Niloy J. Mitra

Symmetry Detection and Symmetrization
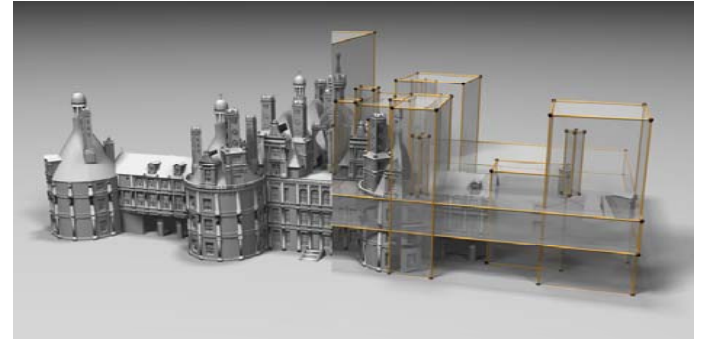
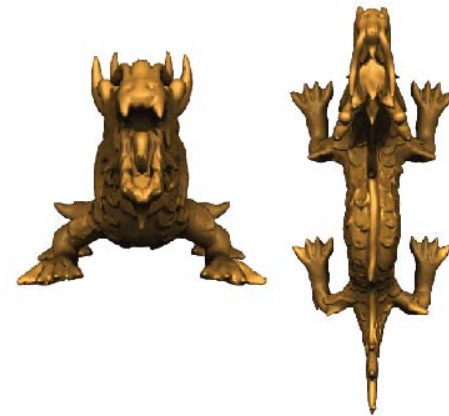*joint work with M. Pauly and L. Guibas*

TU WIEN

Goals

Detect symmetries in 3D geometry
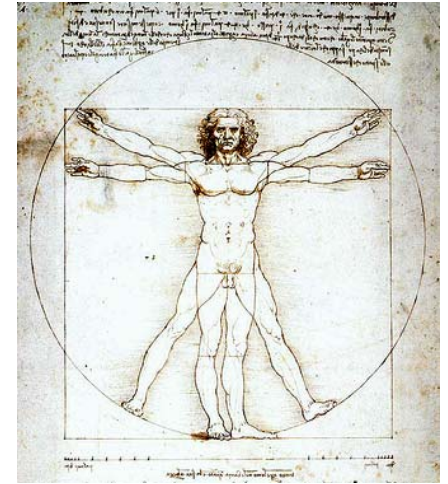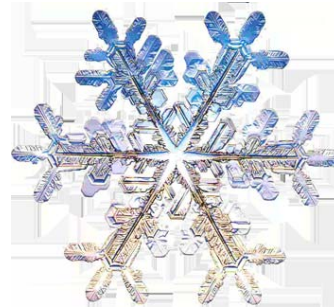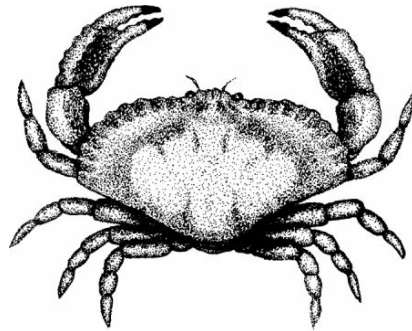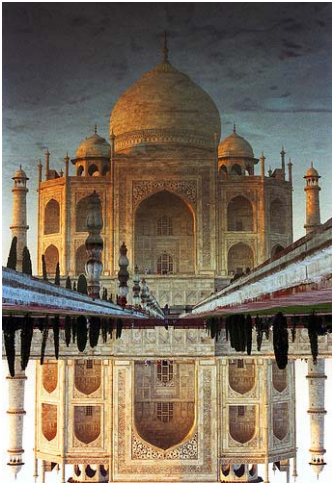

Photo: H. Hoppe

Symmetrize 3D shapes

Symmetry Detection

Symmetry

"Symmetry is a complexity-reducing concept [...]; seek it everywhere."
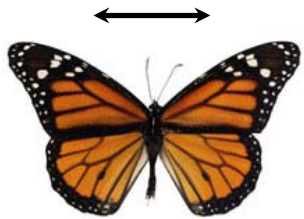
\- Alan J. Perlis

"Females of several species, including [...] humans, prefer symmetrical males."
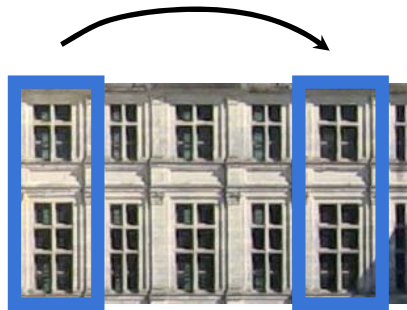
\- Chris Evans

What is Symmetry?

Invariance under a class of transformations
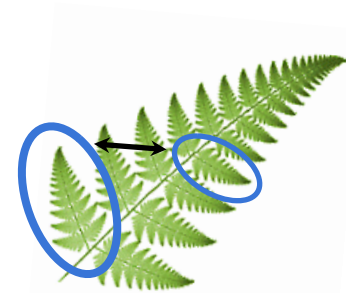

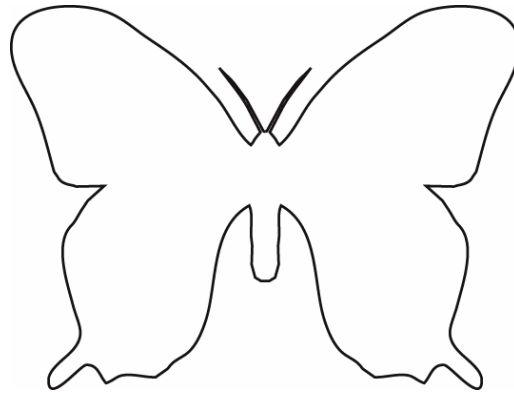Reflection   Translation   Rotation   Reflection + Translation + Rotation + Scaling

- global vs. partial
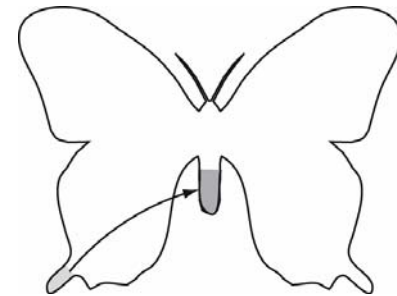- exact vs. approximative

Symmetry Detection

Given
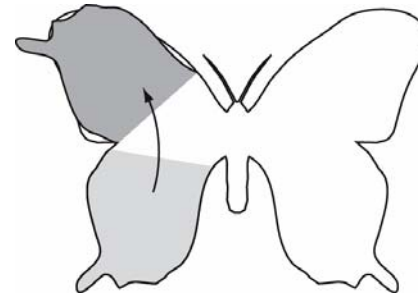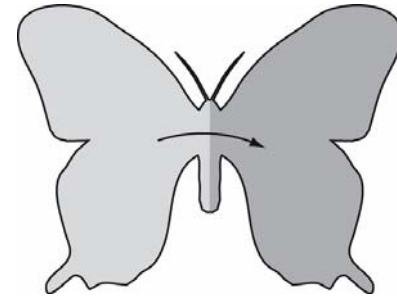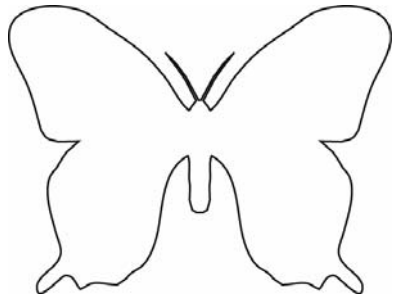
Object/shape (represented as point cloud, mesh, …)

Goal

Identify and extract *similar* (symmetric) patches of different *size* across different *resolutions*
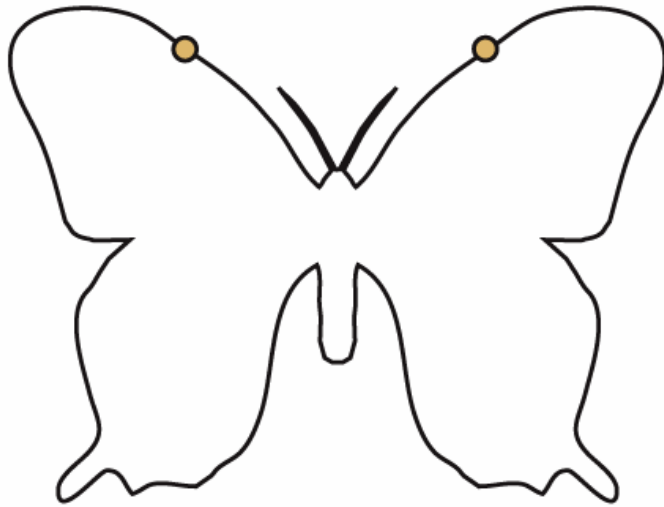
Partial Symmetry

Transform Types:

- Reflection

- Rotation + Translation

- Uniform Scaling

Reflective Symmetry

Reflective Symmetry : A Pair Votes
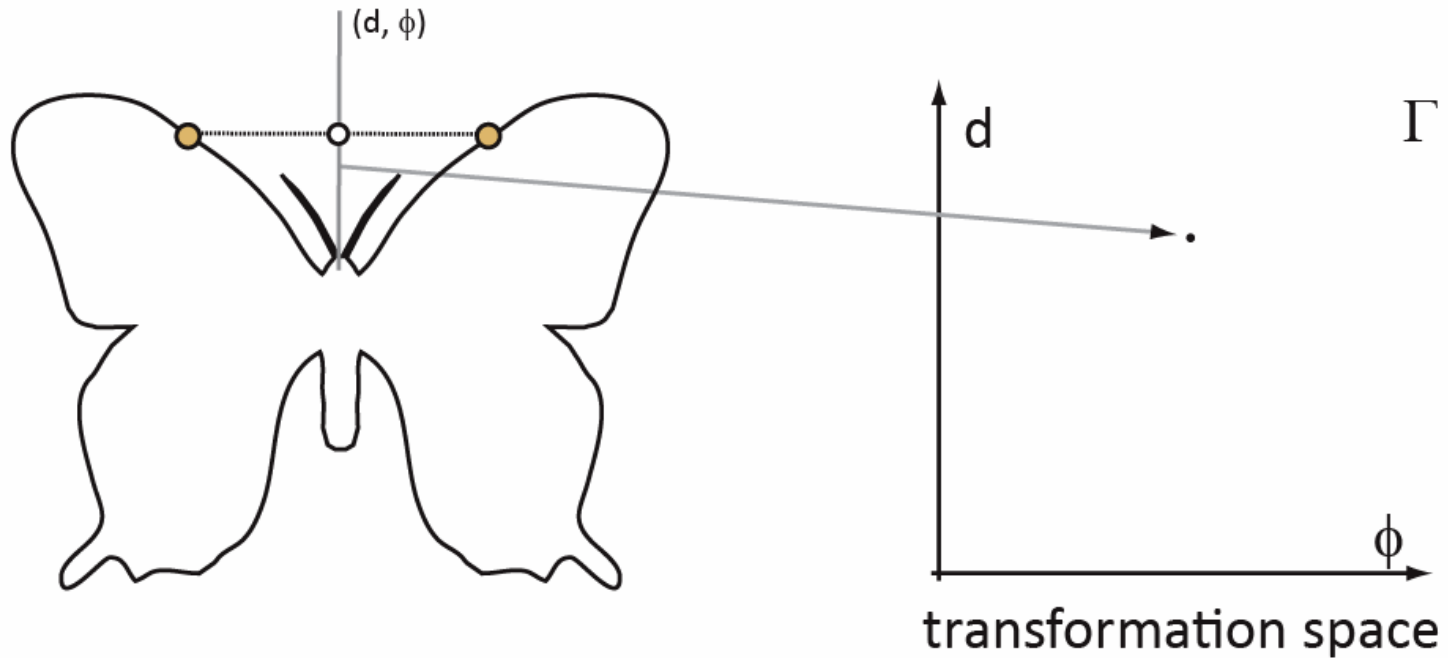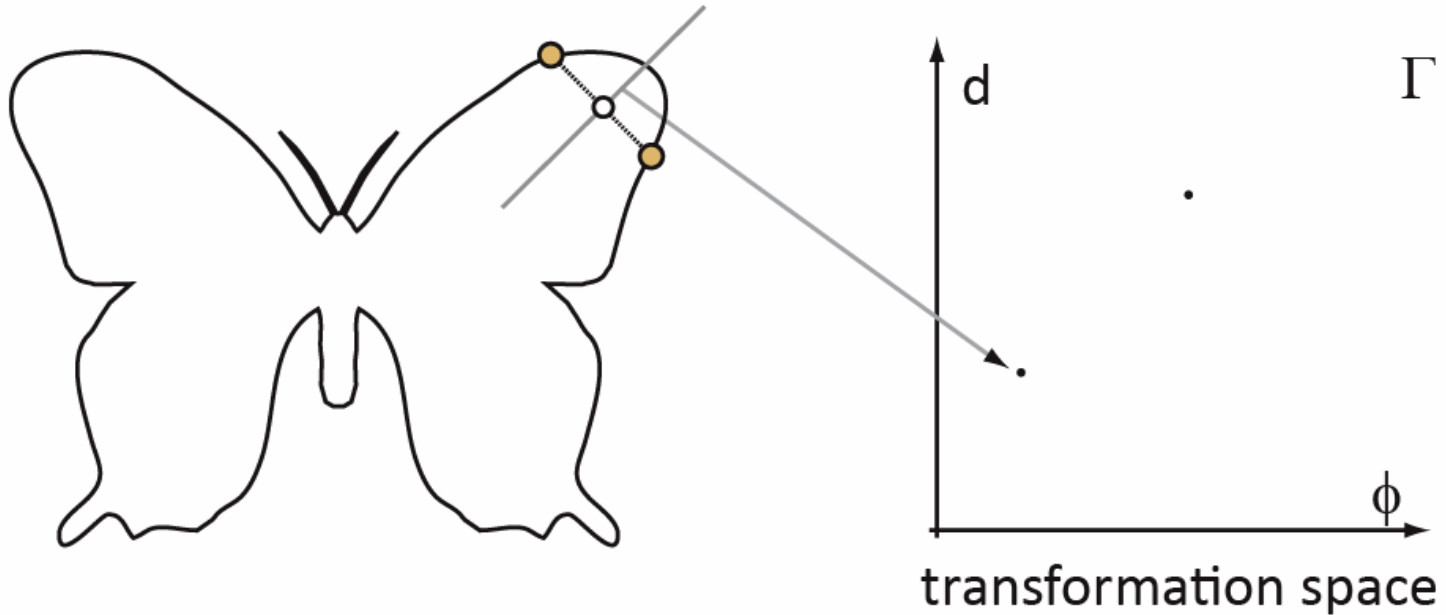
transformation space
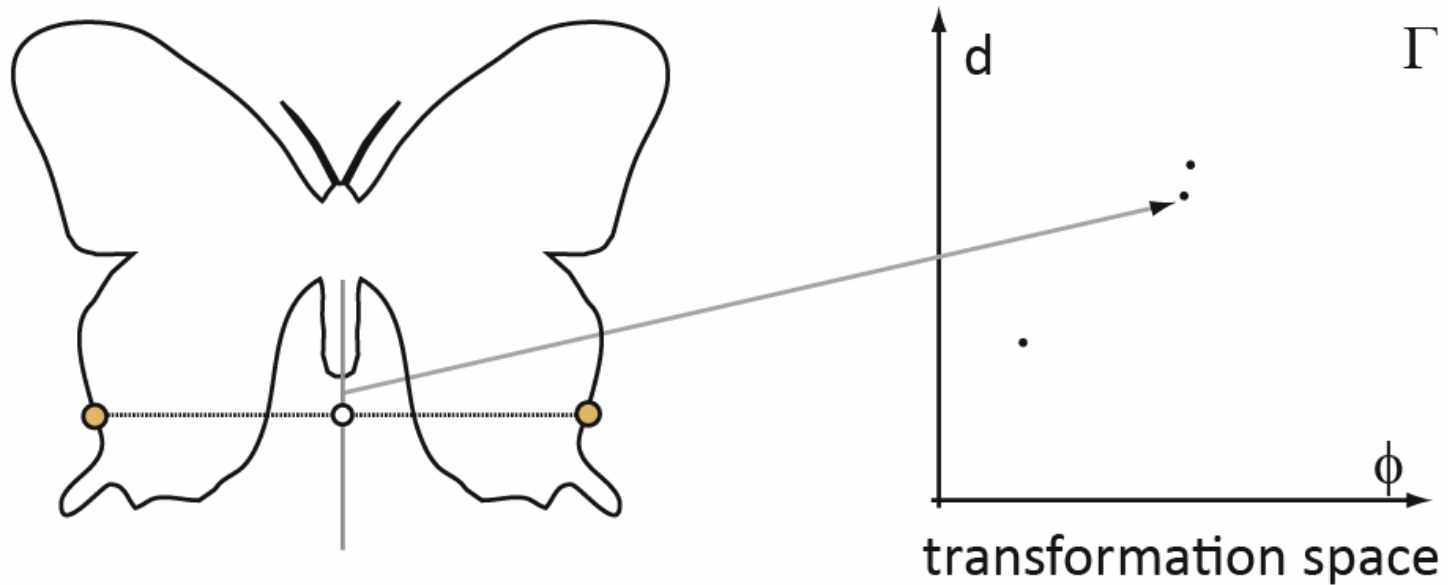
Reflective Symmetry : Voting Continues

