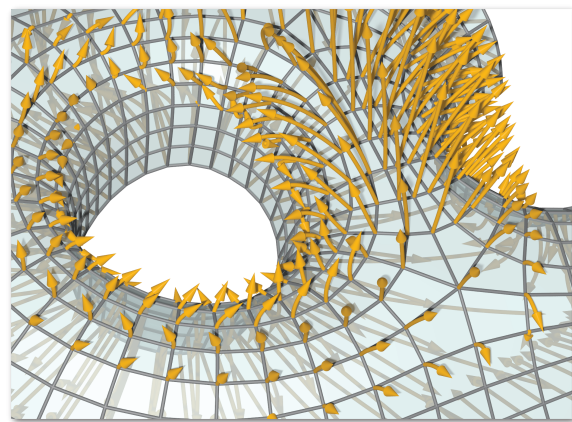
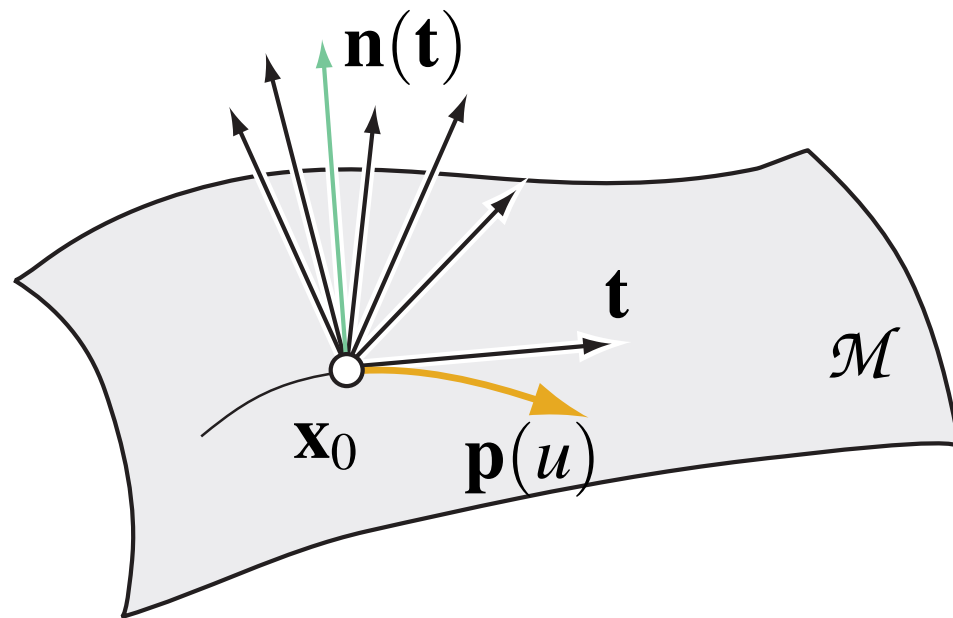
# Walking on the Tangent Space



