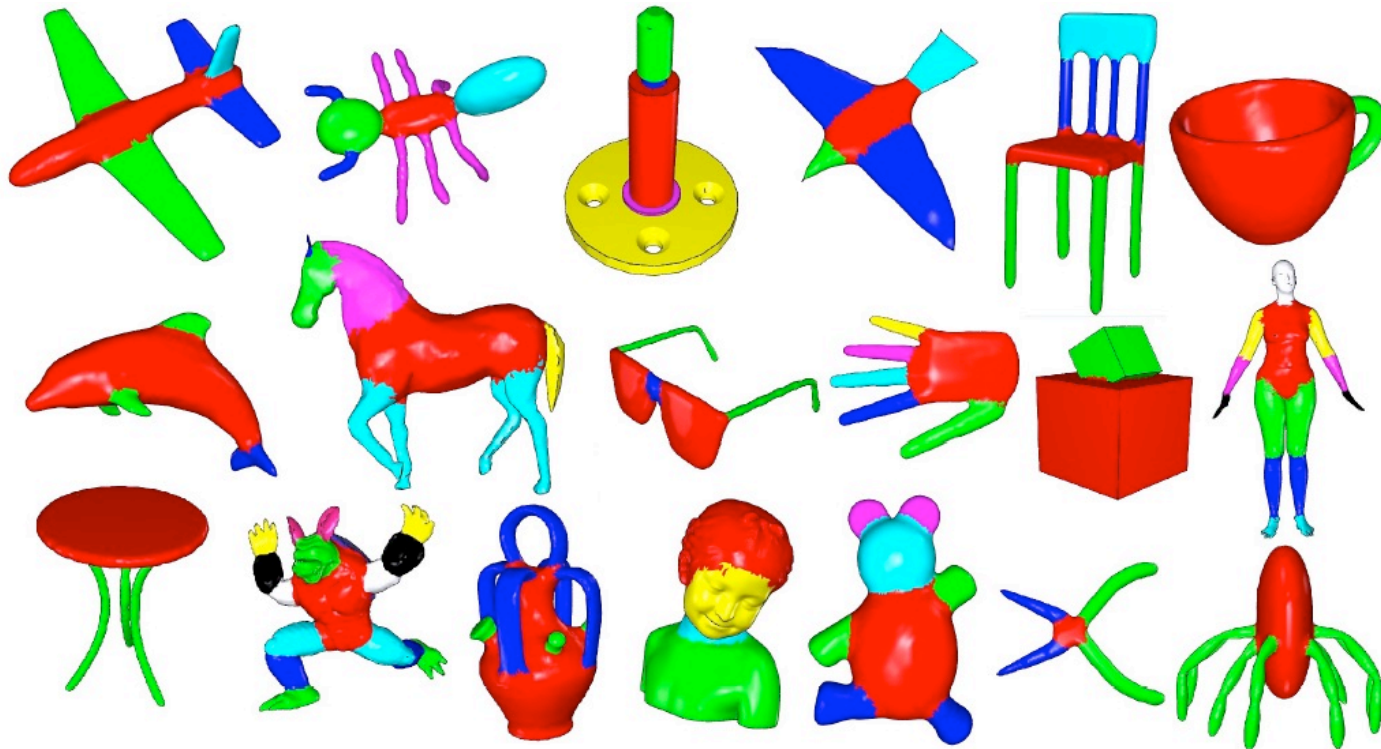


# IFT 6113

## CLUSTERING AND SEGMENTATION

[tiny.cc/ift6113](http://tiny.cc/ift6113)

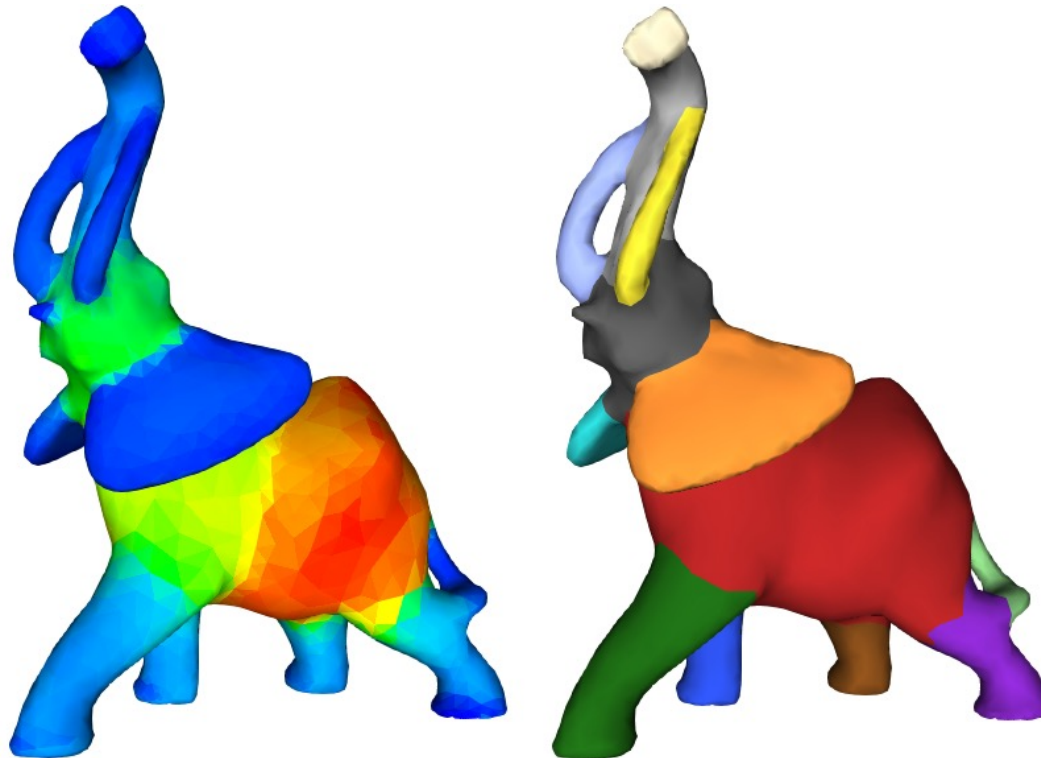


Semi-supervised Mesh Segmentation and Labeling by Lv et al., 2012

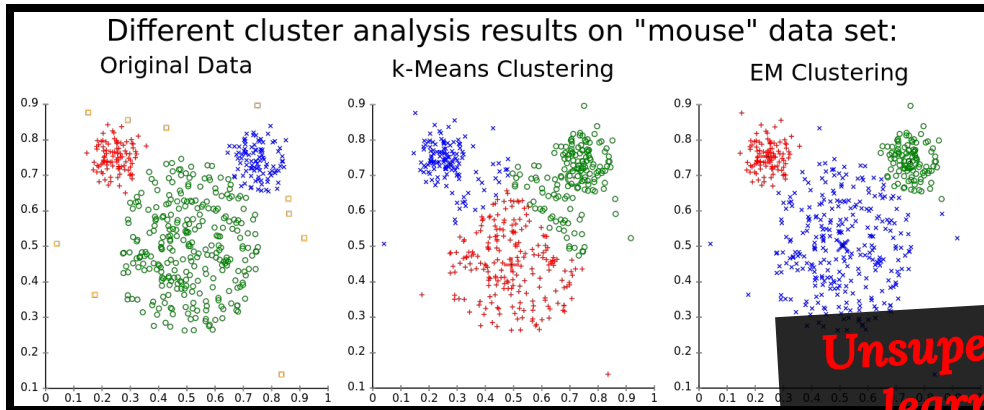
# Mikhail Bessmeltsev

# ***Task:***

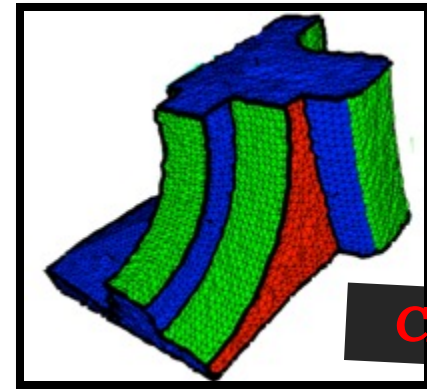
Break a shape into meaningful pieces.