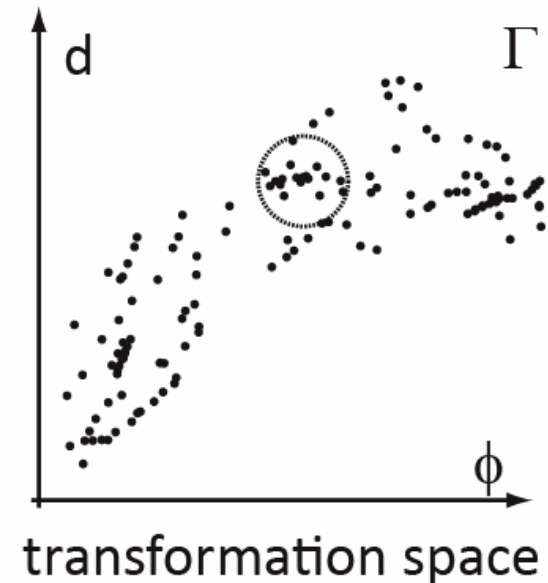# Reflective Symmetry : Voting Continues

Reflective Symmetry : Voting Continues
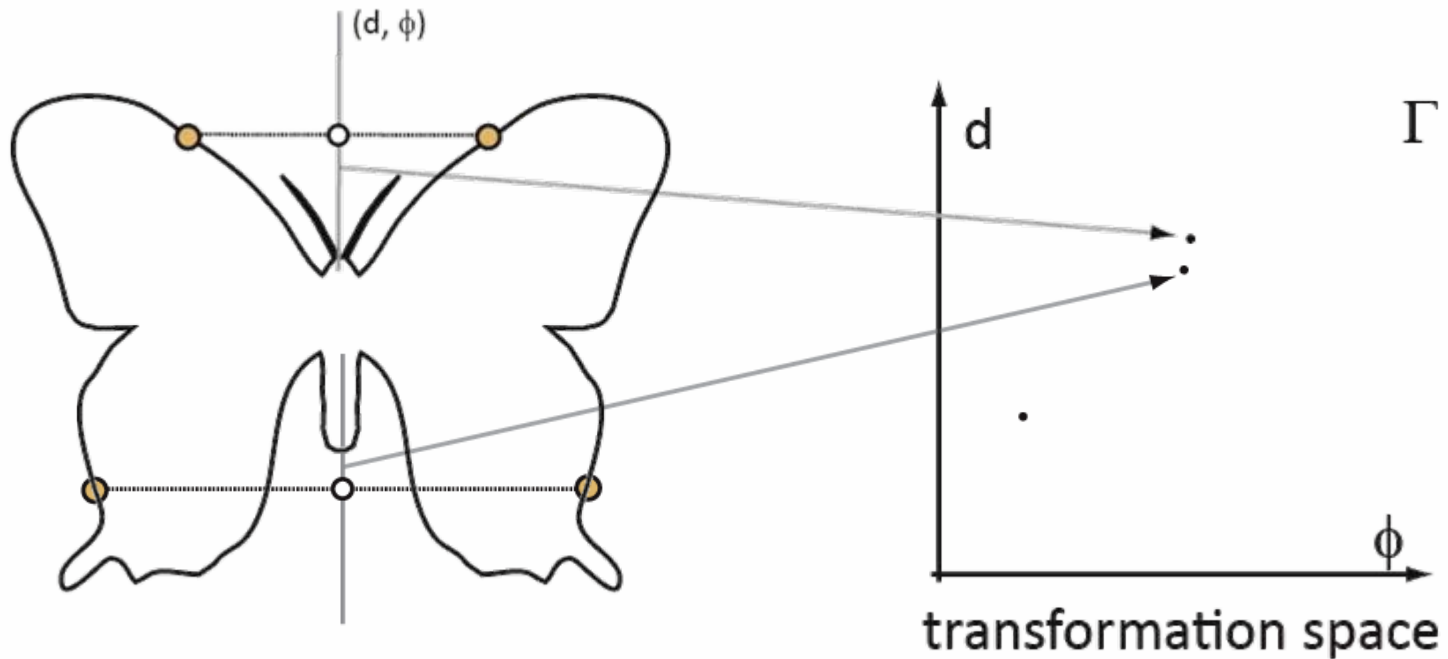
Reflective Symmetry : Largest Cluster
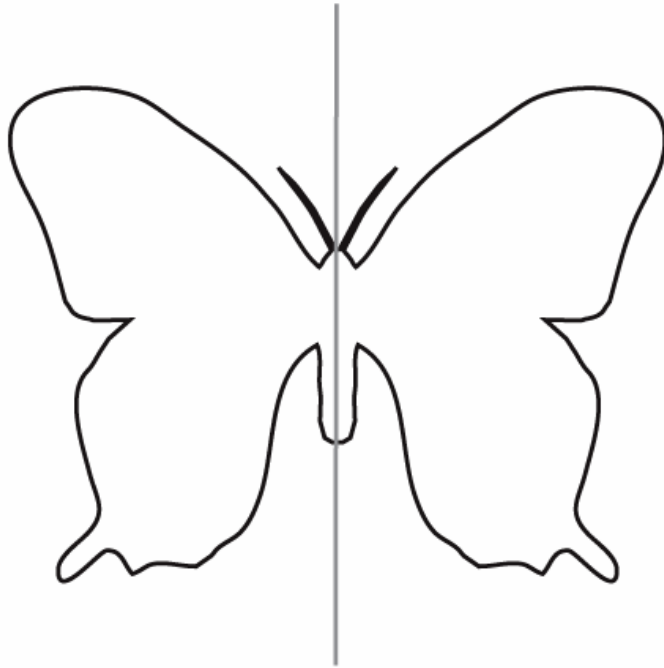
- Spread of cluster $\rightarrow$ approximation level

- Height of cluster $\rightarrow$ size of patch

Symmetry Detection

Accumulation of local evidence

local evidence for symmetry plane
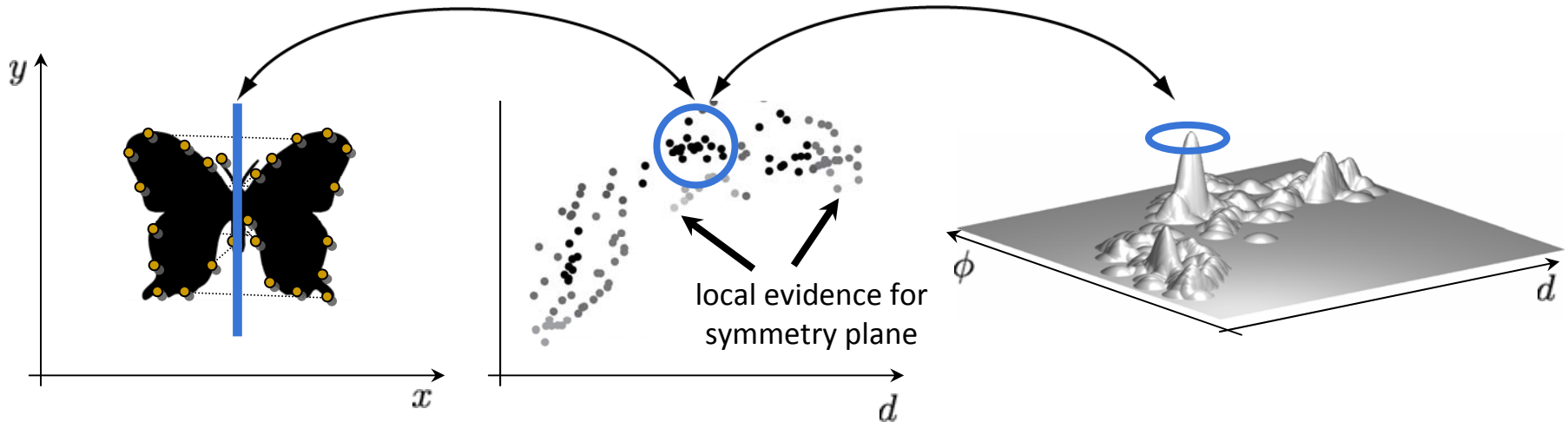
- clustering to extract symmetry transformation

- verification to extract symmetric patches

Pipeline

input model → sampling → sample set → analysis → signatures Ω → pairing → transformations Γ → clustering → density → patching → surface patches

Pruning: Local Signatures

- *Local* signature $\rightarrow$ invariant under transforms
- Signatures disagree $\rightarrow$ points don't correspond

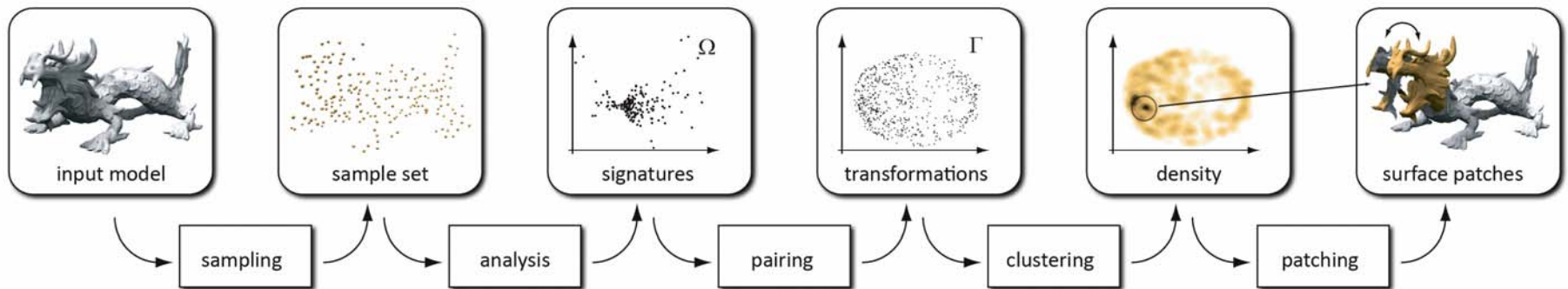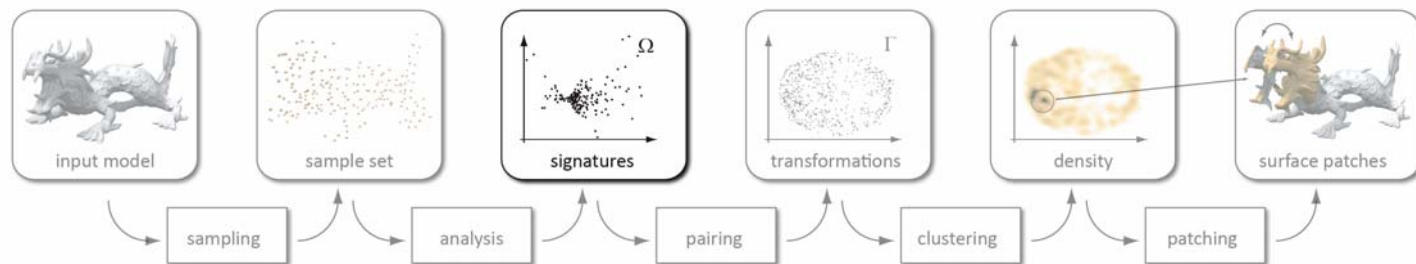b