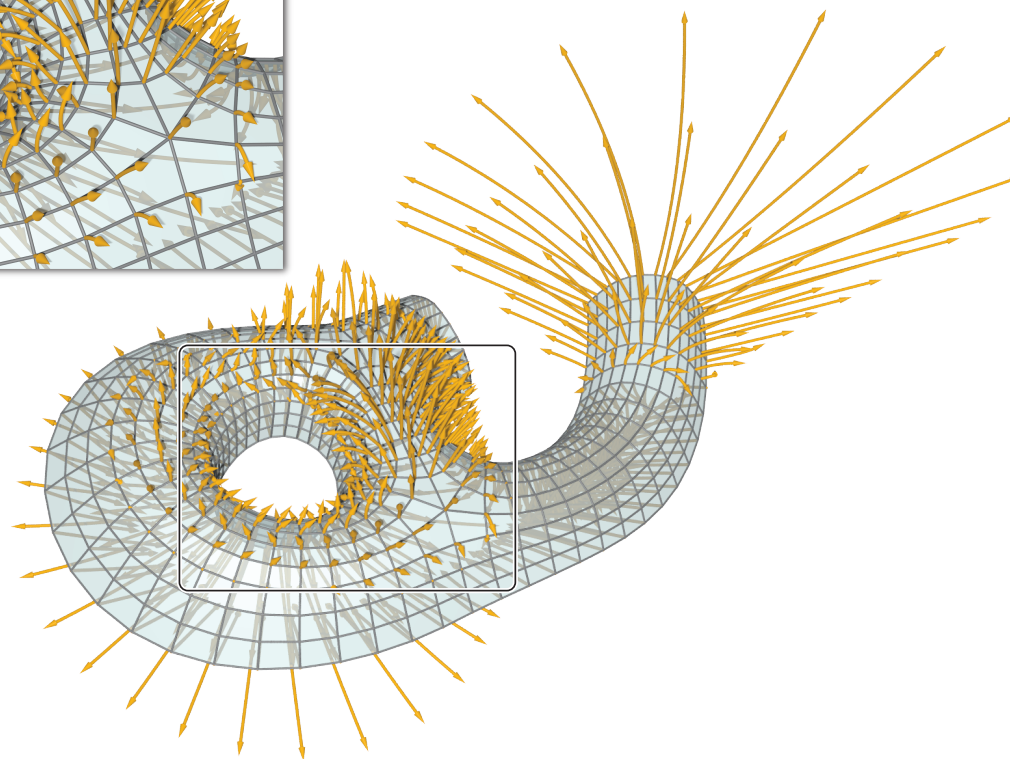
# Walking on the Osculant



# Walking on the Osculant

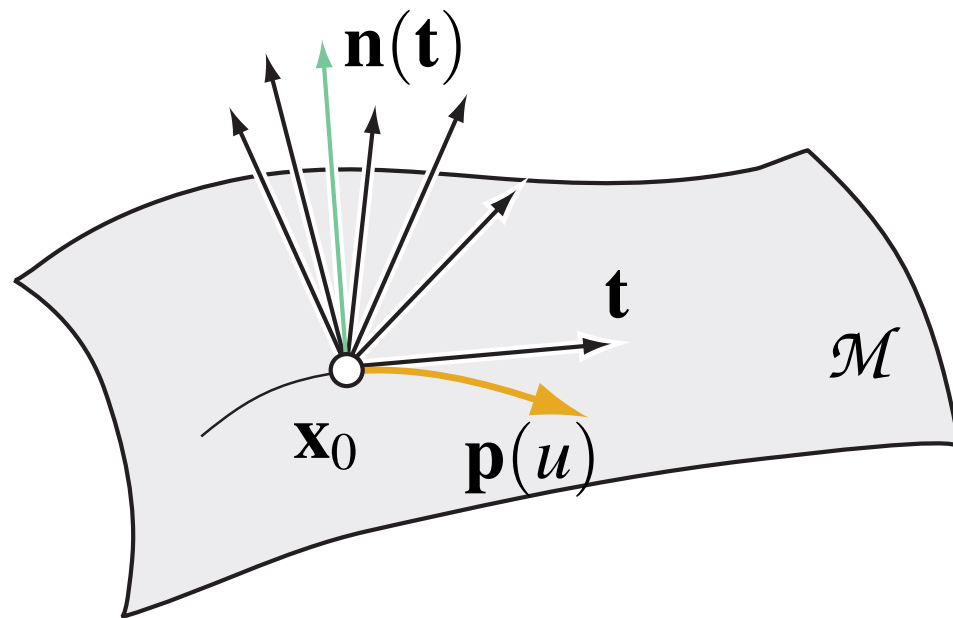


osculant  
deformation  
field