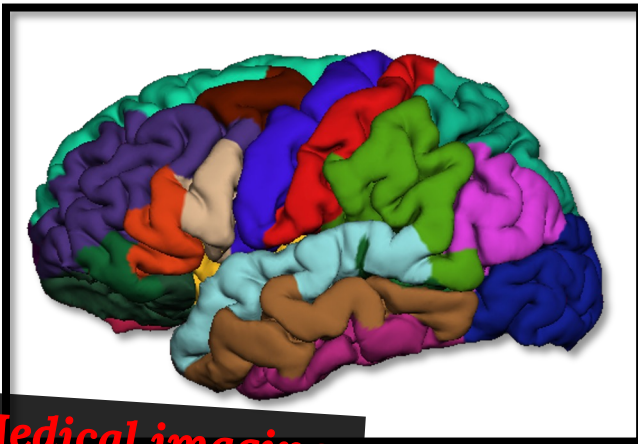
# Many Applications



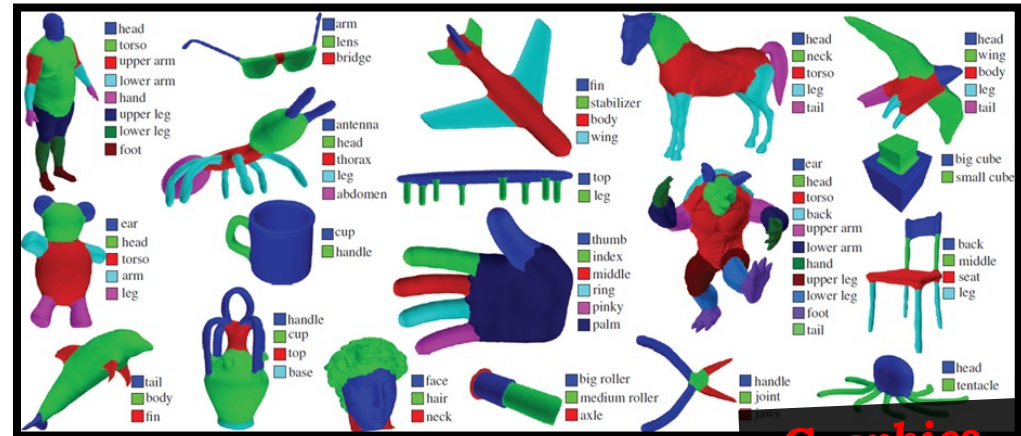
**Unsupervised learning**



**CAD**

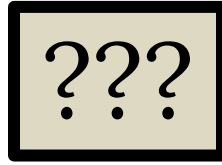


**Medical imaging**

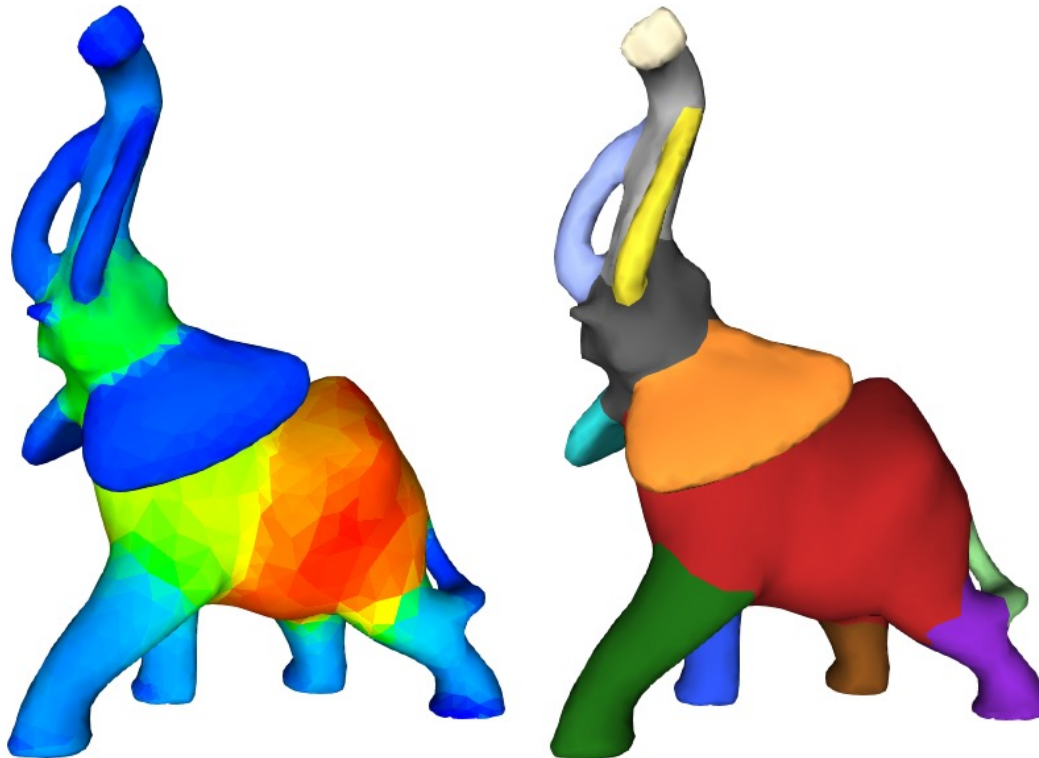


**Graphics**

# Task:

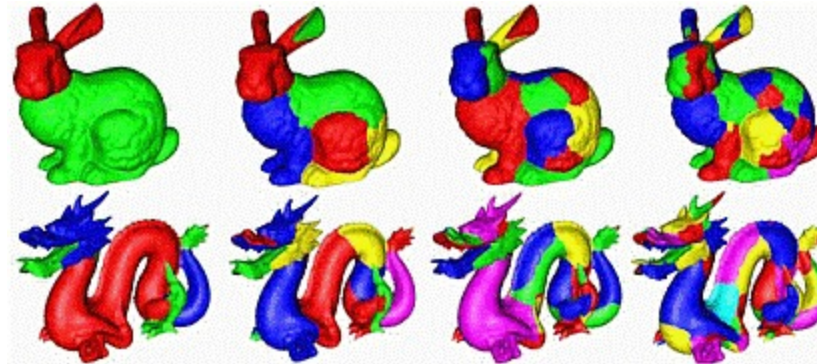


Break a shape into **meaningful** pieces.



# What's meaningful?

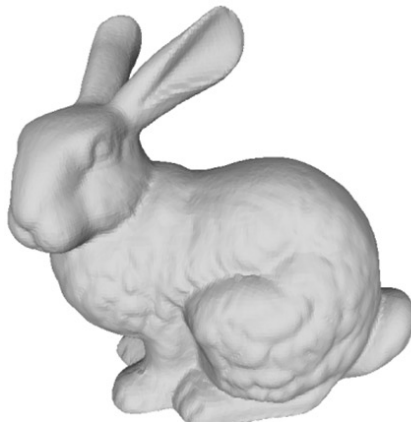
- Semantically?
- Flat/Developable?
  - Convex?
- Voronoi-like?



# What's meaningful?

- Semantically?
- Flat/Developable?
  - Convex?
- Voronoi-like?

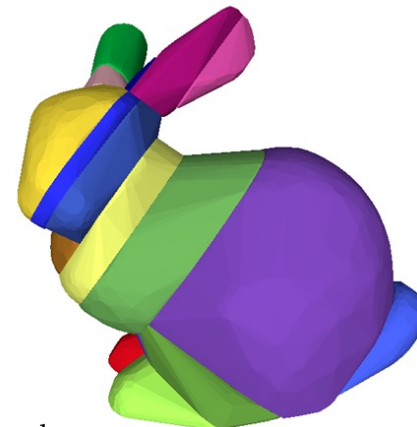
Original Mesh



Exact Convex  
Decomposition  
(7611 parts)



Approximate Convex  
Decomposition  
(20 parts)



*Recall:*

# Related: zippables

## Shape Representation by Zippables

CHRISTIAN SCHÜLLER, ROI PORANNE, and OLGA SORKINE-HORNUNG, ETH Zurich, Switzerland

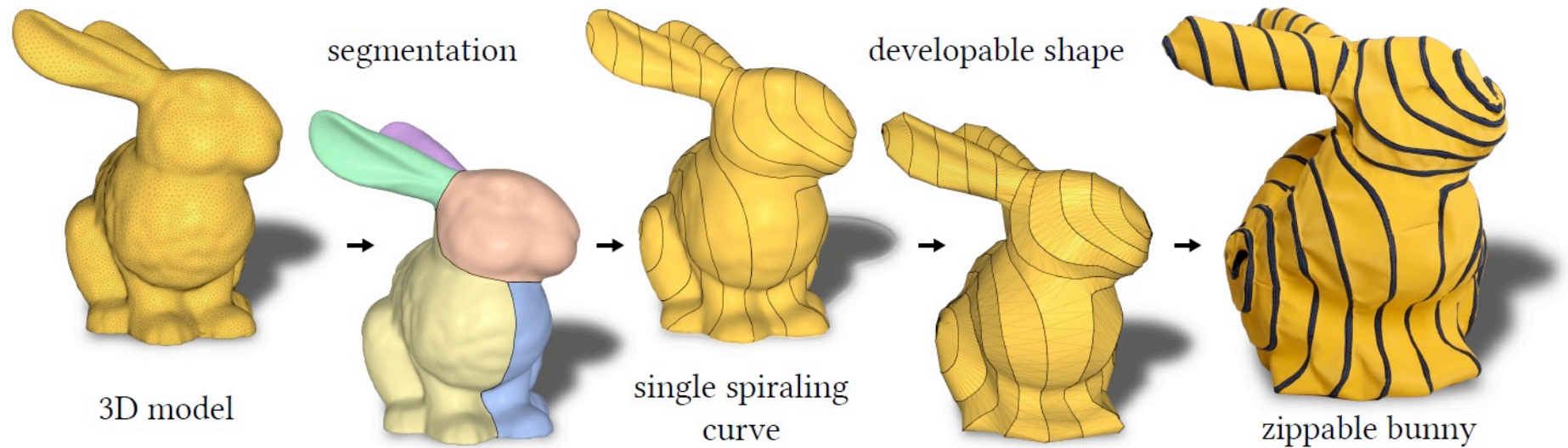
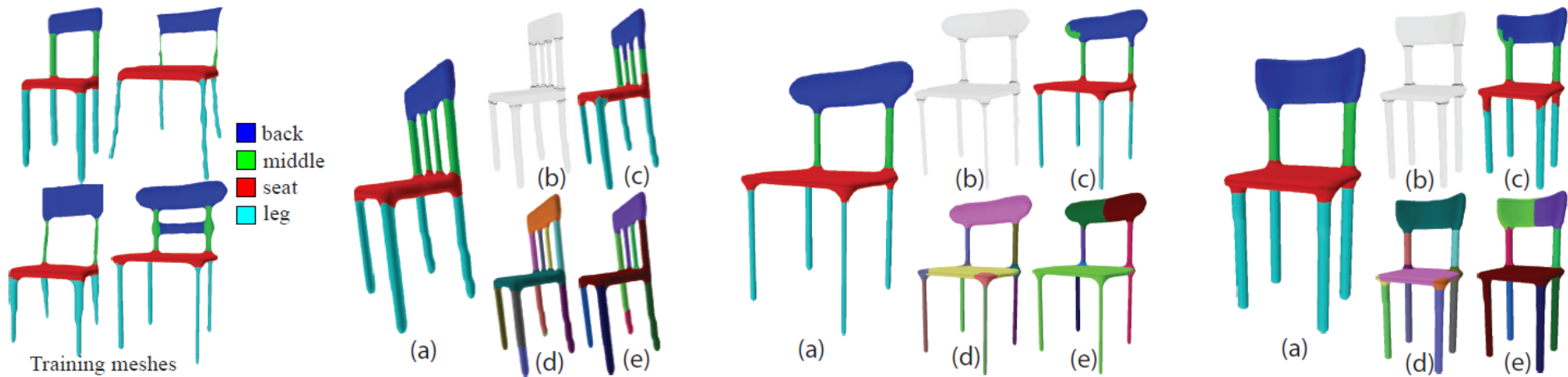


Fig. 1. The pipeline of our approach. Starting from a 3D model, the user decomposes the shape into topological cylinders. Our algorithm automatically produces a single continuous curve on the shape that spirals along the cylinders. It proceeds to cut the shape along the curve and creates a developable surface that can be trivially unfolded into a single 2D shape – the so called *zippable*. Based on the flattening, plans for laser cutting it from fabric are generated. Finally, we attach a zipper with a single slider to the boundary of the zippable. Zipping it up reproduces a faithful approximation of the input model.