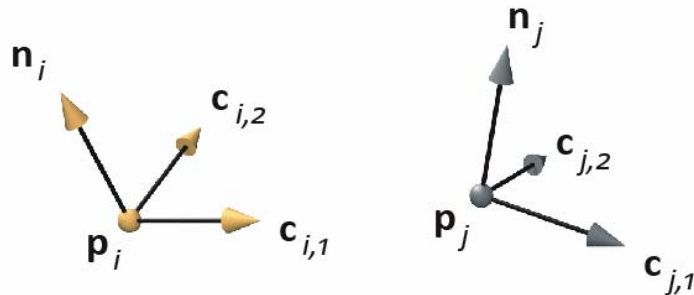$(0, 1/a)$

a

$(1/(a+b), 1/a))$
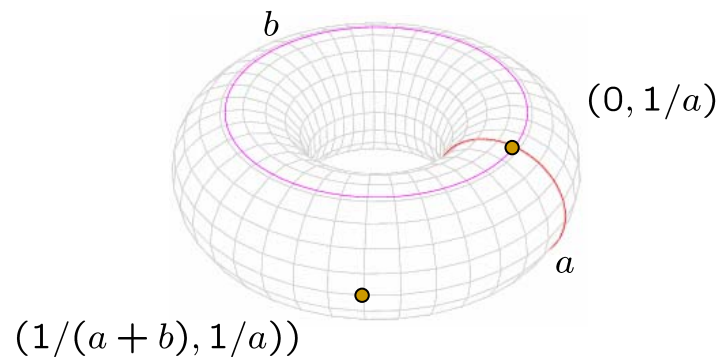
Use (κ_1, κ_2) for curvature based pruning

input model — sample set — signatures Ω — transformations Γ — density — surface patches
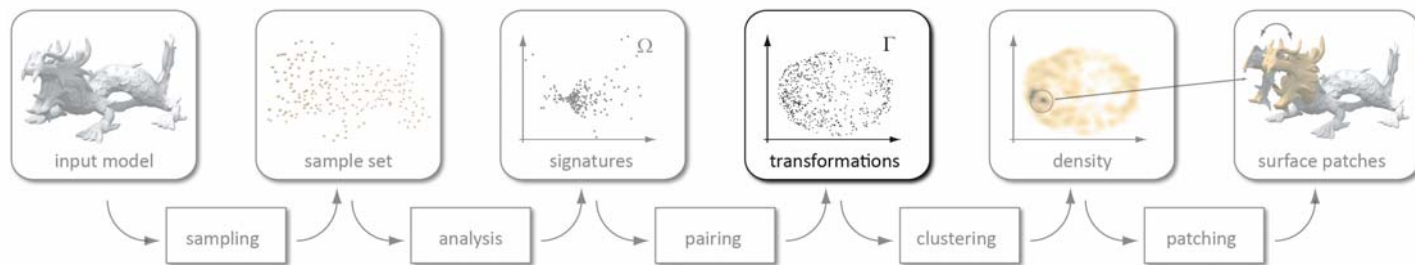
sampling — analysis — pairing — clustering — patching

Transformations

- Reflection $\rightarrow$ point-pairs
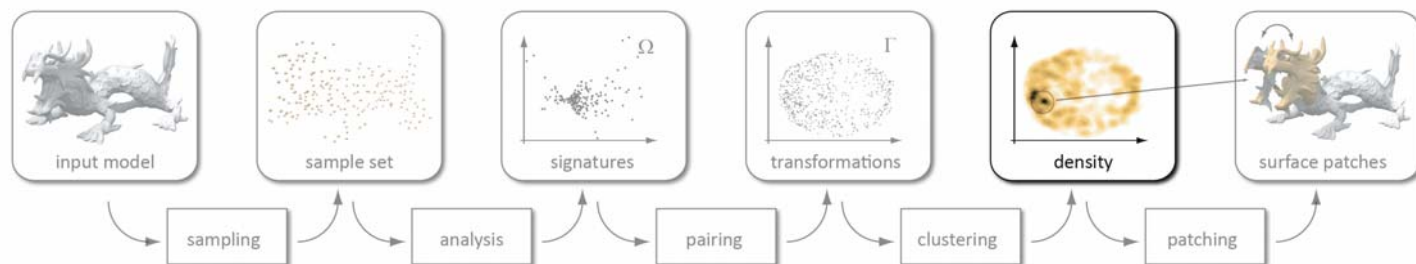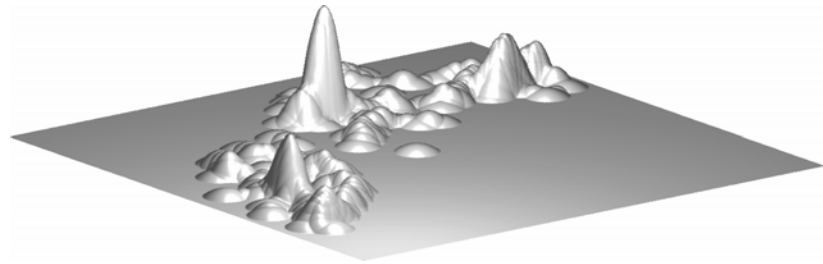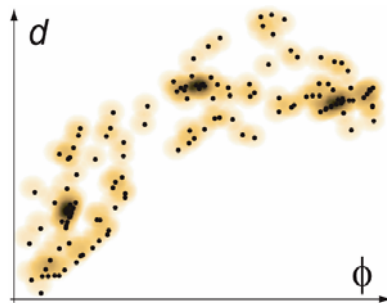- Rigid transform $\rightarrow$ more information
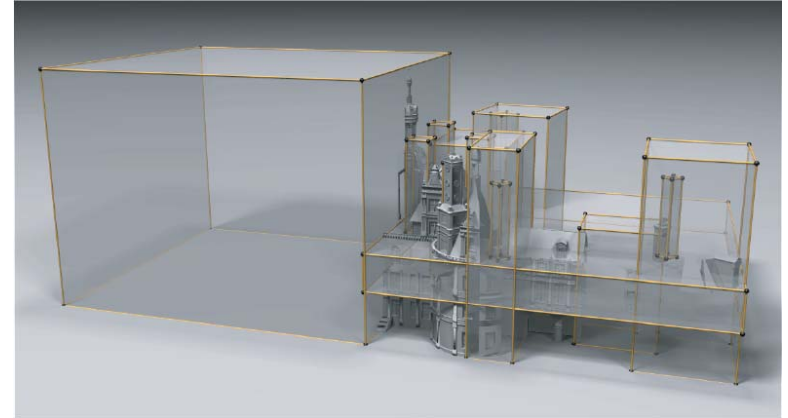

Robust estimation of principal curvature frames
[Cohen-Steiner et al. `03]

Mean-Shift Clustering

Kernel:

- Type → radially symmetric hat
- Radius

Compression: Chambord

Compression: Chambord

52%

14%
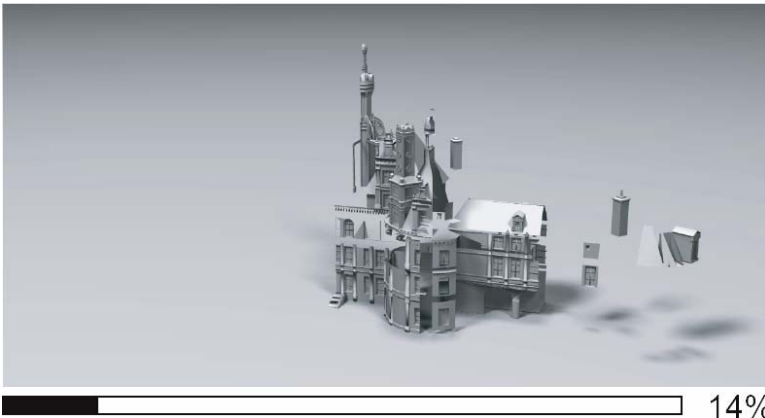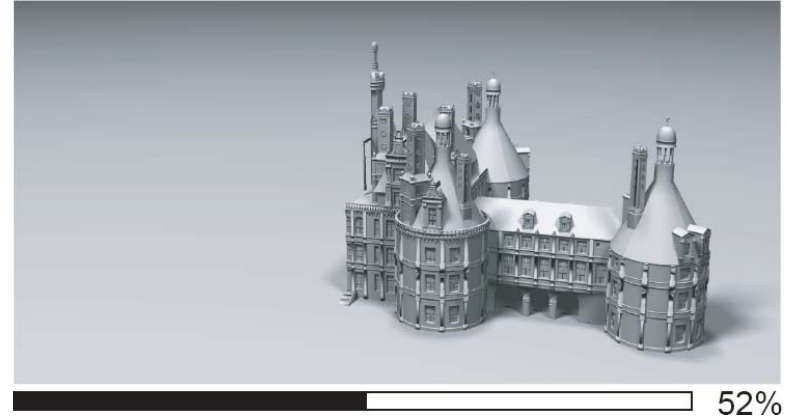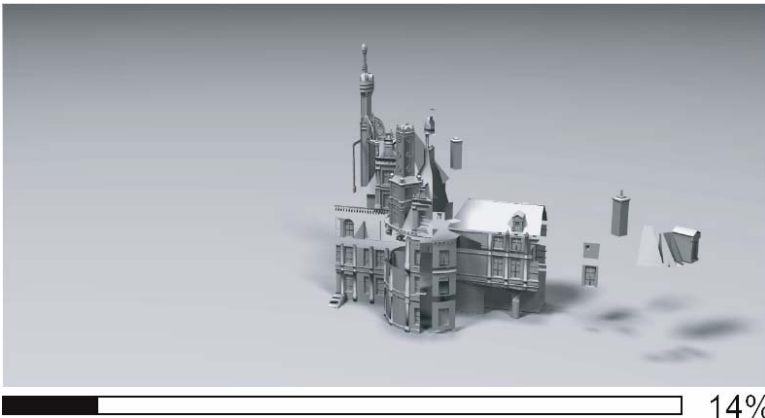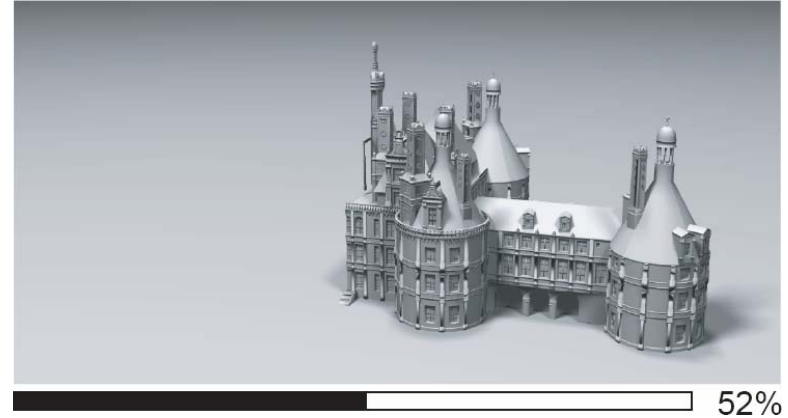
Approximate Symmetry: Dragon

detected symmetries

-0.02   0.02
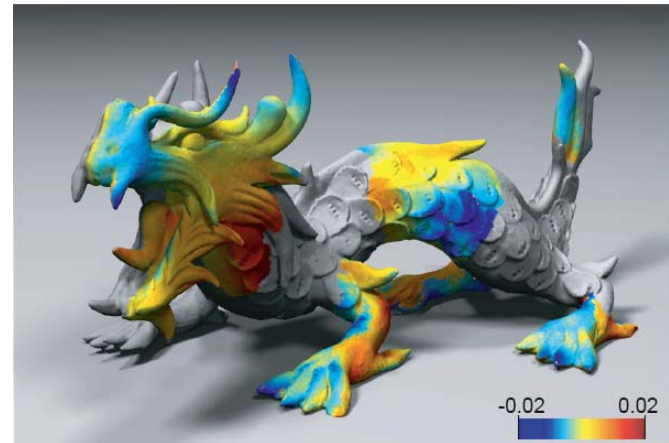
correction field

UNITS: fraction of bounding box diagonal

Symmetrization

Symmetrization

Goal: Symmetrize 3D geometry
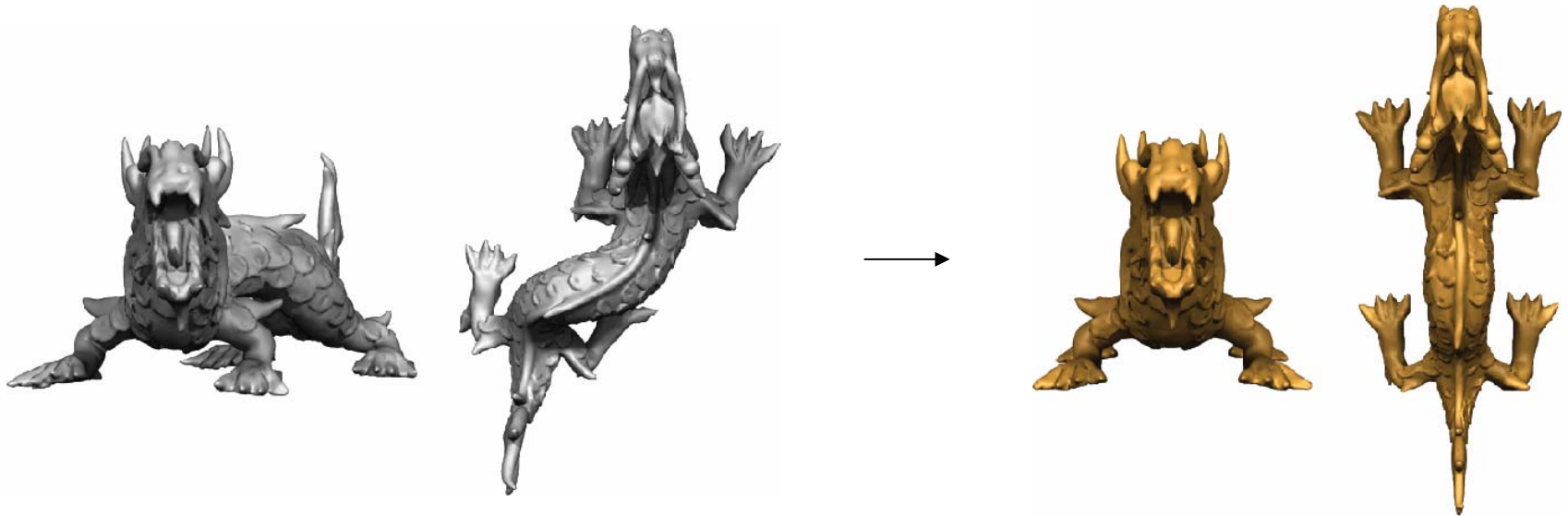
Approach: Minimally *deform* the model in the *spatial domain* by *optimizing* the distribution in *transformation space*