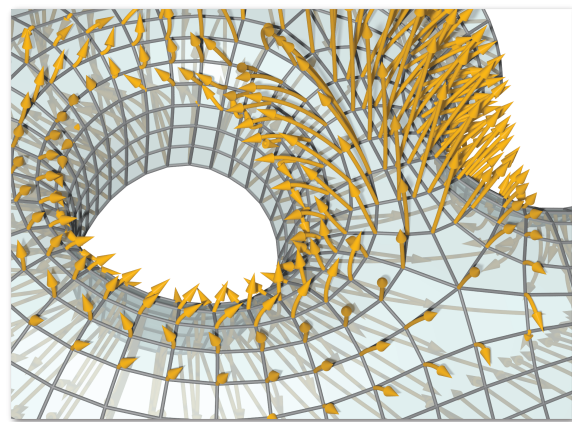




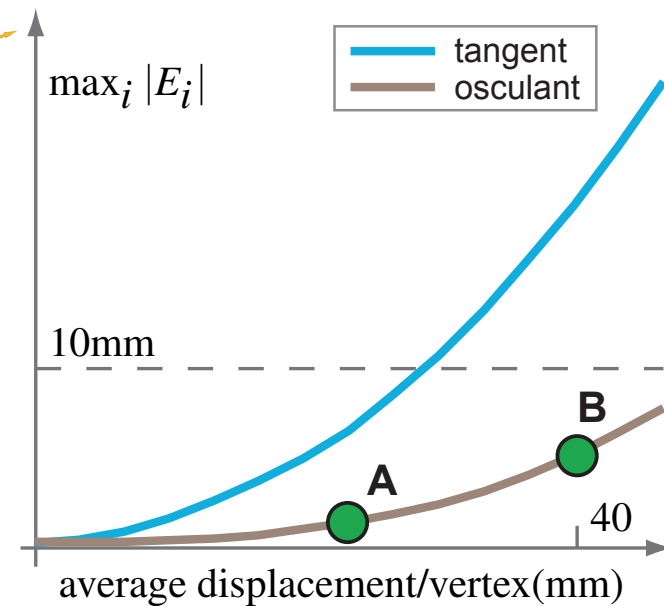
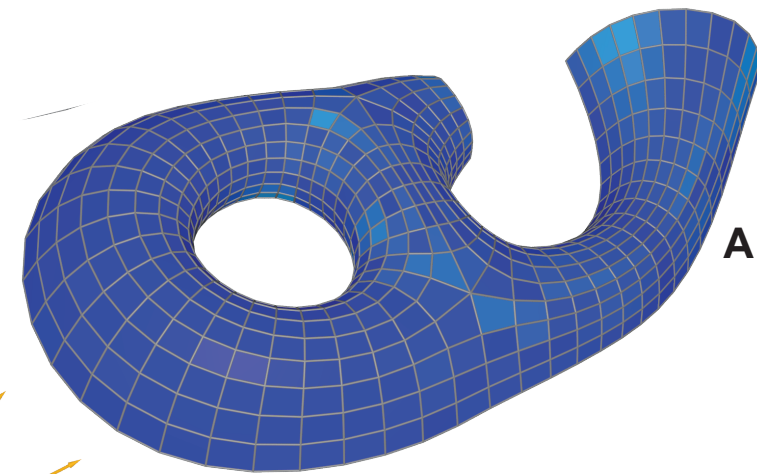
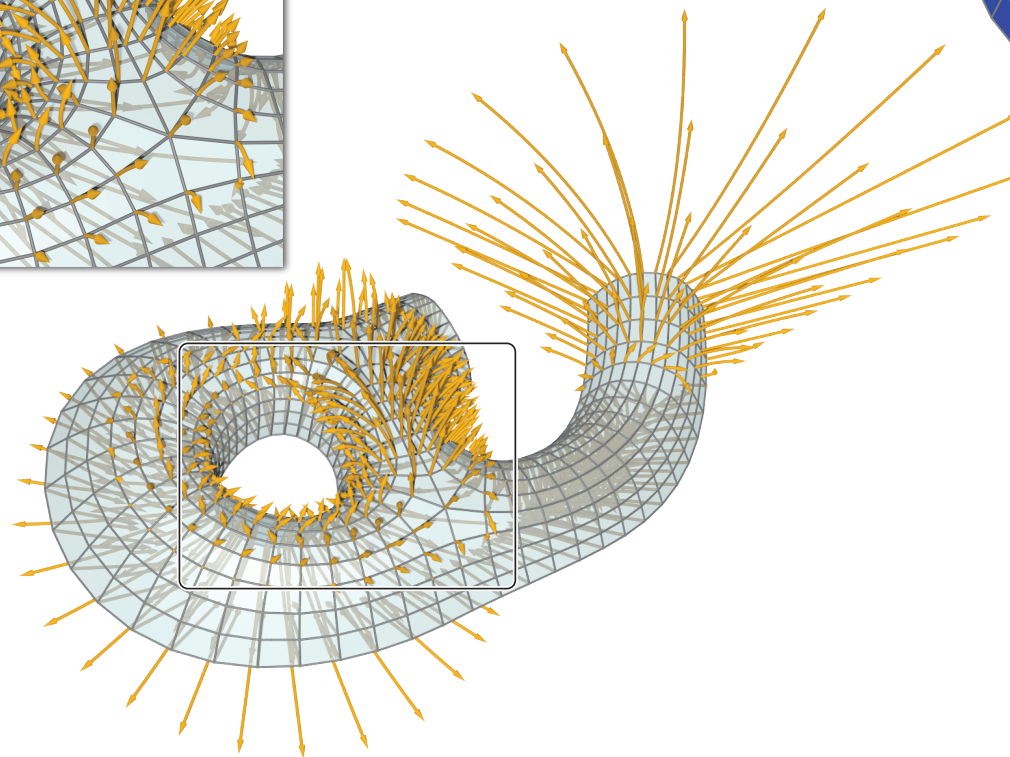
# Walking on the Osculant