# Semantic

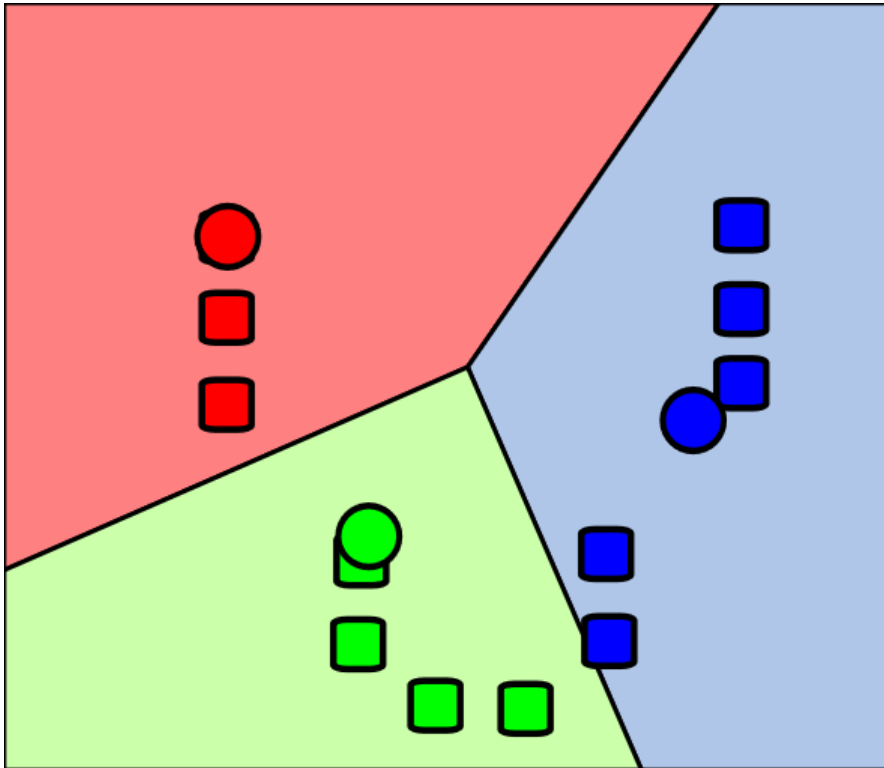
Modeling new chairs using parts from old ones



Learning 3D Mesh Segmentation and Labeling by Evangelos Kalogerakis et al., ACM TOG 2010



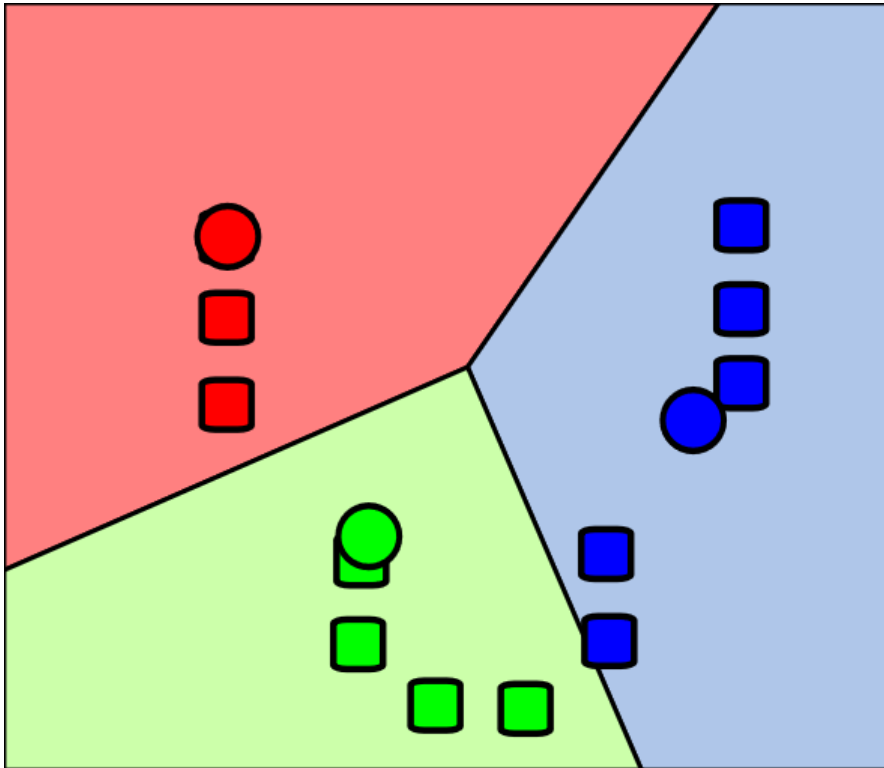
# k-means clustering



$$\min_{S, \mu} \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

[https://upload.wikimedia.org/wikipedia/commons/d/d2/K\\_Means\\_Example\\_Step\\_4.svg](https://upload.wikimedia.org/wikipedia/commons/d/d2/K_Means_Example_Step_4.svg)

# k-means clustering



$$\min_{S, \mu} \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

**NP**-hard for variable  $k$   
*even on a plane*

$O(|S_i|^{d(k+1)})$ , if  $k$  is fixed

# Lloyd iterations

*'the' k-means algorithms*

$$\min_{S, \mu} \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

- **Initialization?**
- **Assignment step (S)**  
 $S_i \leftarrow \{x: \|x - \mu_i\| \leq \|x - \mu_j\| \forall j\}$