Optimal Displacements

Goal: Minimally displace two points to make them symmetric with respect to a *given transformation*

$$\mathbf{p}_i$$

$$\mathbf{p}_j$$

Optimal Displacements

Goal: Minimally displace two points to make them symmetric with respect to a ***given transformation***

Optimal Displacements

Goal: Minimally displace two points to make them symmetric with respect to a ***given transformation***
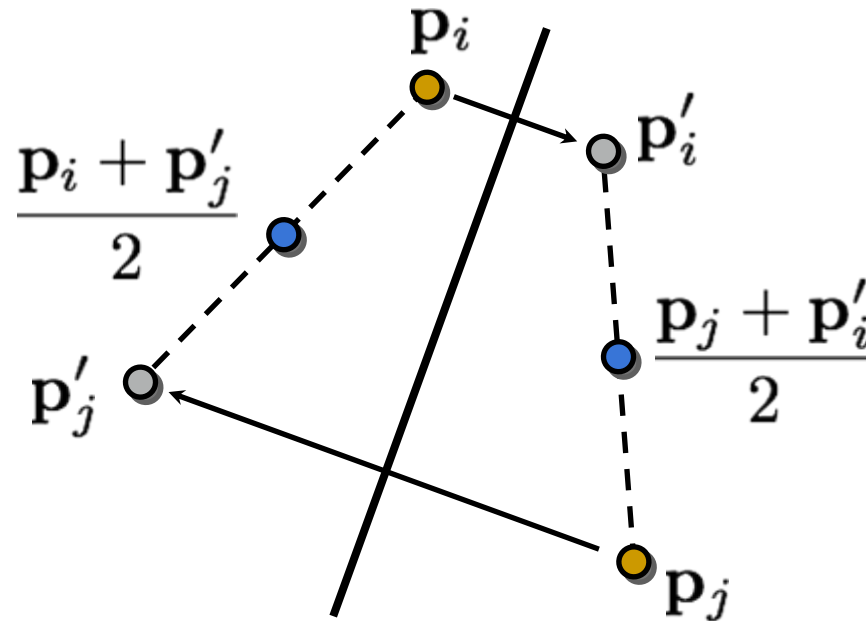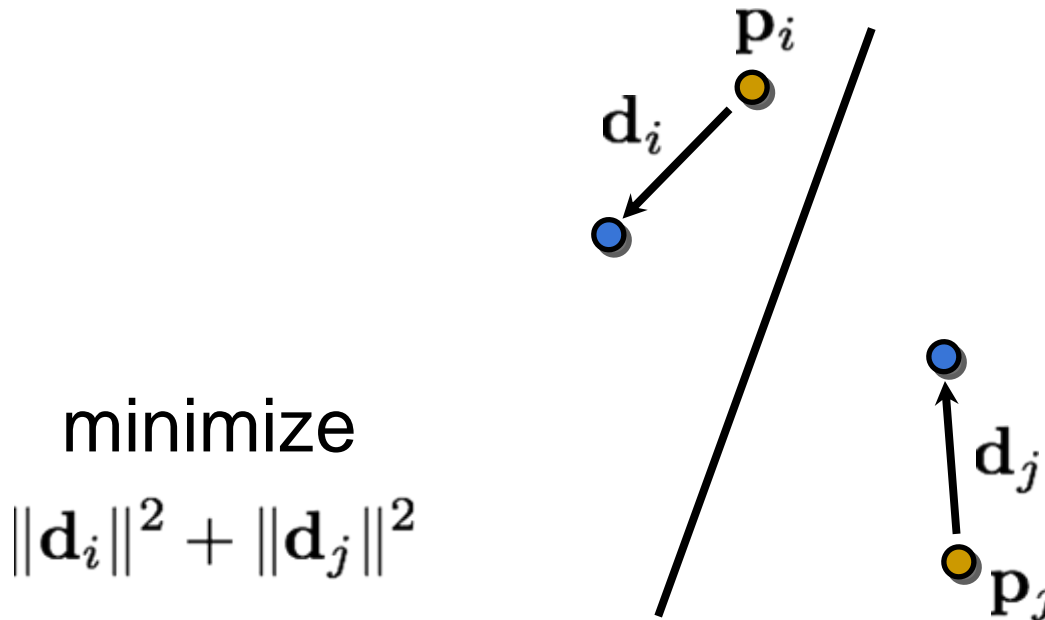
minimize

$$\|\mathbf{d}_i\|^2 + \|\mathbf{d}_j\|^2$$

Optimal Transformation

Goal: Find *optimal transformation* and *minimal displacements* for a set of point-pairs
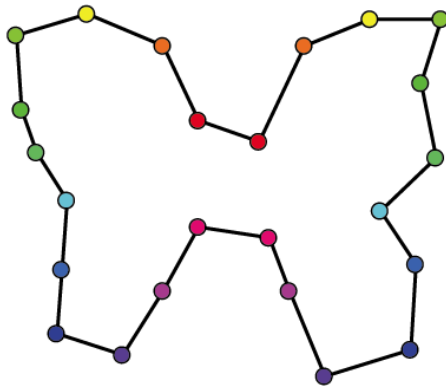
Optimal Transformation

Goal: Find *optimal transformation* and *minimal displacements* for a set of point-pairs

optimal transform

closed form solution

2D Example: Symmetry Detection

original

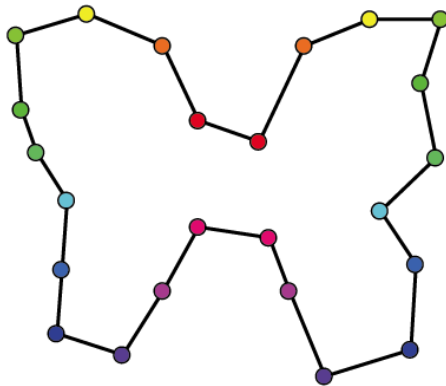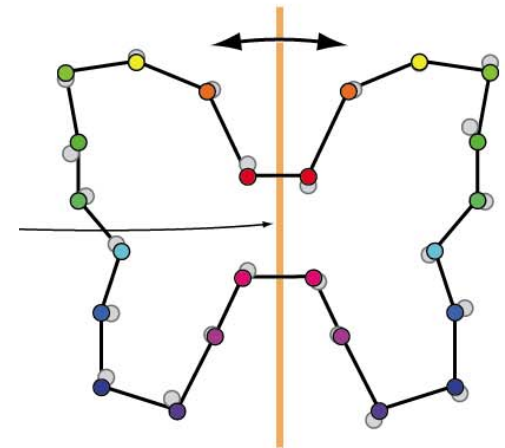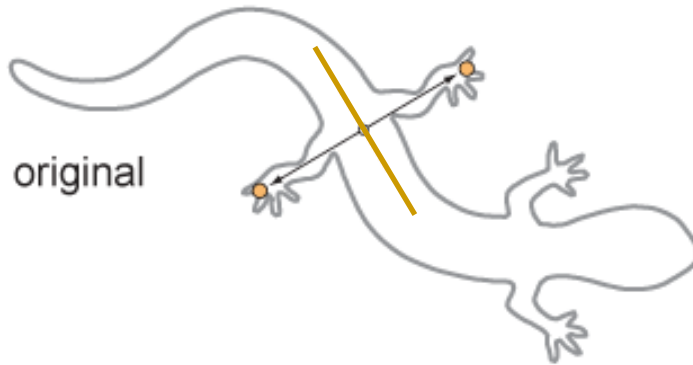# 2D Example: Symmetry Detection

original

2D Example: Symmetry Detection

2D Example: Symmetry Detection

object domain
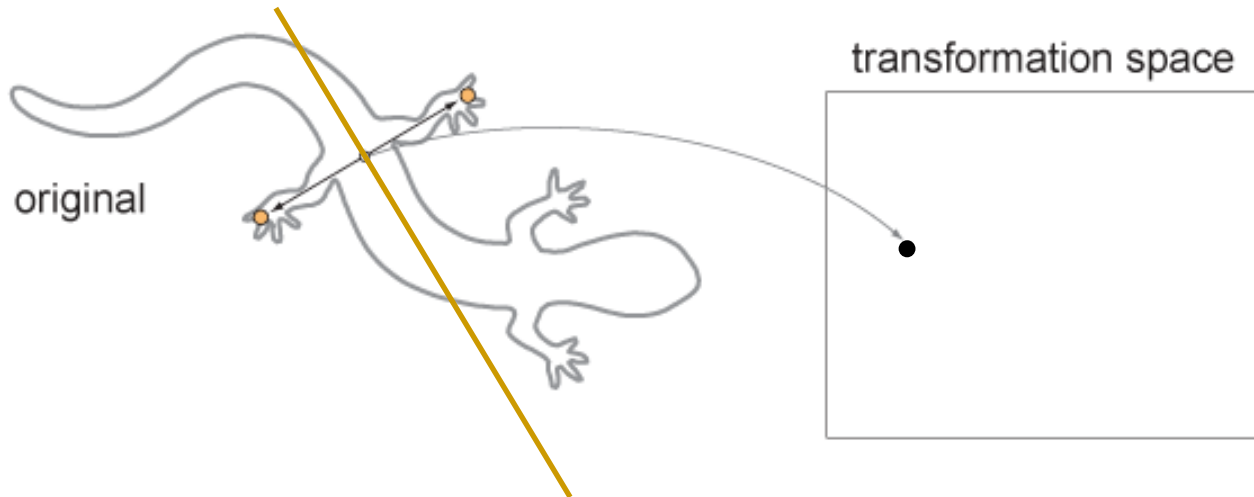
transform domain
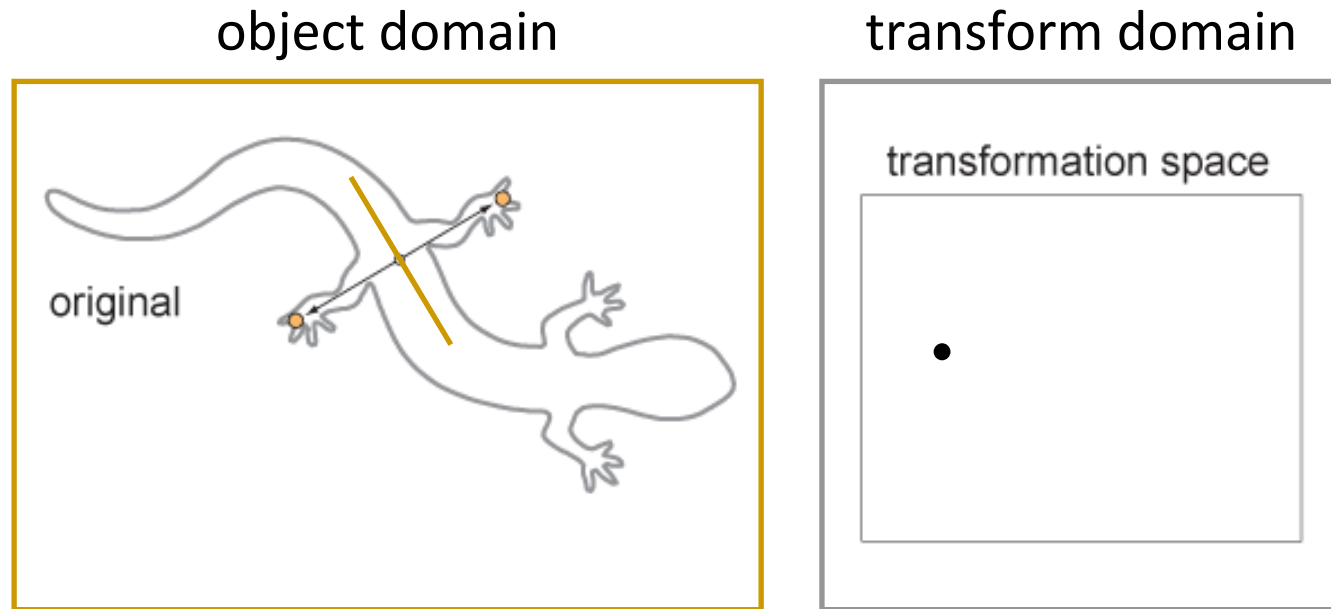
pair of points   →   point