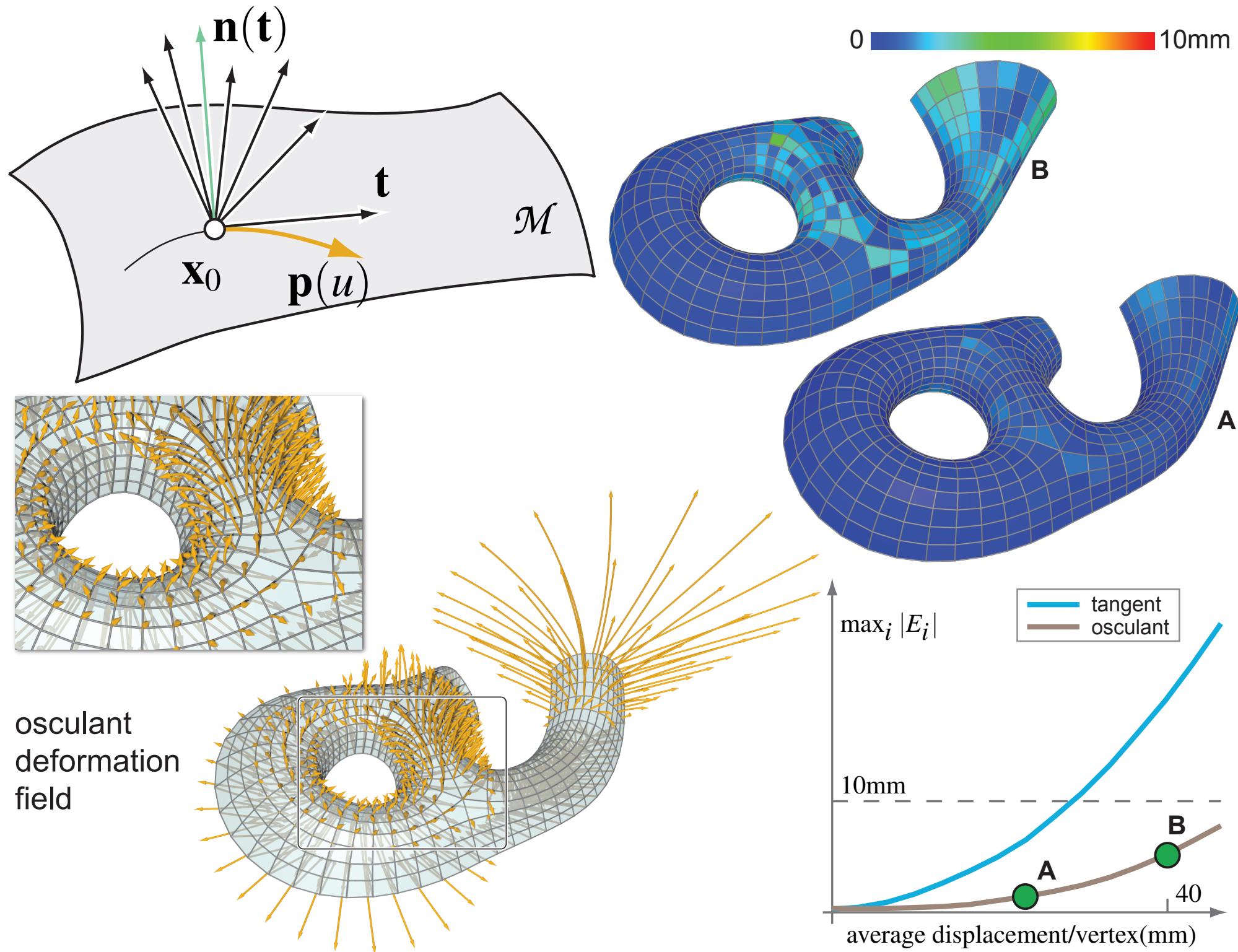
0  10mm



osculant  
deformation  
field



# Walking on the Osculant



## Flat Circular Mesh Exploration