- **Update step ( $\mu$ )**

$$\mu_i \leftarrow \frac{1}{|S_i|} \sum_{x \in S_i} x$$

*Recall:*

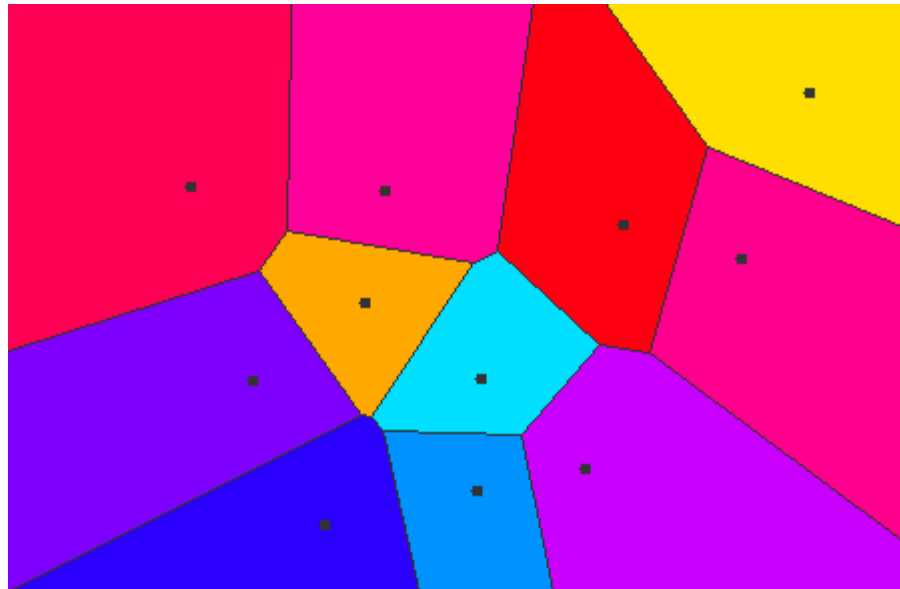
# Voronoi Diagrams



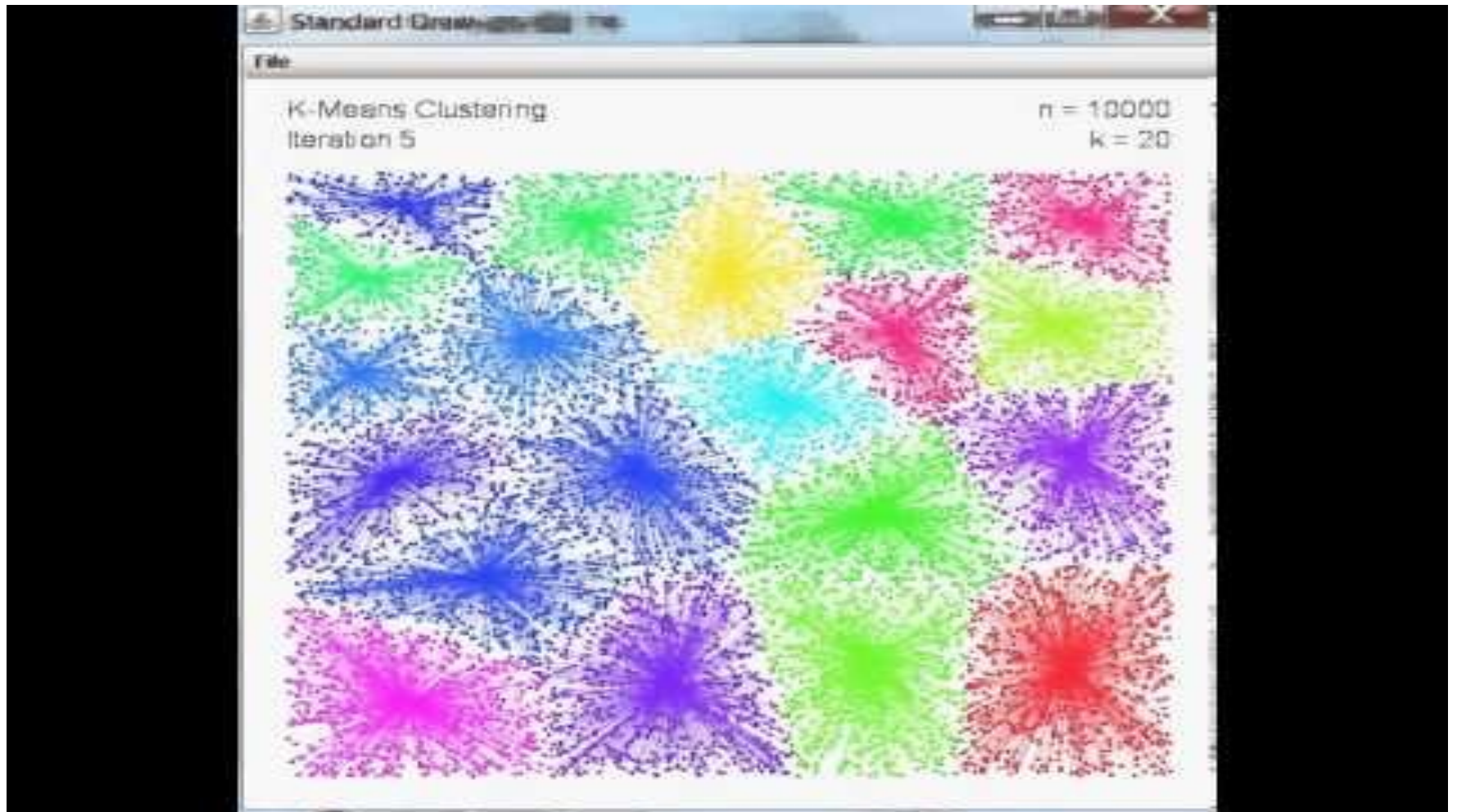
Georgy Voronoi

Георгий Феоодосьевич Вороной

1868-1908



# Example



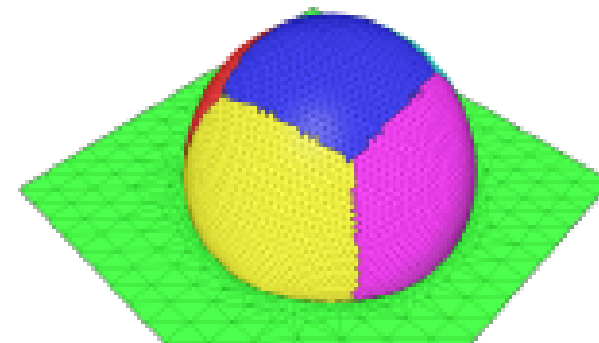
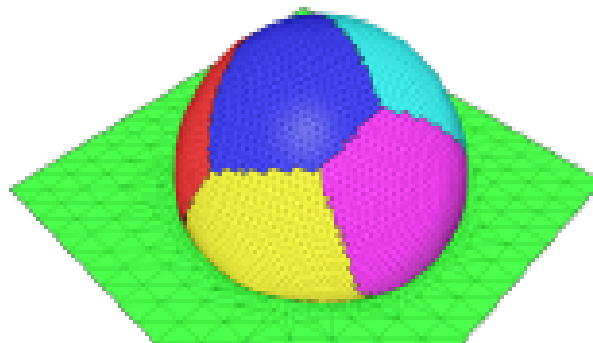
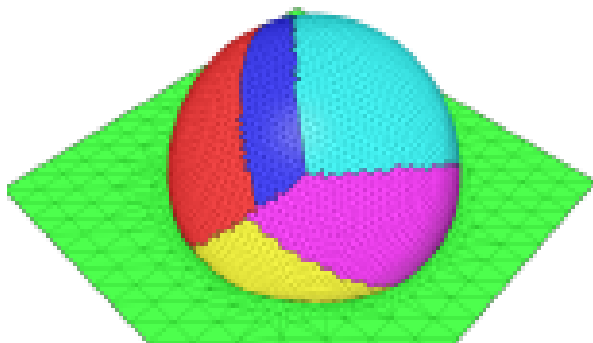
*Recall:*

# Lloyd Iterations

*for segmentation*

Initialization: select random triangles = seeds

1. Grow charts around seeds greedily
2. Find new seed for each chart  
– E.g. centroid
3. Repeat



# Application to Color Space

K=2



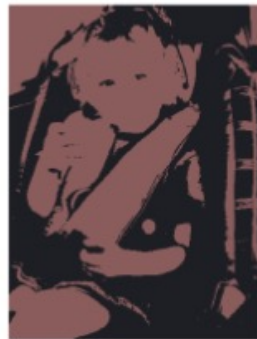
K=3



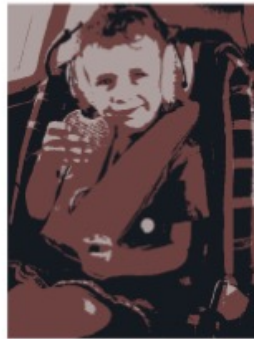
K=10



Original



4%



8%

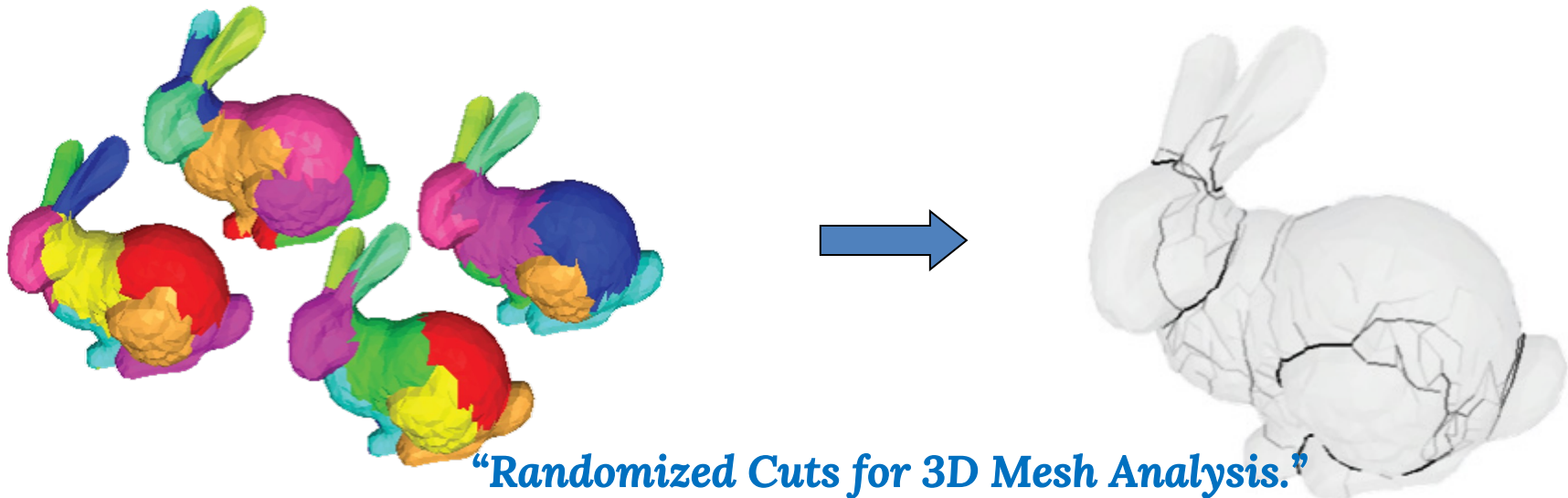


17%



# Dependence on Initial Guess

- Initialize K segment seeds, iterate:
  - Assign faces to closest seed
  - Move seed to cluster center
- Randomization: random initial seeds



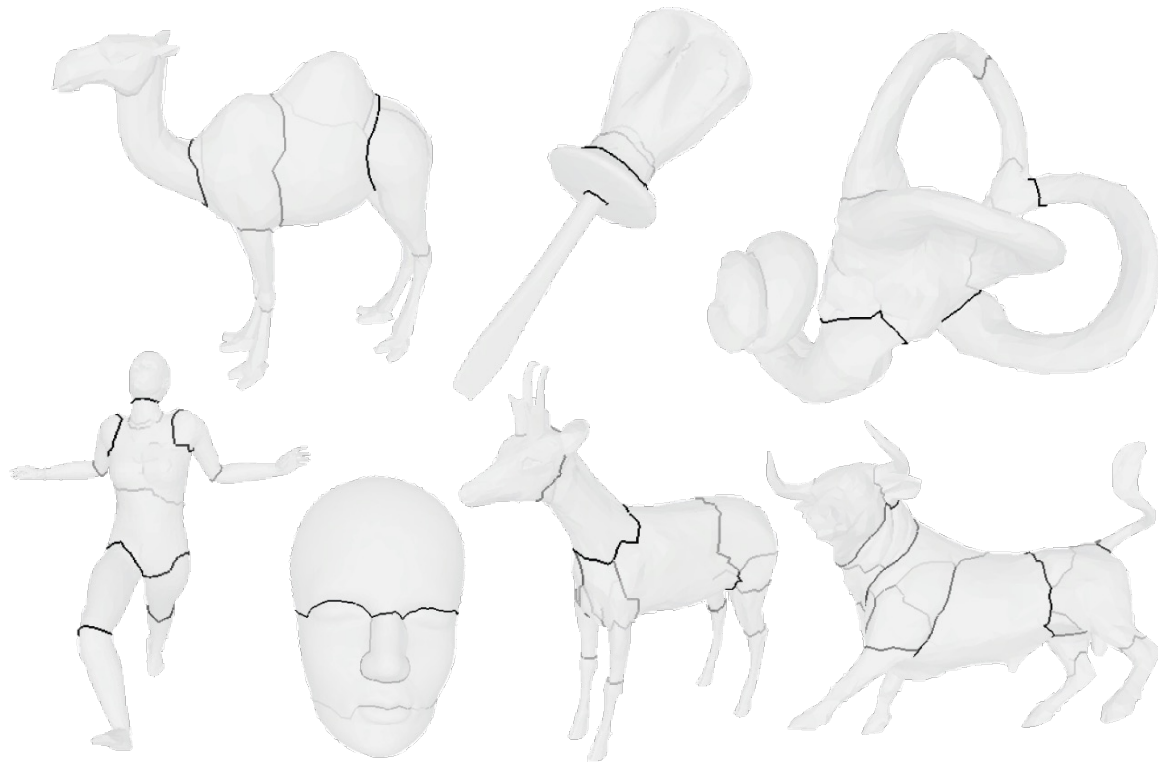
*“Randomized Cuts for 3D Mesh Analysis.”*

*Golovinskiy and Funkhouser; SIGGRAPH Asia 2008*

***Bug ... or feature?***



# Dependence on Initial Guess



*“Randomized Cuts for 3D Mesh Analysis.”*

*Golovinskiy and Funkhouser; SIGGRAPH Asia 2008*

**Bug ... or feature?**

Aside:

# Issue: Choice of $k$

*J. R. Statist. Soc. B* (2001)  
63, Part 2, pp. 411–423

## Estimating the number of clusters in a data set via the gap statistic

Robert Tibshirani, Guenther Walther and Trevor Hastie  
Stanford University, USA

[Received February 2000. Final revision November 2000]

**Summary.** We propose a method (the ‘gap statistic’) for estimating the number of clusters (groups) in a set of data. The technique uses the output of any clustering algorithm (e.g.  $K$ -means or hierarchical), comparing the change in within-cluster dispersion with that expected under an appropriate reference null distribution. Some theory is developed for the proposal and a simulation study shows that the gap statistic usually outperforms other methods that have been proposed in the literature.

**Keywords:** Clustering; Groups; Hierarchy;  $K$ -means; Uniform distribution

### 1. Introduction

Cluster analysis is an important tool for ‘unsupervised’ learning—the problem of finding groups in data without the help of a response variable. A major challenge in cluster analysis is the estimation of the optimal number of ‘clusters’. Fig. 1(b) shows a typical plot of an error measure  $W_k$  (the within-cluster dispersion defined below) for a clustering procedure *versus* the number of clusters  $k$  employed: the error measure  $W_k$  decreases monotonically as the number of clusters  $k$  increases, but from some  $k$  onwards the decrease flattens markedly. Statistical folklore has it that the location of such an ‘elbow’ indicates the appropriate number of clusters. The goal of this paper is to provide a statistical procedure to formalize that heuristic.