2D Example: Another point-pair votes

original

transformation space
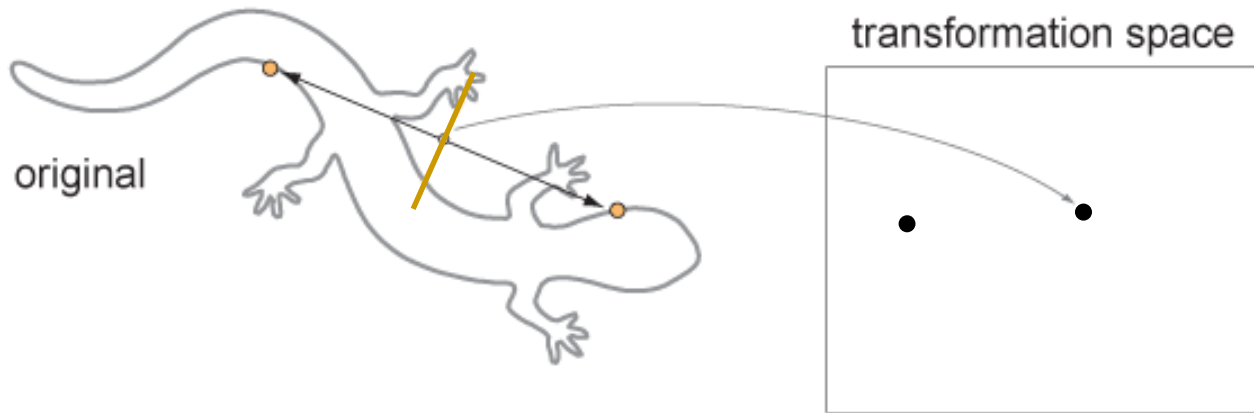
2D Example: Voting Continues

original

transformation space

pairs of sample points define reflective symmetry transform

2D Example: Density Plot

density plot → accumulation of symmetry evidence

2D Example: Density Peaks

density cluster $\rightarrow$ reflective symmetry

2D Example: Symmetry Detection

original

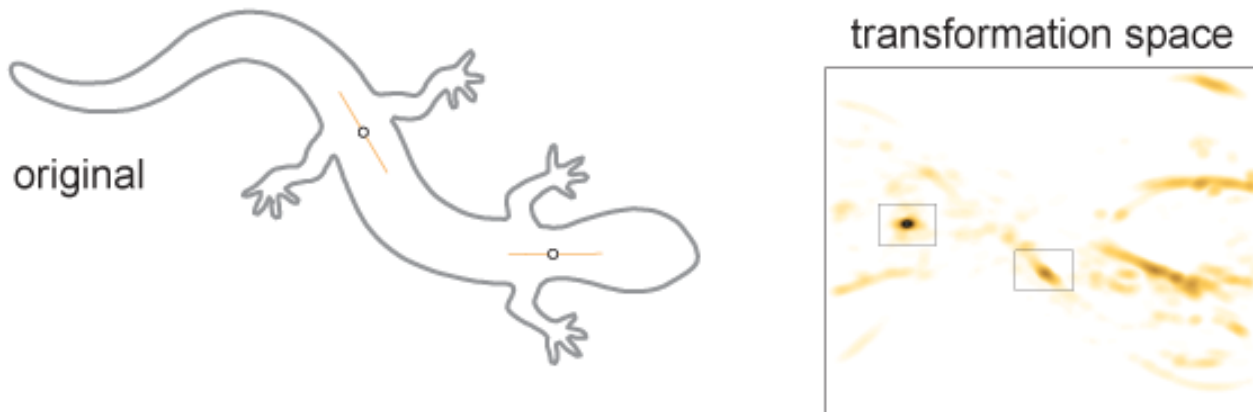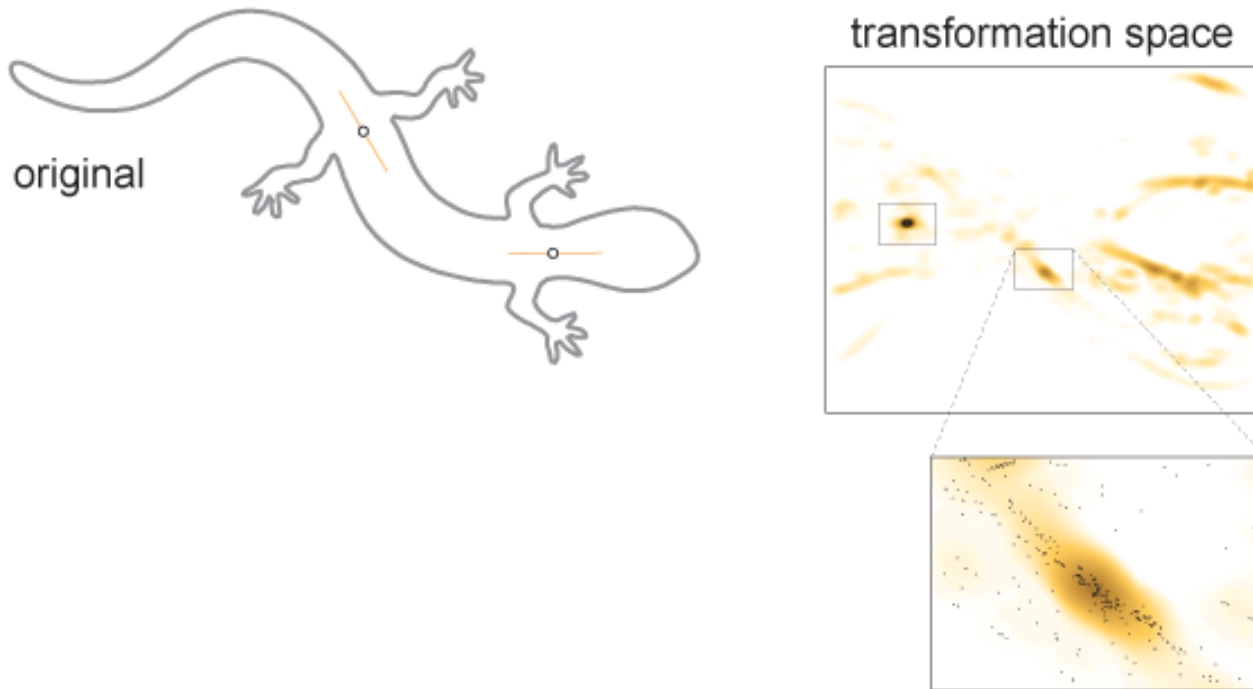transformation space

2D Example: Symmetry Detection

transformation space

original

a set of potential *corresponding point pairs* extracted

2D Example: Symmetry Detection

original

transformation space
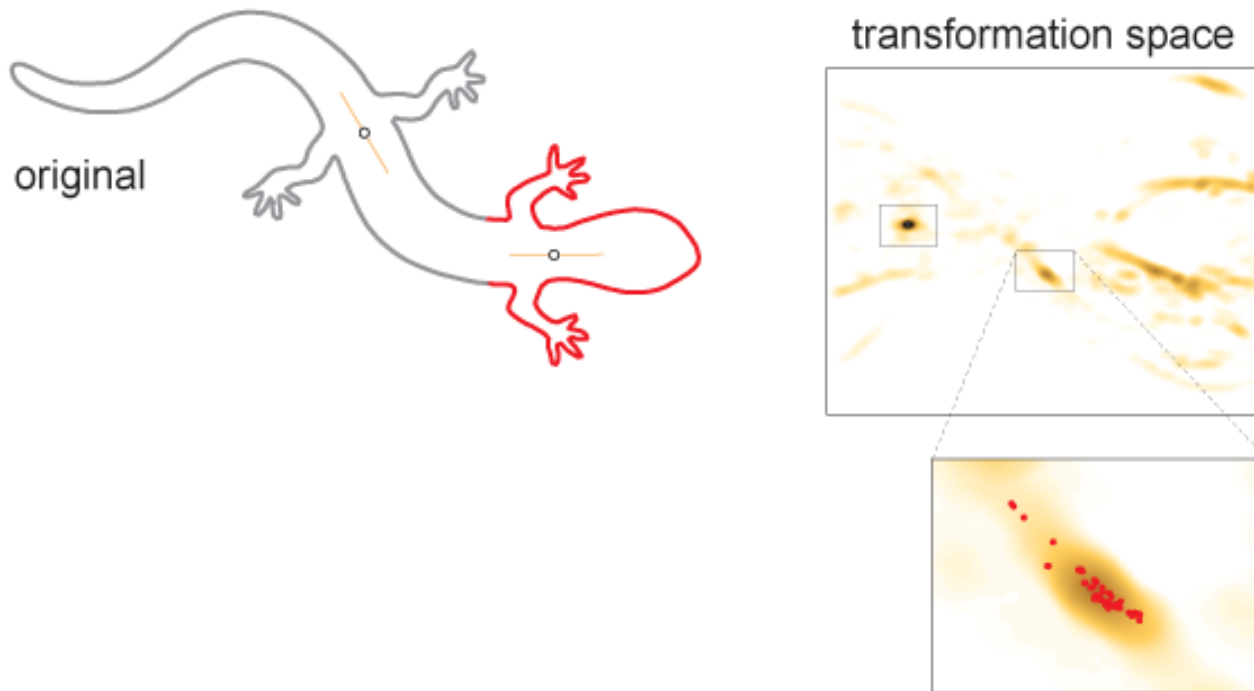
spread of such points → ***deviation*** from exact symmetry

2D Example: Local Symmetrization
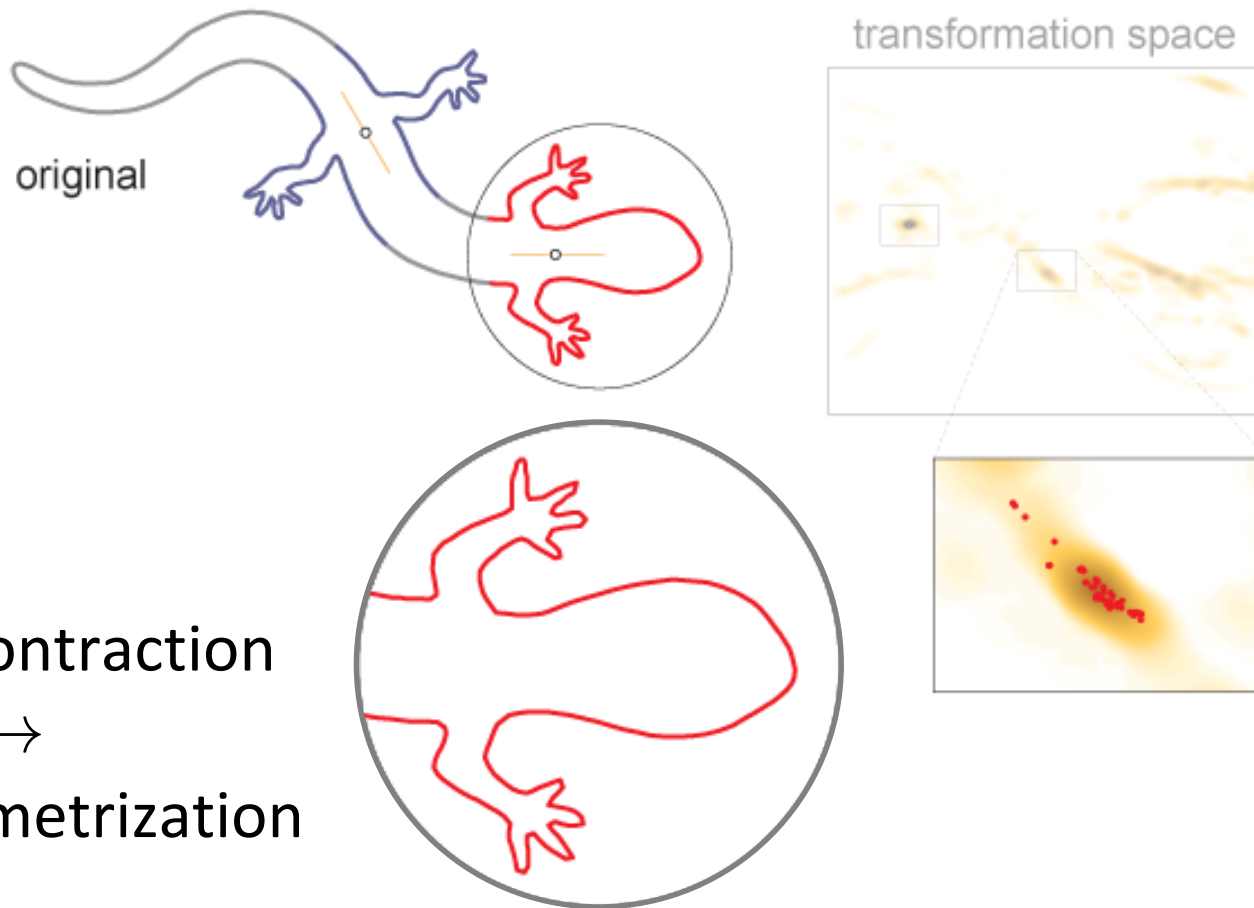
original