For recent studies of the elbow phenomenon, see Sugar (1998) and Sugar *et al.* (1999). A comprehensive survey of methods for estimating the number of clusters is given in Milligan and Cooper (1985), whereas Gordon (1999) discusses the best performers. Some of these methods are described in Sections 5 and 6, where they are compared with our method.

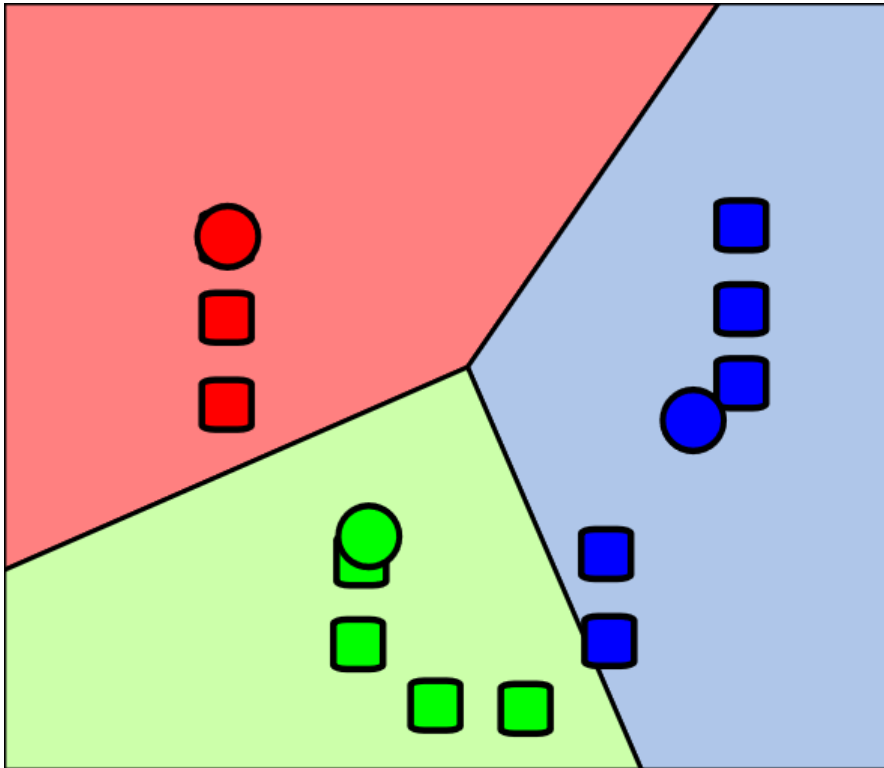
In this paper we propose the ‘gap’ method for estimating the number of clusters. It is designed to be applicable to virtually any clustering method. For simplicity, the theoretical part of our analysis will focus on the widely used  $K$ -means clustering procedure.

### 2. The gap statistic

Our data  $\{x_{ij}\}$ ,  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, p$ , consist of  $p$  features measured on  $n$  independent observations. Let  $d_{ii'}$  denote the distance between observations  $i$  and  $i'$ . The most common choice for  $d_{ii'}$  is the squared Euclidean distance  $\sum (x_{ij} - x_{i'j})^2$ .

“Gap statistic”

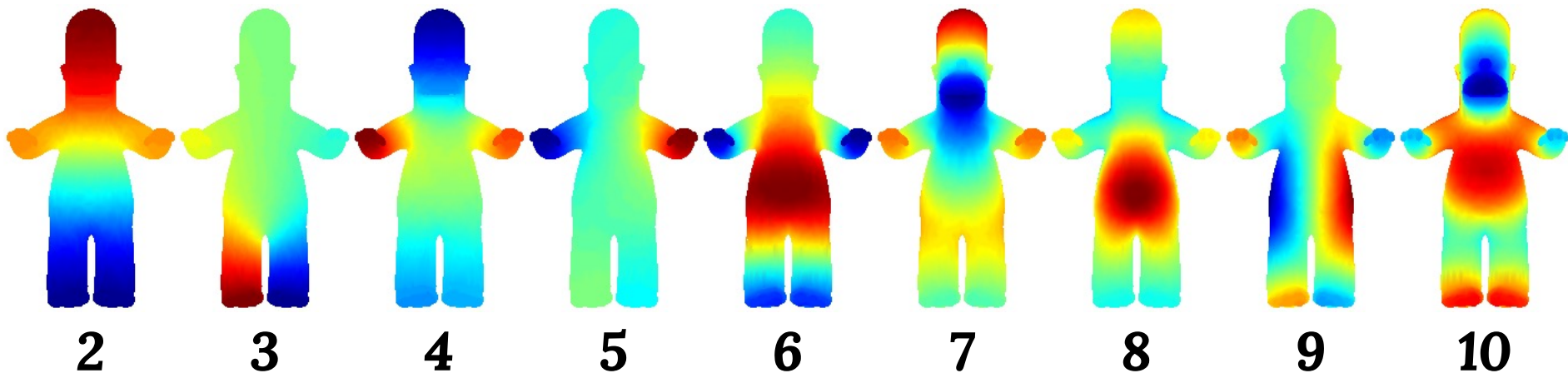
# On a surface?



$$\min_{S, \mu} \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

*Recall:*

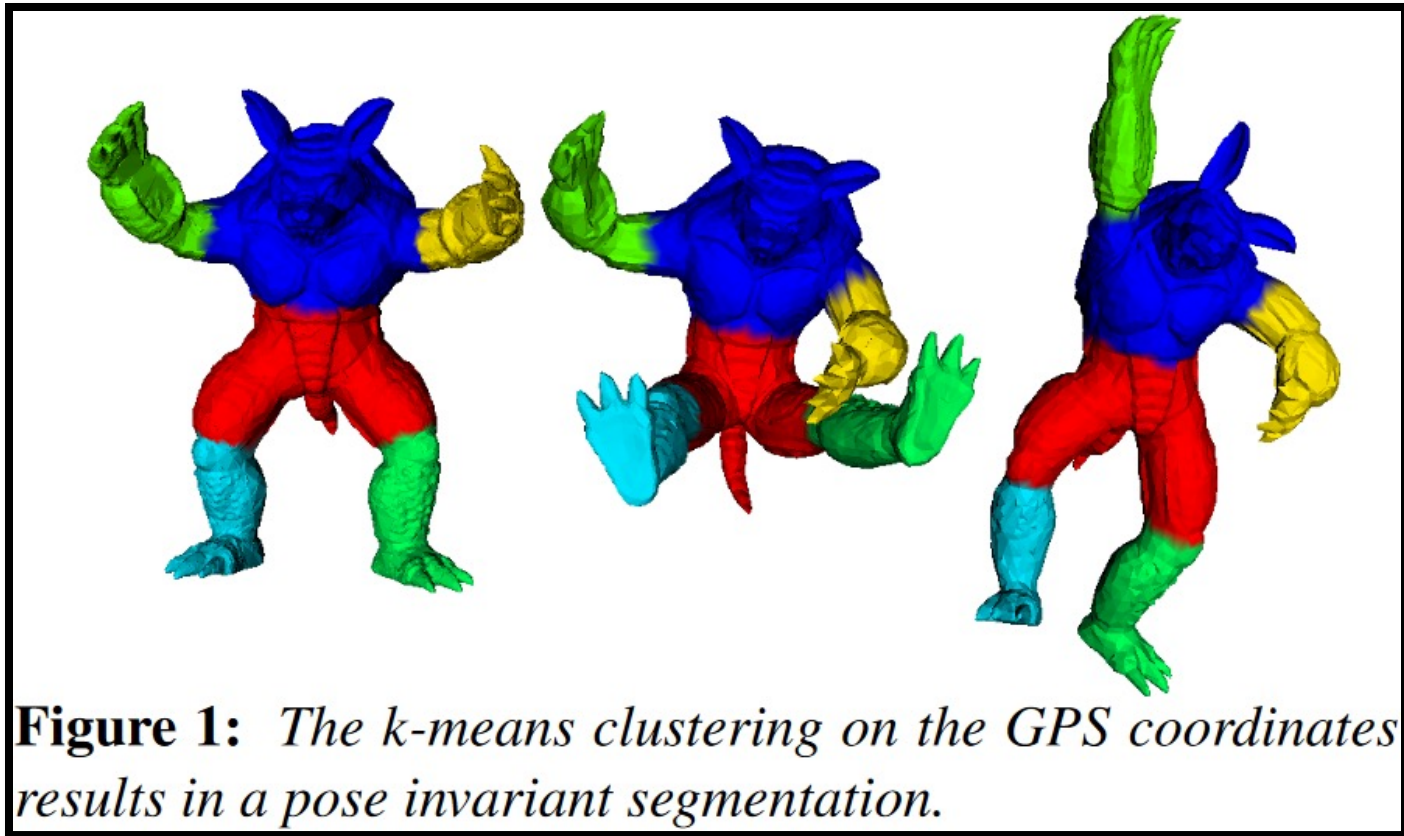
# Global Point Signature



$$\text{GPS}(p) := \left( -\frac{1}{\sqrt{\lambda_1}} \phi_1(p), -\frac{1}{\sqrt{\lambda_2}} \phi_2(p), -\frac{1}{\sqrt{\lambda_3}} \phi_3(p), \dots \right)$$

**“Laplace-Beltrami Eigenfunctions for Deformation Invariant Shape Representation”  
Rustamov, SGP 2007**

# Can Apply to Features





**“Laplace-Beltrami Eigenfunctions for Deformation Invariant Shape Representation.”**  
**Rustamov; SGP 2007**

# Geometry of $k$ -Means

- **Assignment step**
  - Assign point to its **closest** cluster center
- **Update step**
  - **Average** all points in a cluster

**Doesn't have to be Euclidean**

# Geometry of $k$ -Means

- **Assignment step** 
  - Assign point to its **closest** cluster center
- **Update step** 
  - **Average** all points in a cluster

**In a metric space**

# Fréchet Mean

$$\arg \min_{p \in M} \sum_{i=1}^N d(p, x_i)^2$$

*“Fréchet variance”*

*On the board:*

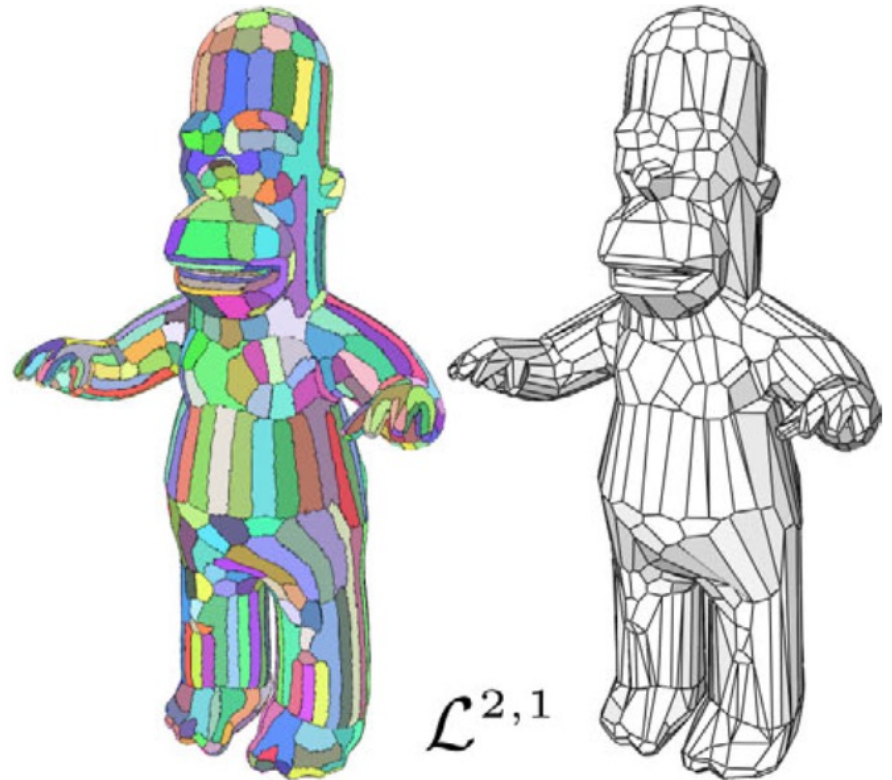
**Generalizes Euclidean notation of “mean.”**



# Extension to Regions on a Surface

Alternate between

1. Fitting primitive parameters
2. Assign points to patches



*“Variational Shape Approximation.”*

Cohen-Steiner, Alliez, and Desbrun; SIGGRAPH 2004

# *k*-Medoids

- **Assignment step**
  - Assign point to its **closest** cluster center

- **Update step**
  - Replace cluster center with most **central data point**

**When Fréchet means won't work**

# Related Technique

## **Region Growing Algorithm**

Initialize a priority queue  $Q$  of elements

Loop until all elements are clustered

    Choose a seed element and insert to  $Q$

    Create a cluster  $C$  from seed

    Loop until  $Q$  is empty

        Get the next element  $s$  from  $Q$

        If  $s$  can be clustered into  $C$

            Cluster  $s$  into  $C$

            Insert  $s$  neighbors to  $Q$

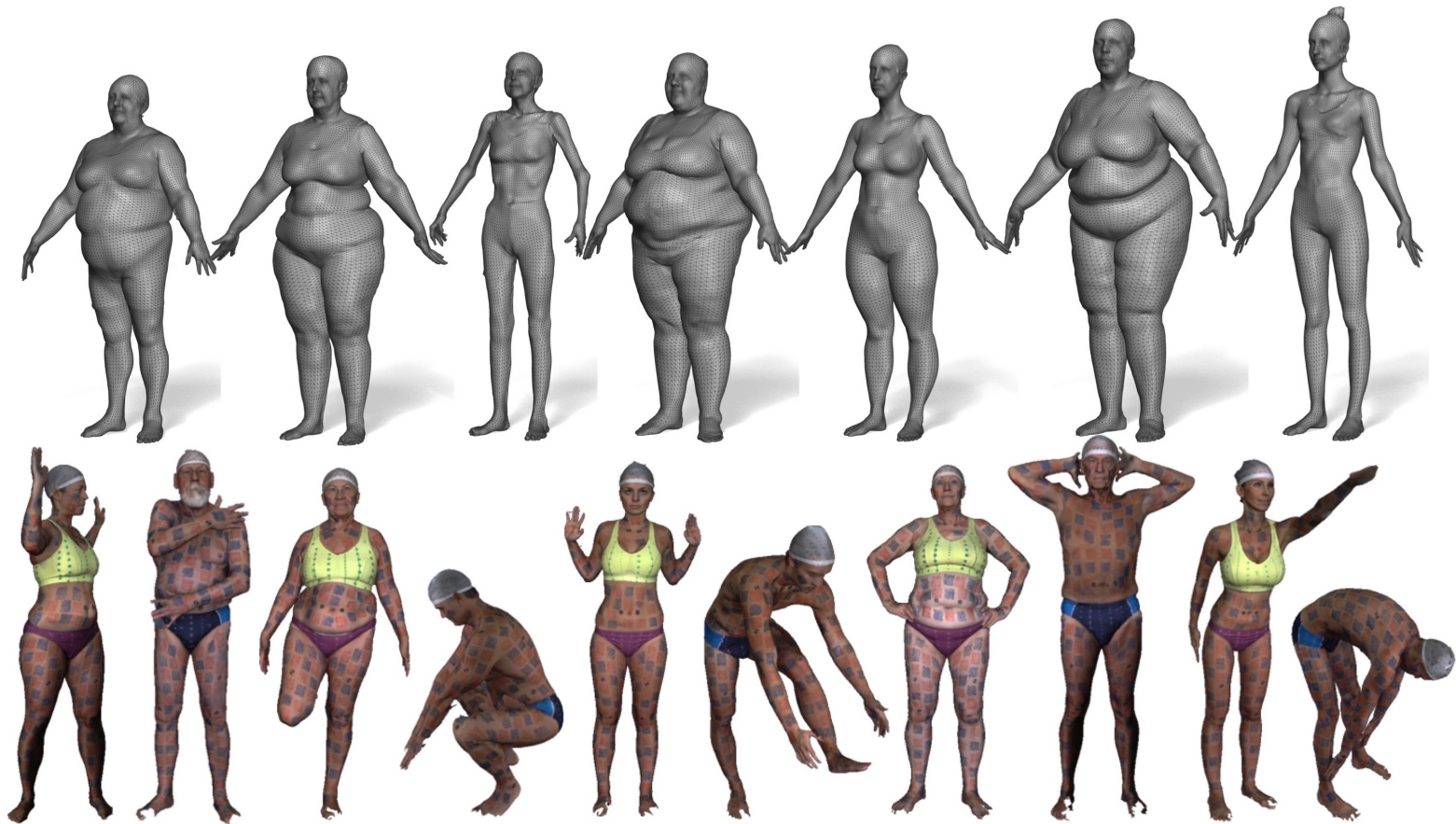
Merge small clusters into neighboring ones

*“Segmentation and Shape Extraction of 3D Boundary Meshes.”*

**Shamir; EG STAR 2006.**

**Region growing algorithm**

# Example Task



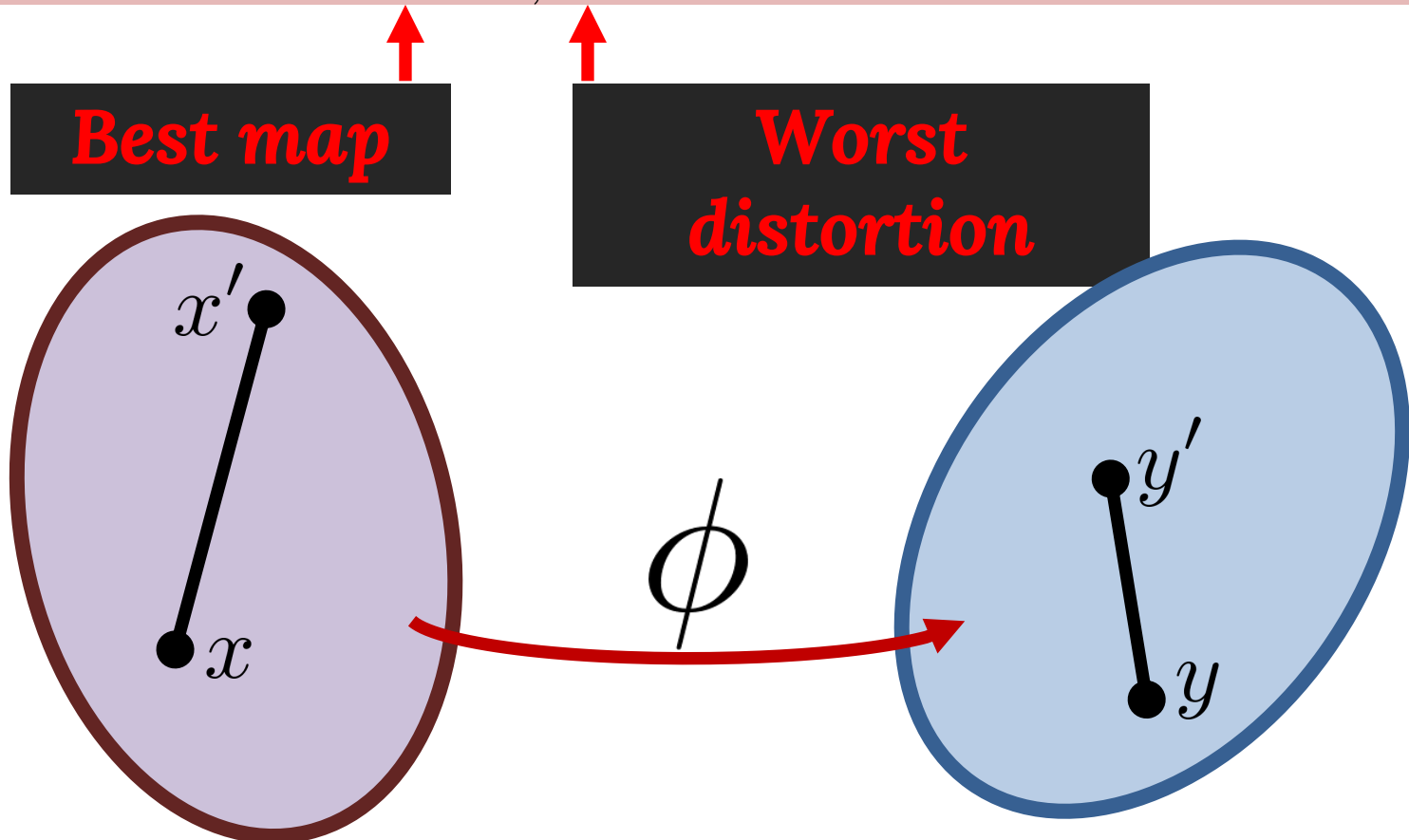
[https://ps.is.tuebingen.mpg.de/research\\_projects/3d-mesh-registration](https://ps.is.tuebingen.mpg.de/research_projects/3d-mesh-registration)

**Clustering shapes?**

# Gromov-Hausdorff Distance

**Distance between metric spaces  $X, Y$**

$$d_{\text{GH}}(X, Y) := \inf_{\phi: X \rightarrow Y} \sup_{x, x' \in X} |d_X(x, x') - d_Y(\phi(x), \phi(x'))|$$



# Gromov-Hausdorff Clustering

Eurographics Symposium on Point-Based Graphics (2007)  
 M. Botsch, R. Pajarola (Editors)

## On the use of Gromov-Hausdorff Distances for Shape Comparison

Facundo Mémoli<sup>1†</sup>

<sup>1</sup>Department of Mathematics, Stanford University, California, USA.