transformation space

cluster contraction

$\longrightarrow$

local symmetrization

2D Example: Local Symmetrization

transformation space

cluster contraction
$\longrightarrow$
local symmetrization

2D Example: Local Symmetrization
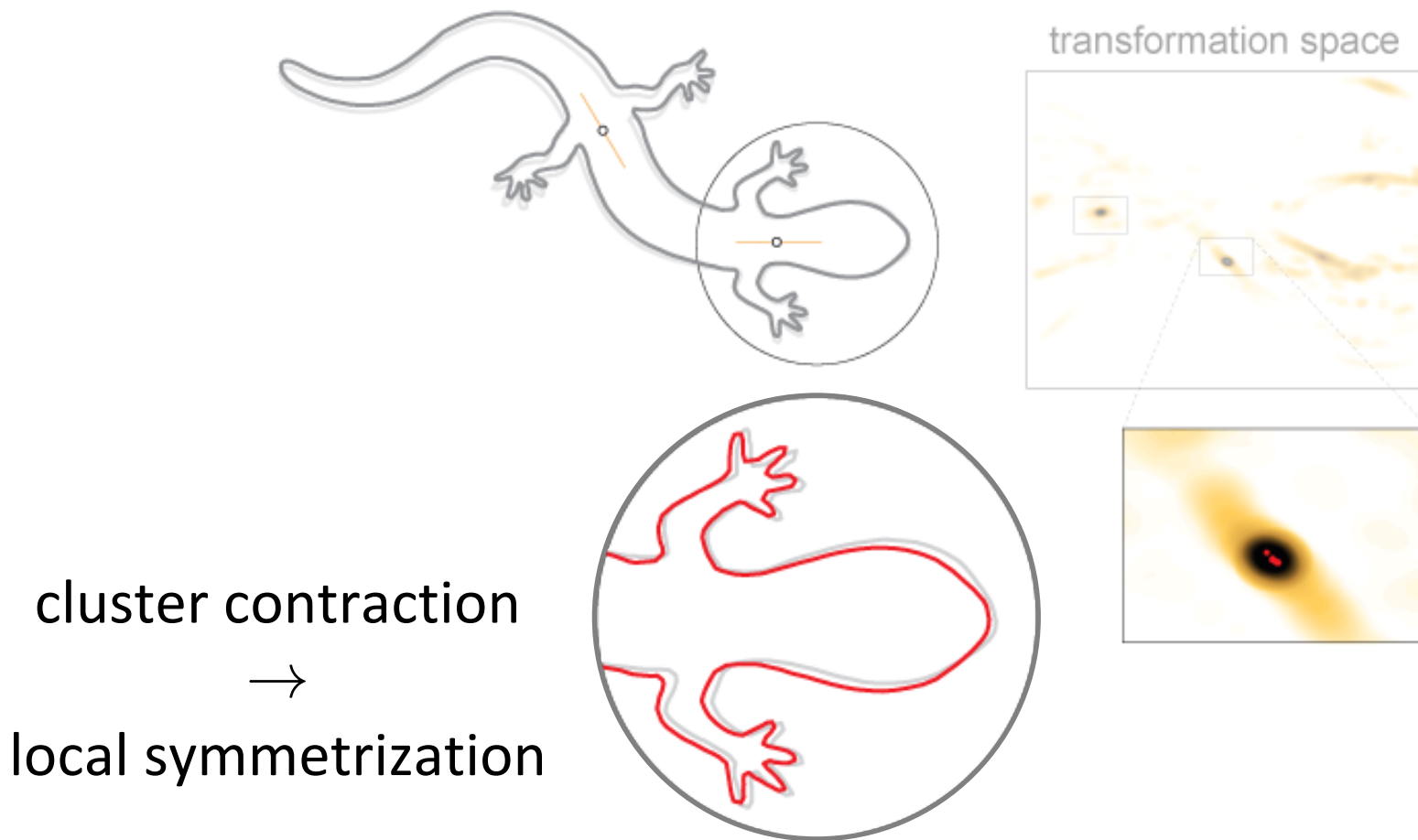
transformation space
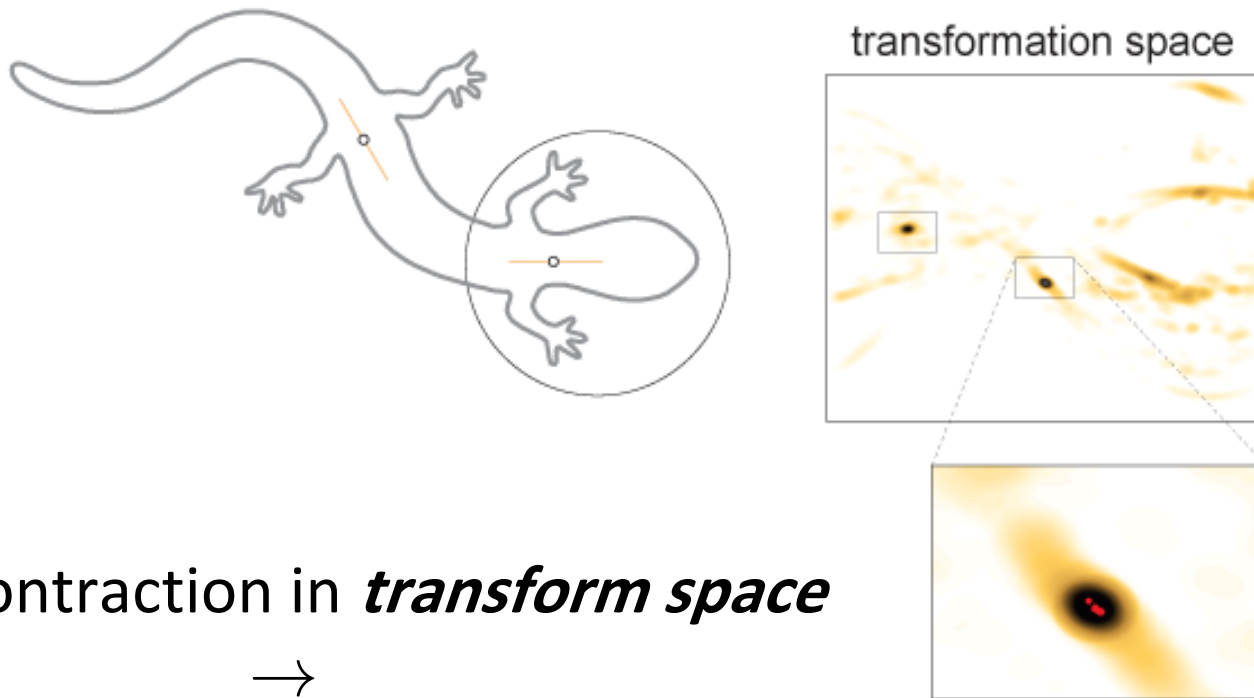
cluster contraction in ***transform space***
$$\rightarrow$$
constrained deformation in ***object space***
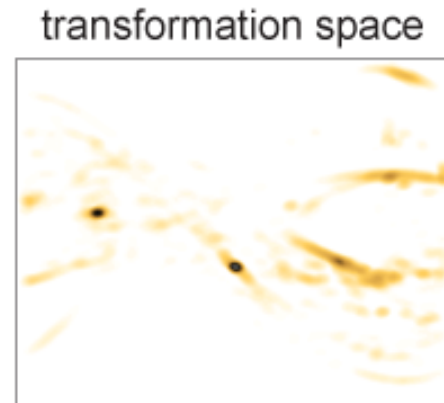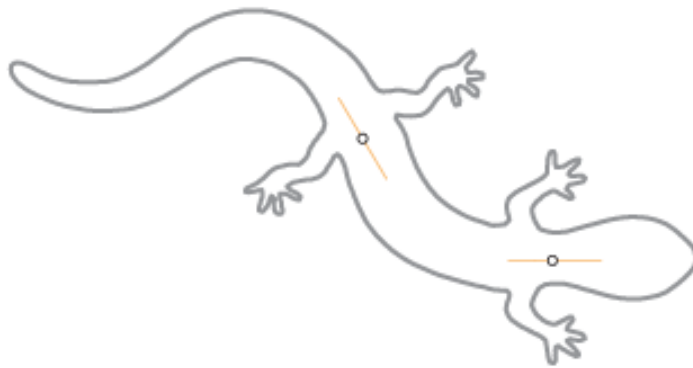
2D Example: Local Symmetrization
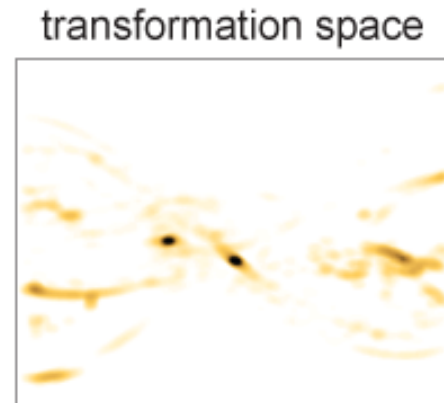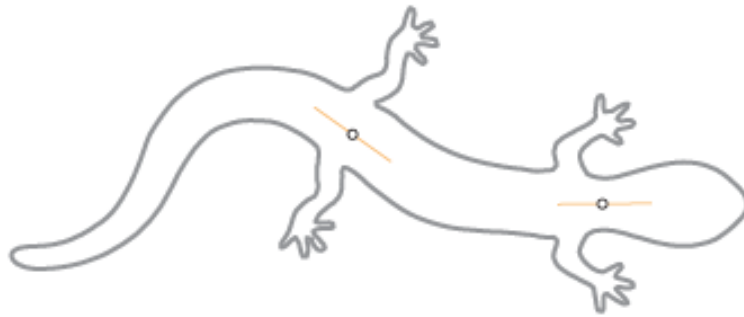

transformation space
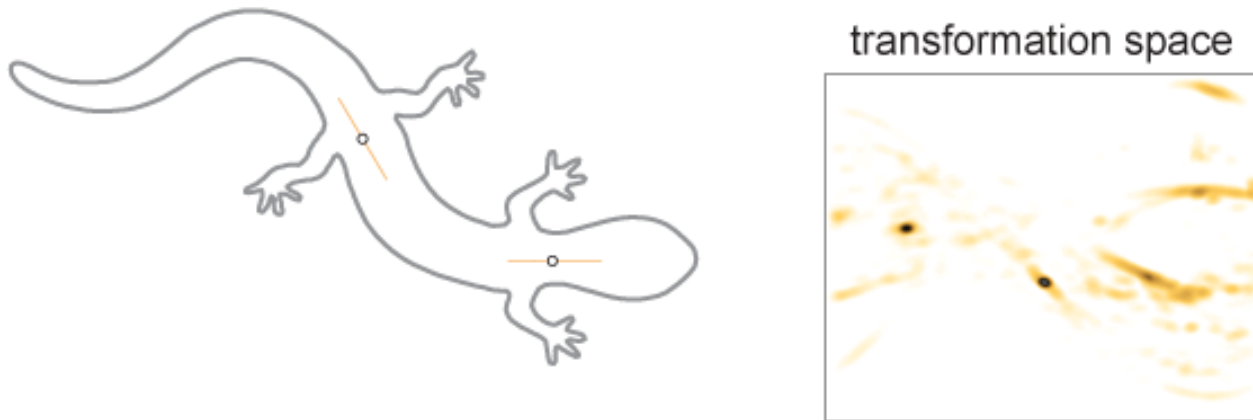
shape after cluster contraction

Recap

- Object space *point pairs → points* in transform space

- *Cluster* in transform space corresponds to approximate symmetry

- *Cluster contraction* in transform space corresponds to *constrained deformation* in object space that enhances object symmetry

iterate

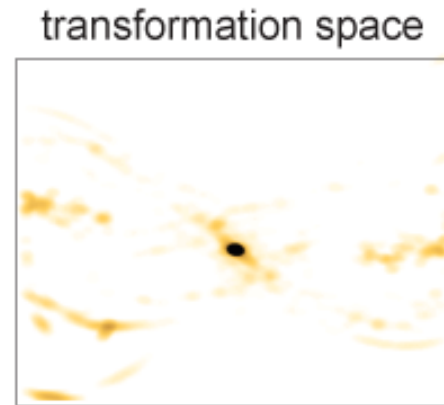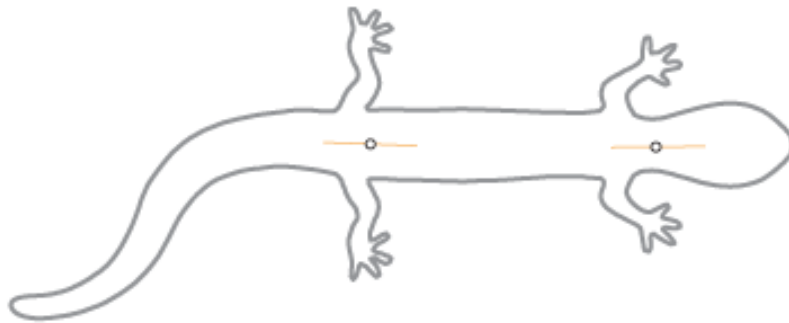# 2D Example: Local Symmetrization

transformation space

2D Example: Global Symmetrization

transformation space

cluster merging → global symmetrization

2D Example: Global Symmetrization

transformation space

cluster merging → global symmetrization

2D Example: Global Symmetrization

transformation space

cluster merging/contraction → global symmetrization

2D Example: Global Symmetrization

transformation space

cluster merging/contraction $\rightarrow$ global symmetrization

2D Example: Global Symmetrization

transformation space
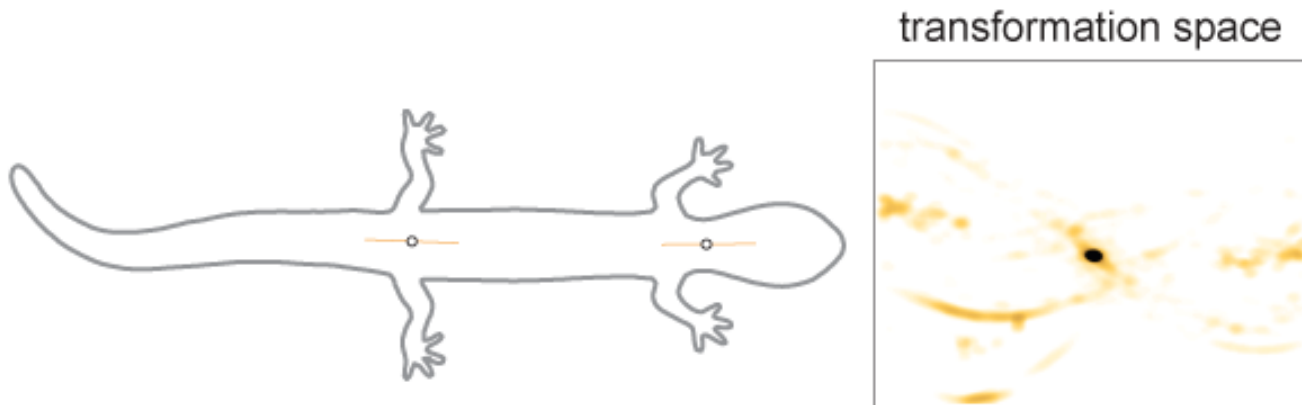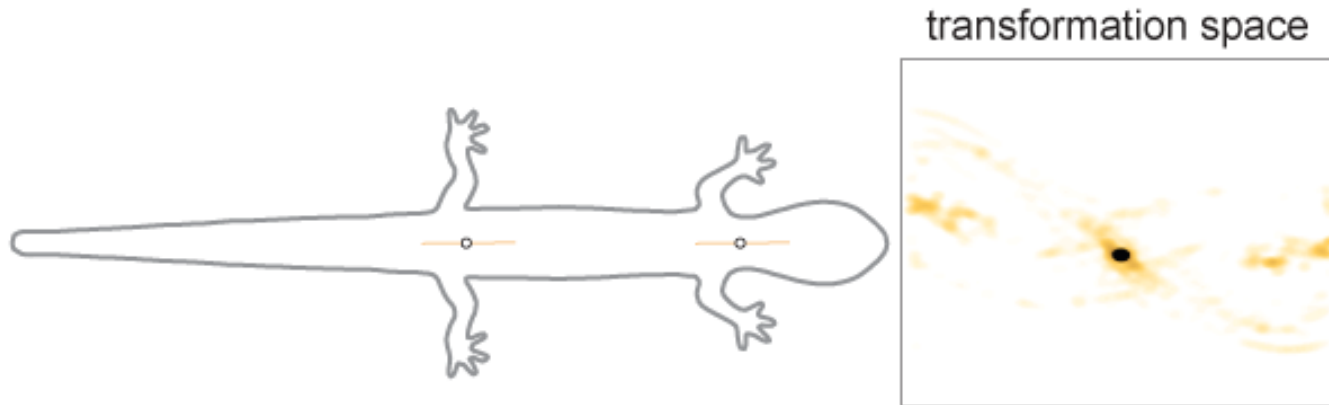
cluster merging/contraction → global symmetrization

Sub-problems

Local Symmetrization

- Cluster contraction
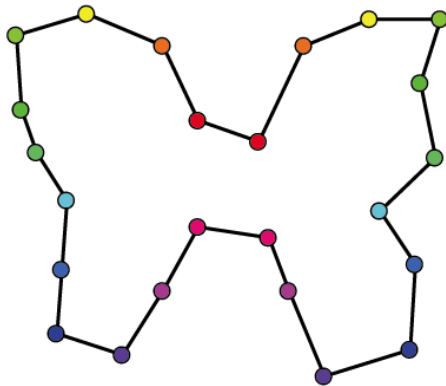
  Where to move in *transform space* ?