### Abstract

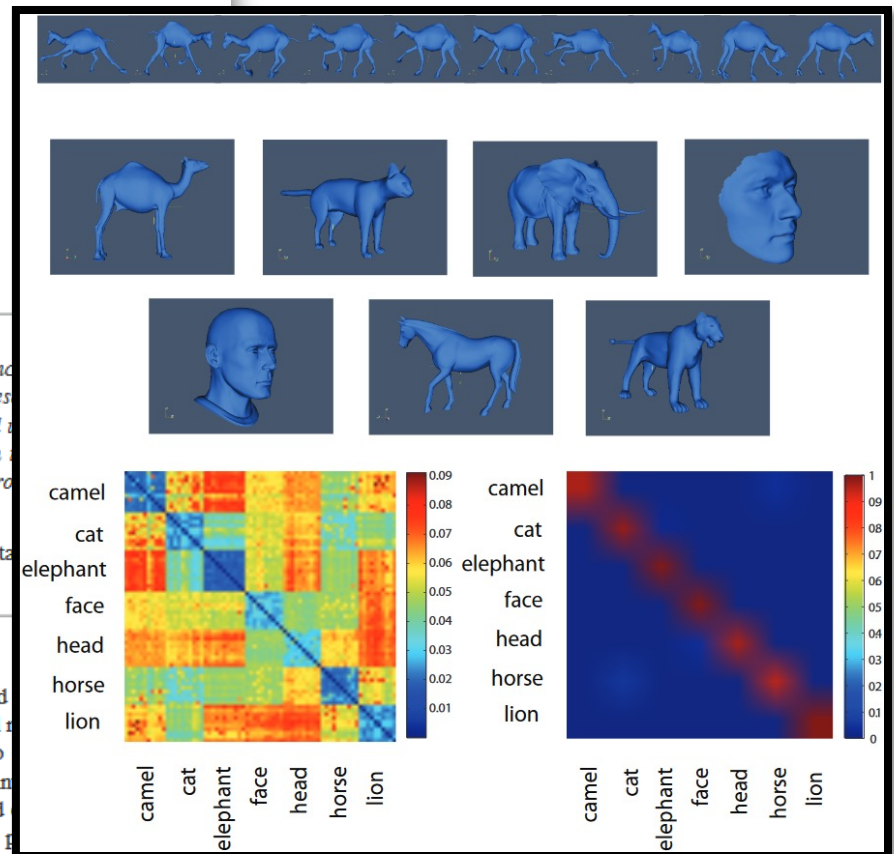
*It is the purpose of this paper to propose and discuss certain modifications of the ideas concerning Hausdorff distances in order to tackle the problems of shape matching and comparison. These modifications render these distances more amenable to practical computations without sacrificing theoretical soundness. A second goal of this paper is to establish links to several other practical methods proposed in the literature for comparing/matching shapes in precise terms. Connections with the Quadratic Assignment Problem are also established, and computational examples are presented.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modelling.

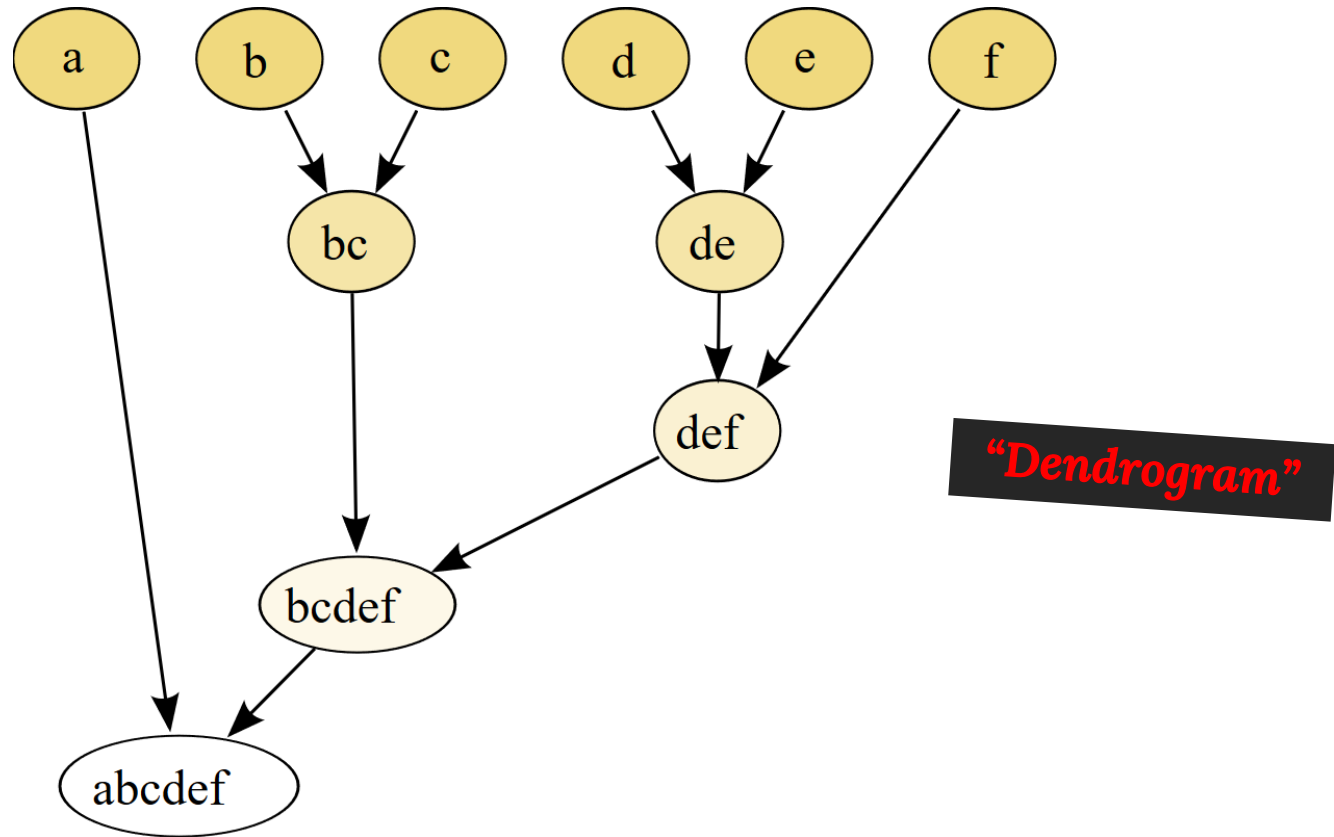
### 1. Introduction

Given the great advances in recent years in the fields of shape acquisition and modelling, and the resulting huge collections of digital models that have been obtained it is of great importance to be able to define and compute meaningful notions of similarity between shapes which exhibit invariance to different deformations and or poses of the objects represented

structure, that is, shapes are viewed as point sets. The notion of distance compares the full structure of the shapes, as opposed to simple features. Two shapes will be declared *equal* if and only if they are isometric. This means that the invariance property encoded by the metrics one chooses to endow the shapes with. For example, if the shapes are endowed with Euclidean met-



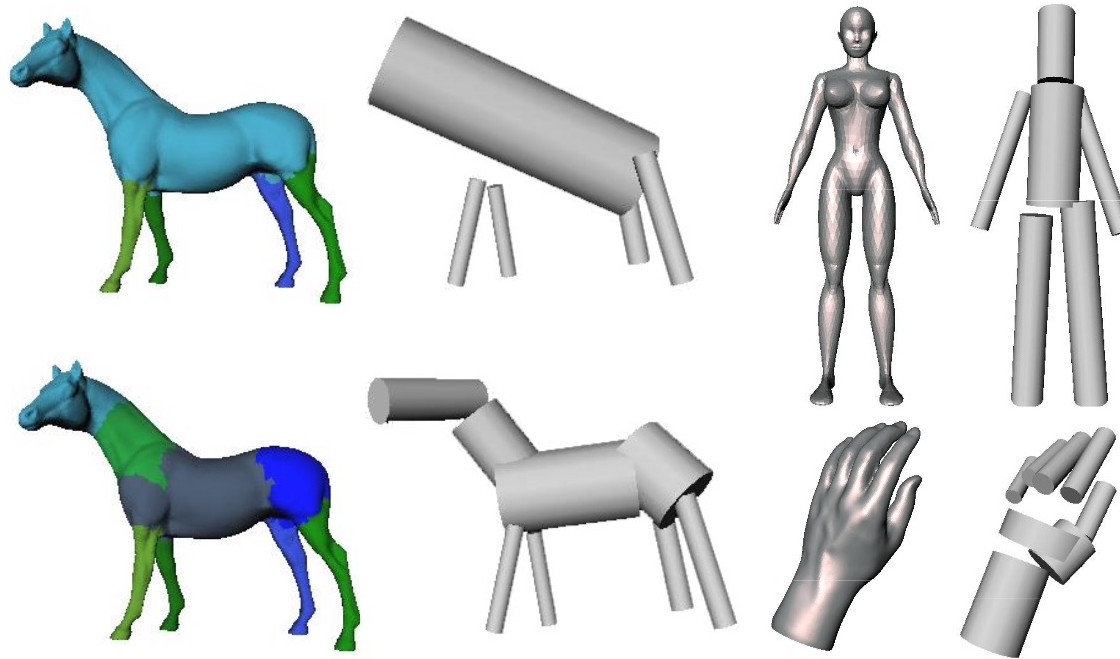
# Agglomerative Clustering



[https://upload.wikimedia.org/wikipedia/commons/a/ad/Hierarchical\\_clustering\\_simple\\_diagram.svg](https://upload.wikimedia.org/wikipedia/commons/a/ad/Hierarchical_clustering_simple_diagram.svg)