  How to deform in the *spatial domain* ?

Global Symmetrization

- Cluster merging

Optimal Transformation

Goal: Find *optimal transformation* and *minimal displacements* for a set of point-pairs

Optimal Transformation

Goal: Find *optimal transformation* and *minimal displacements* for a set of point-pairs
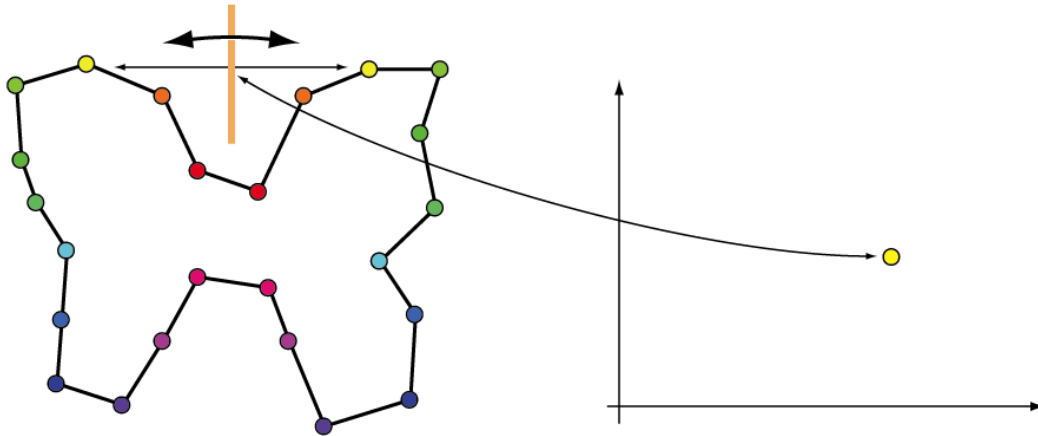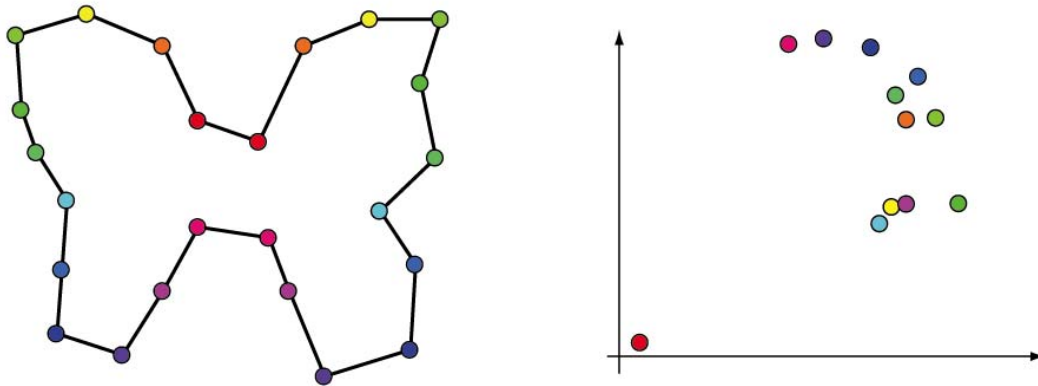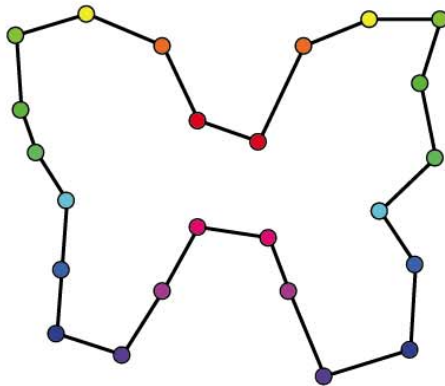
Optimal Transformation

Goal: Find *optimal transformation* and *minimal displacements* for a set of point-pairs
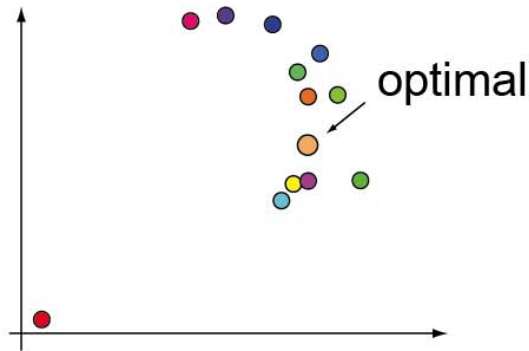
Optimal Transformation

Goal: Find *optimal transformation* and *minimal displacements* for a set of point-pairs

minimize displacements
in object space

closed form solution to compute optimal transformation
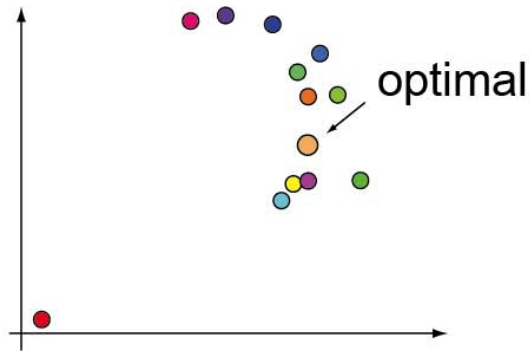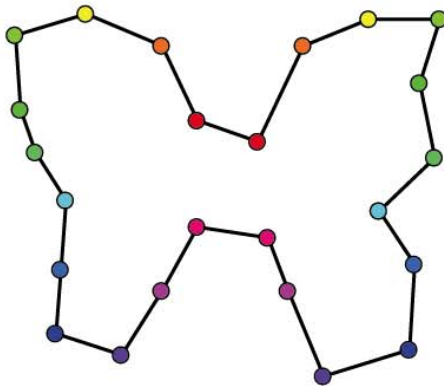
Optimal Transformation

Reflection

- eigenvalue problem

Rigid Transform

- SVD problem

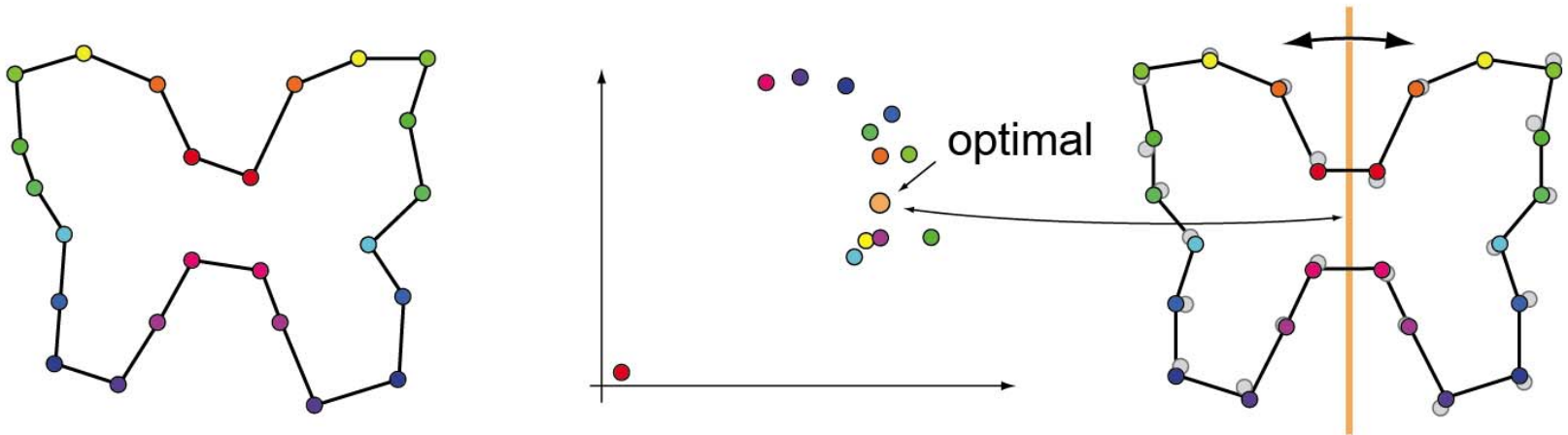- similar to one step of ICP (Iterative Closest Point)

Optimal Transformation

Goal: Find *optimal transformation* and *minimal displacements* for a set of point-pairs

Optimal Transformation

Goal: Find *optimal transformation* and *minimal displacements* for a set of point-pairs

Sub-problems

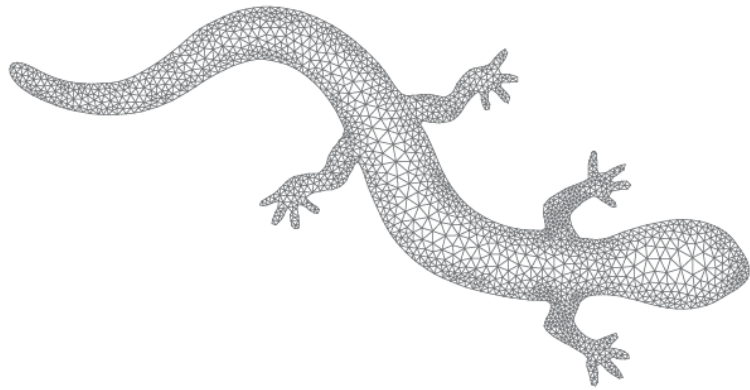Local Symmetrization

- Cluster contraction

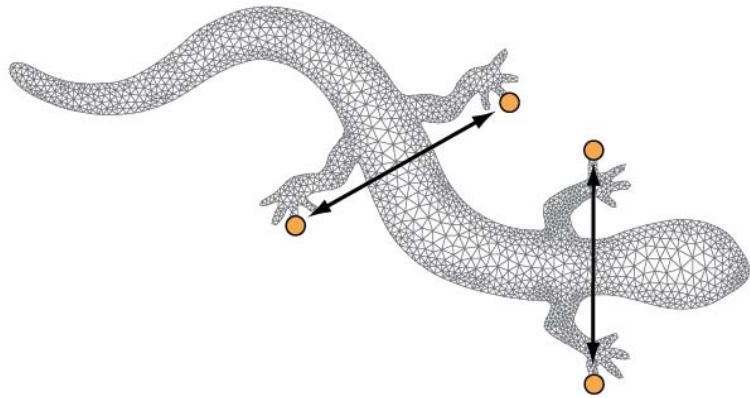 *Where* to move in *transform space* ?

  How to deform in the *spatial domain* ?

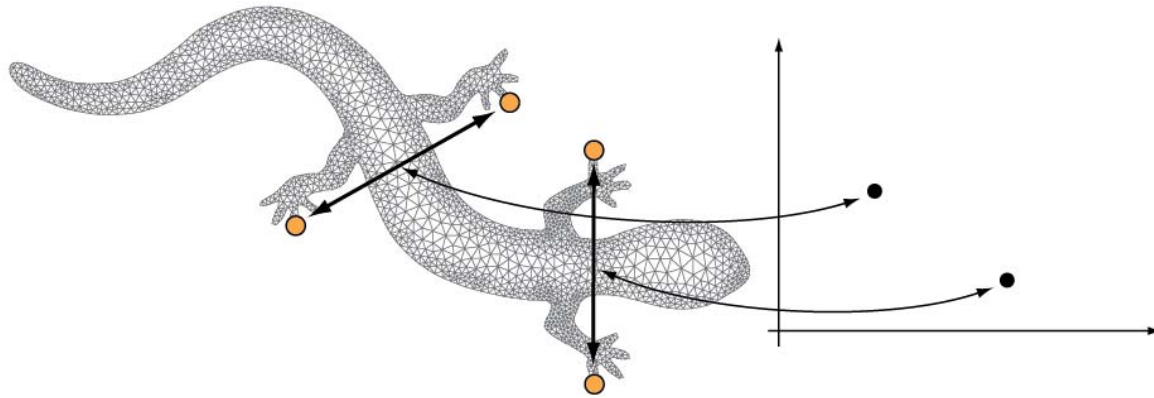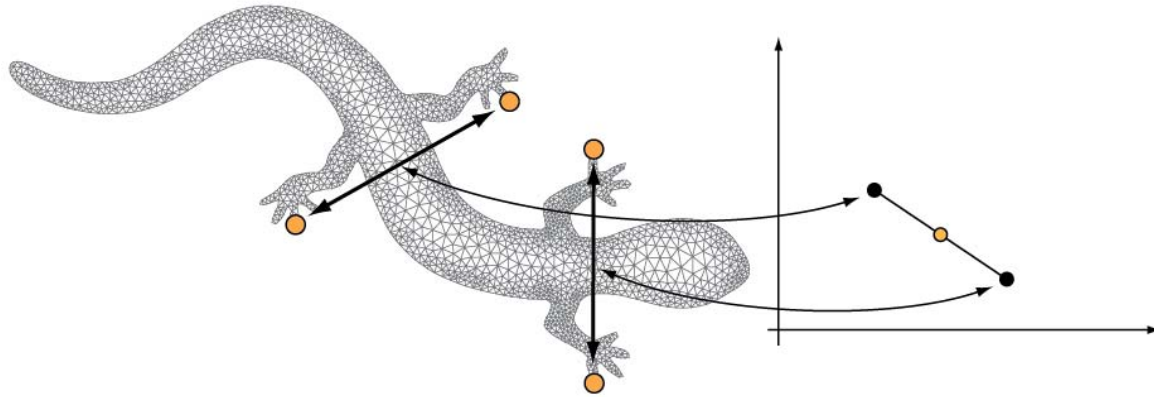## Global Symmetrization

- Cluster merging

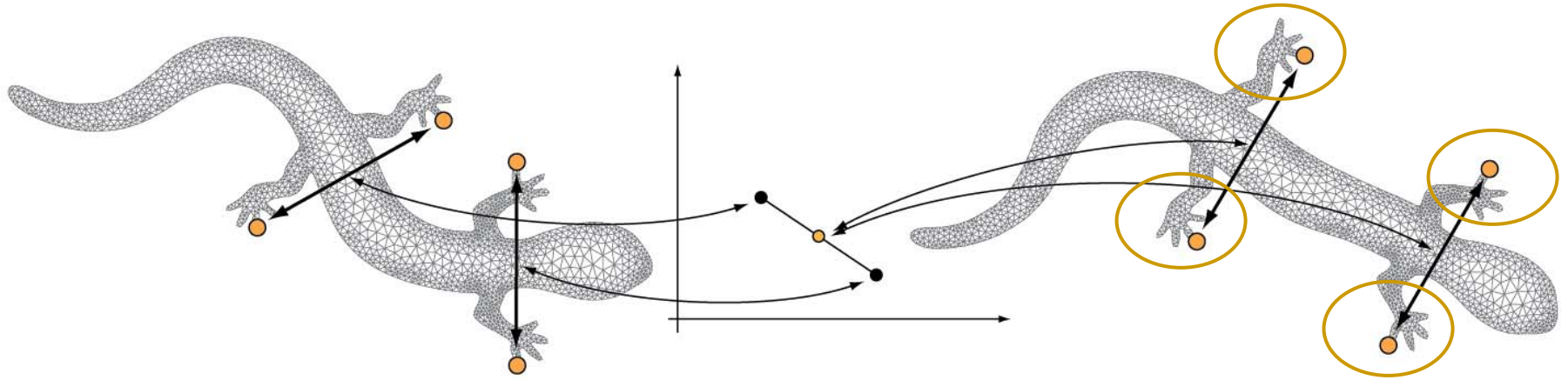# Global Symmetrization

Global Symmetrization
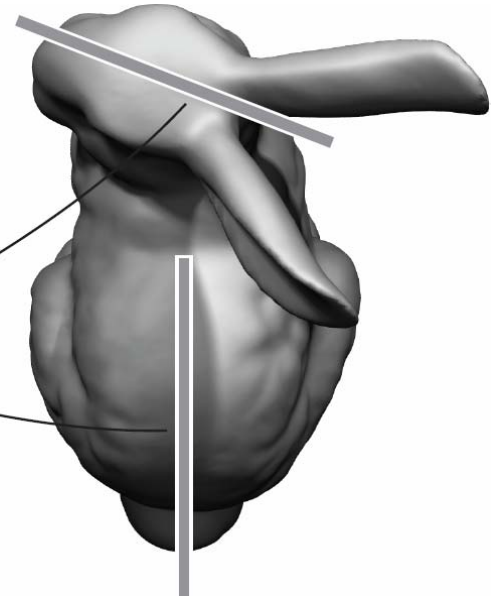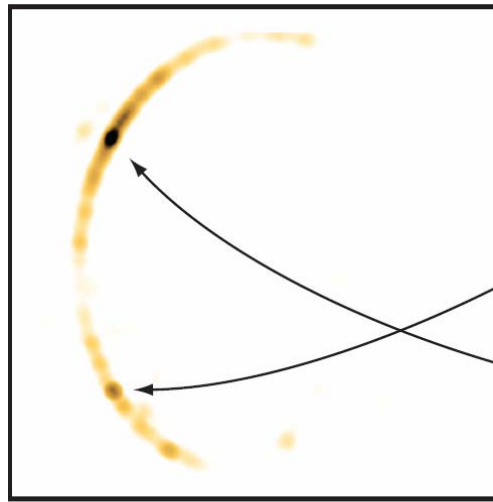
Global Symmetrization

Global Symmetrization

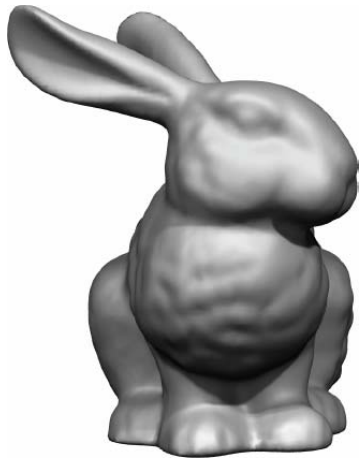# Global Symmetrization

2D: Igarashi, Moscovich, Hughes: *As-Rigid-As-Possible Shape Manipulation*, SIGGRAPH 2005

3D: Botsch, Pauly, Gross, Kobbelt: *PriMo: Coupled Prisms for Intuitive Surface Modeling*, SGP 2006

Stanford Bunny

6D transform space
$\rightarrow$
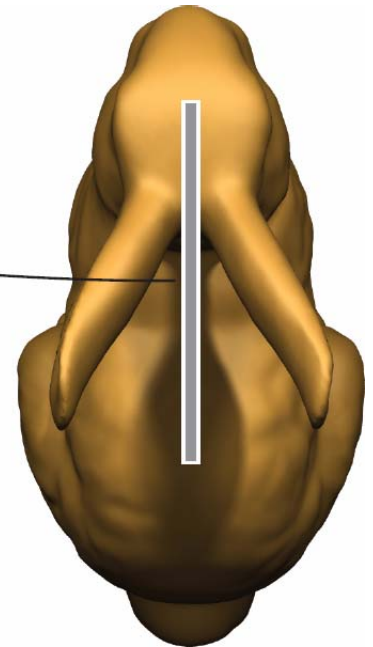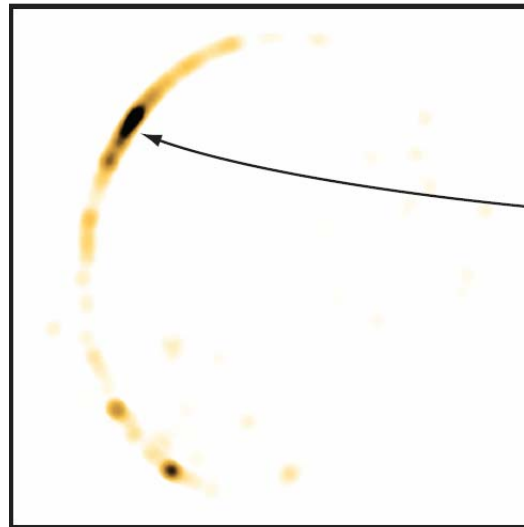2D for visualization

Symmetrizing the Bunny

Cluster Contraction

Transformation Space

Symmetrized Bunny
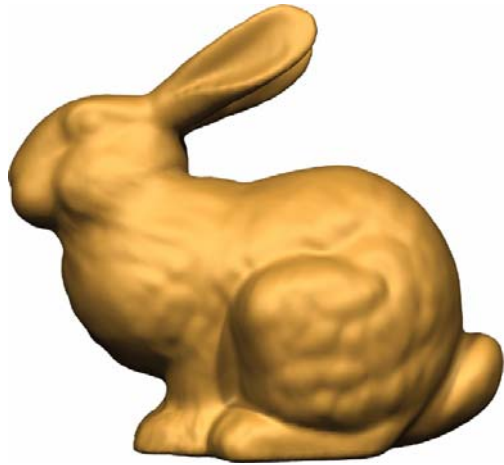
Bunny Feet
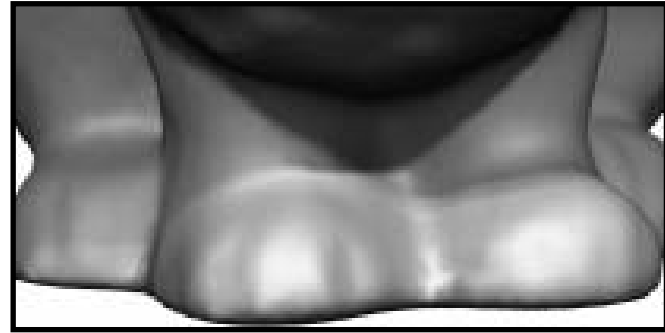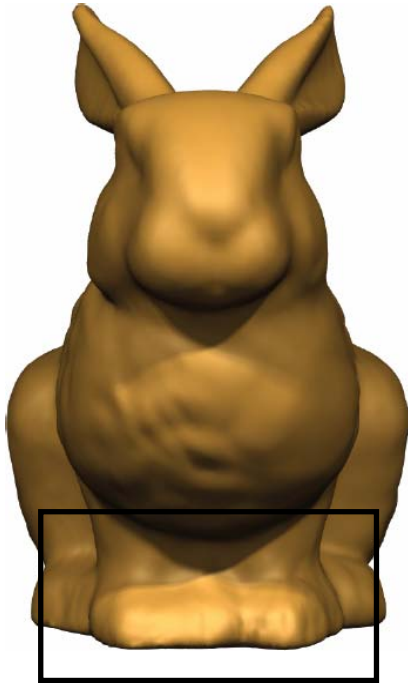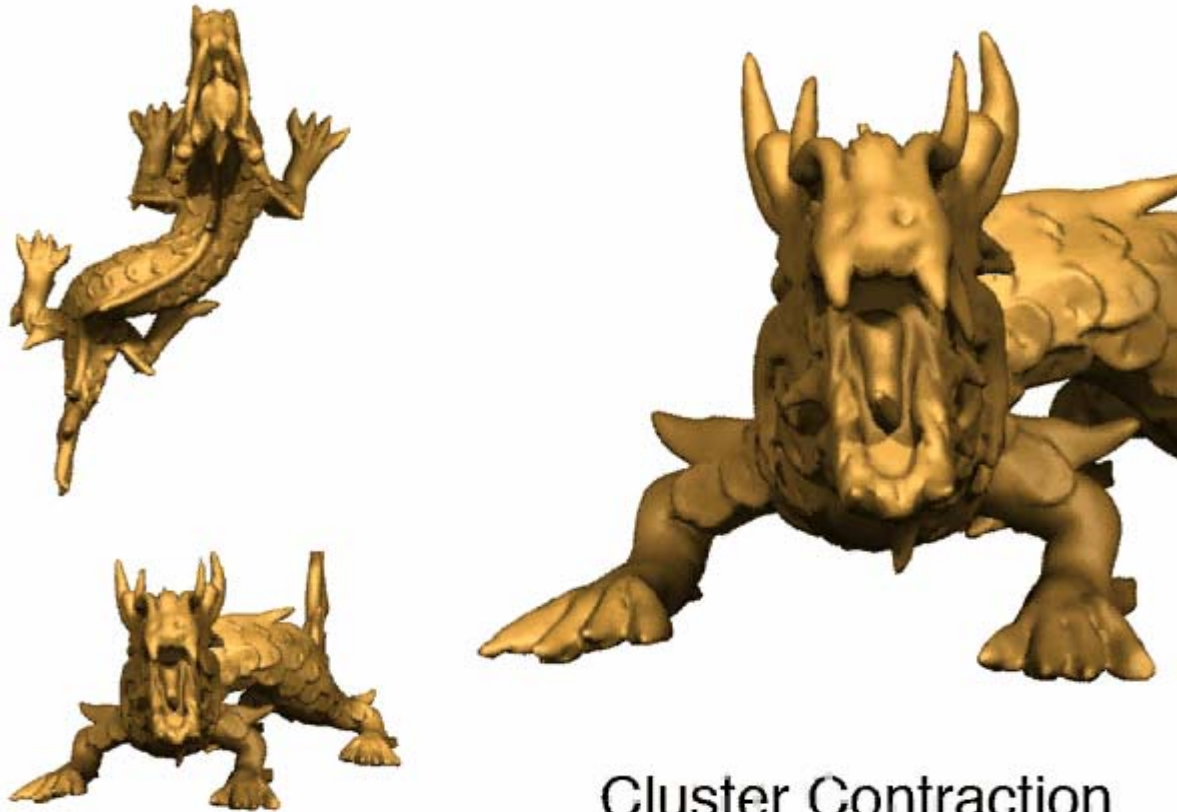
Dragon

Dragon

Dragon

Symmetrizing the Dragon

Cluster Contraction

Symmetrized Dragon

Symmetric Meshing

3000 faces

2000 faces

Registration: Articulated Bodies

Registration: Articulated Bodies

Cluster Merging

Registration: Articulated Bodies

Registration: Geometric Distortion

Conclusion

Optimization that ***couples*** symmetry transformation space and object space to enhance approximate symmetries while minimally altering the shape

Future Work

- symmetry respecting geometry processing
- hierarchical shape semantics
- perception, art, design
- other data, e.g. motion data, derived spaces

Acknowledgments

Mario Botsch

Takeo Igarashi

Helmut Pottmann

Benjamin Schneider

References

Symmetrization

Mitra, Guibas, Pauly
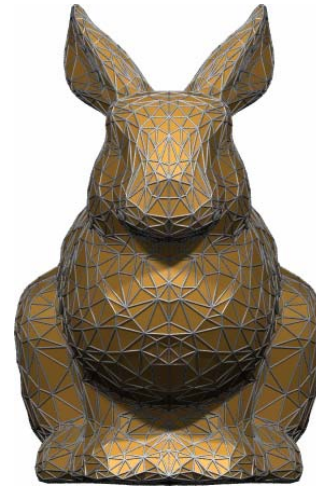
ACM SIGGRAPH 2007 (TOG)

Partial and Approximate Symmetry Detection for 3D Geometry

Mitra, Guibas, Pauly

ACM SIGGRAPH 2006 (TOG)

Thank you!

http://graphics.stanford.edu/~niloy/research/