***Merge from the bottom up***

# Agglomerative Clustering in Geometry

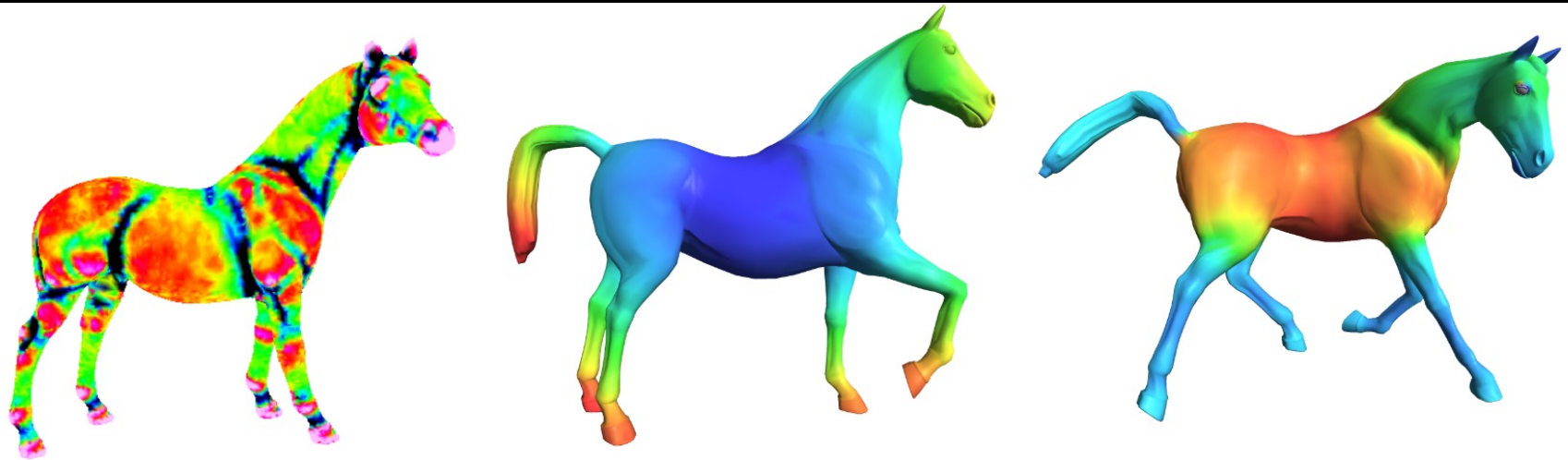


*“Hierarchical mesh segmentation based on fitting primitives.”*  
*Attene, Falcidieno, and Spagnuolo; The Visual Computer 2006*

***Fit a primitive and measure error***



# Typical Features

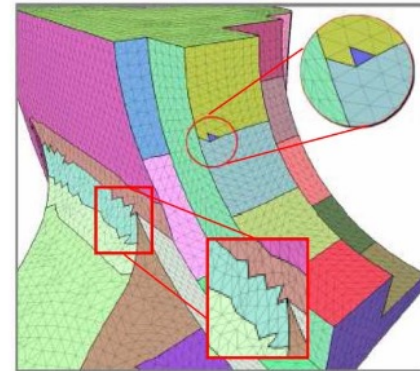


**Figure 4:** Example of mesh attributes used for partitioning. Left: minimum curvature, middle: average geodesic distance, right: shape diameter function.

**“Segmentation and Shape Extraction of 3D Boundary Meshes.”  
Shamir; EG STAR 2006.**

# Additional Desirable Properties

- Cardinality
  - Not too small and not too large or a given number (of segment or elements)
  - Overall balanced partition
- Geometry
  - Size: area, diameter, radius
  - Convexity, Roundness
  - Boundary smoothness
- Topology
  - Connectivity (single component)
  - Disk topology
  - a given number (of segment or elements)



***“Segmentation and Shape Extraction of 3D Boundary Meshes.”***

**Shamir; EG STAR 2006.**

*via Q. Huang, Stanford CS 468, 2012*

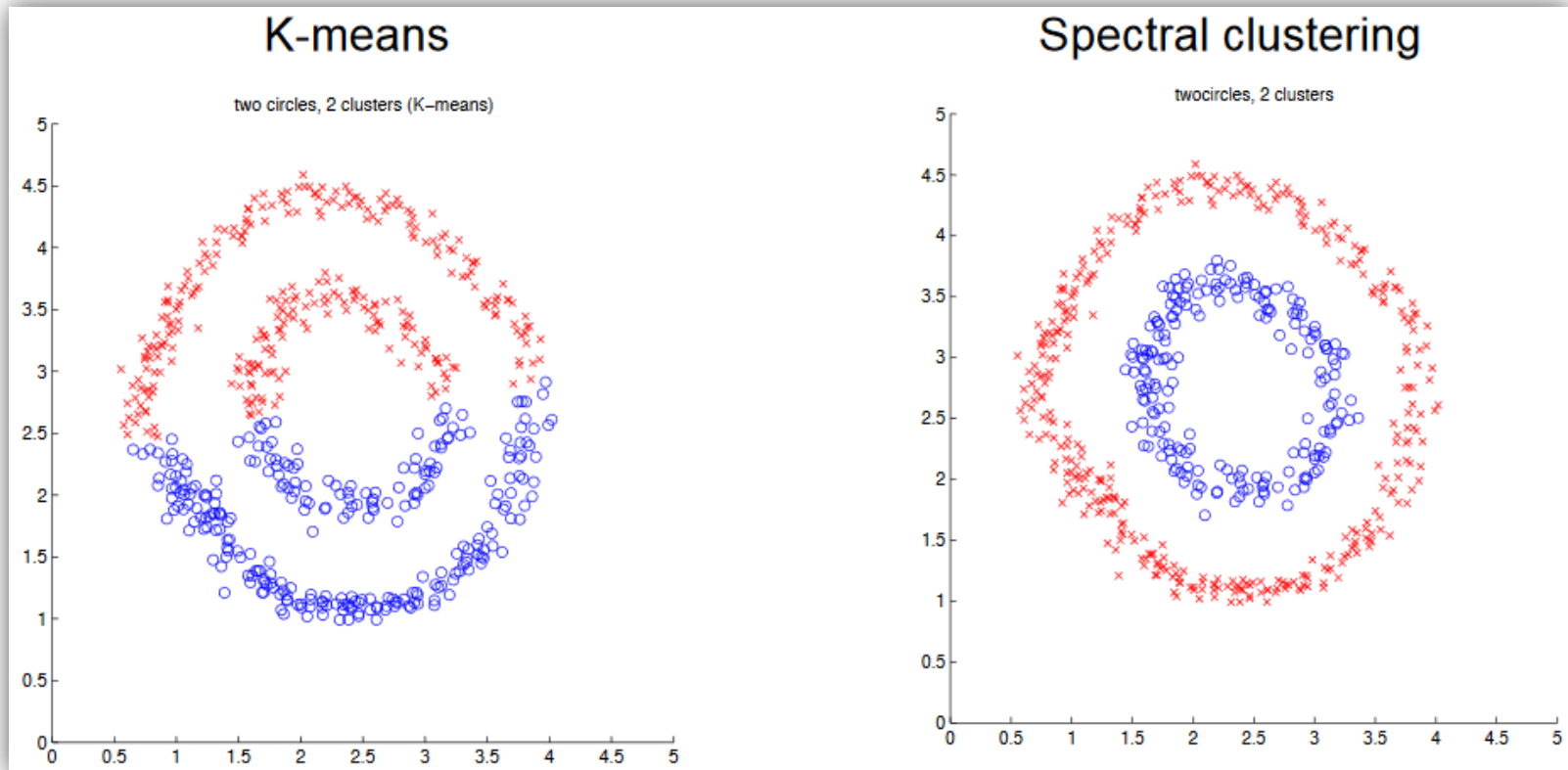
# Issue So Far

*No notion of optimality.*

*No use of  
local relationships.*

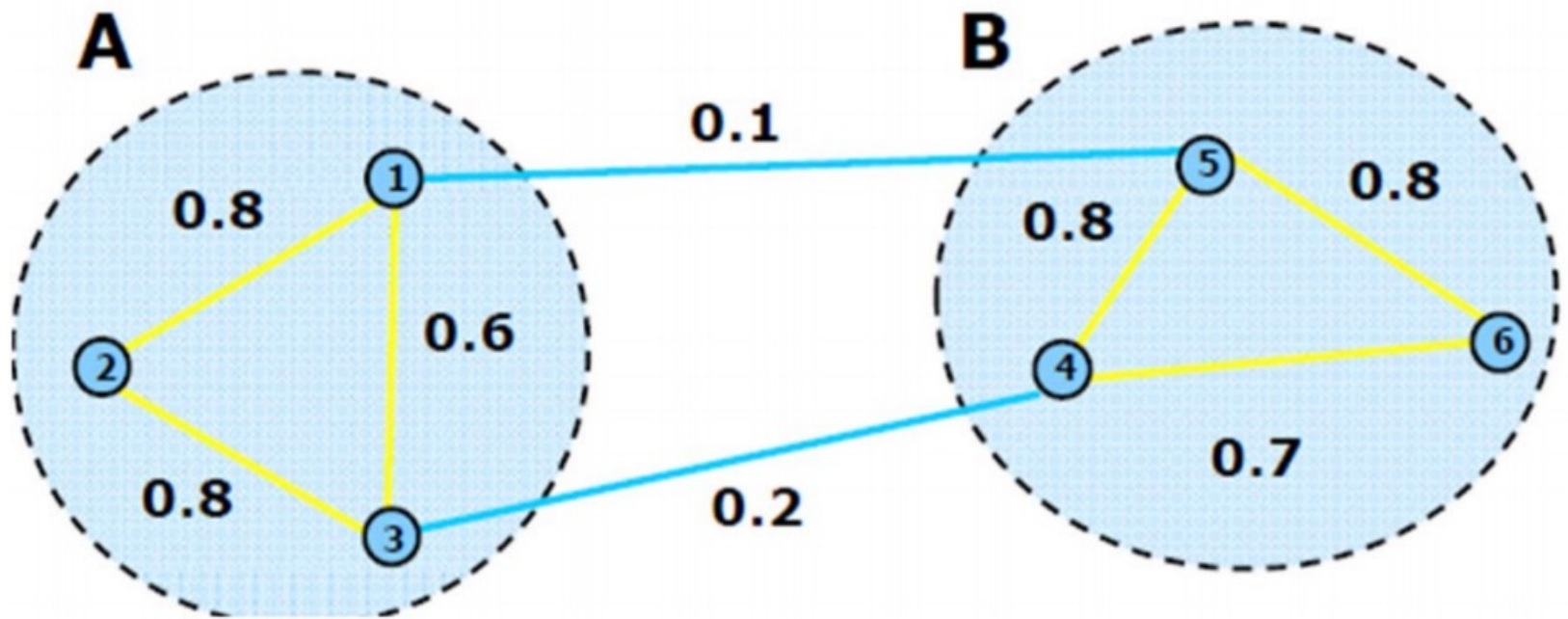
# Spectral Clustering

<http://cs.nyu.edu/~dsontag/courses/ml13/slides/lecture16.pdf>



- Rough notion of **optimality**
- Assembles **local relationships**

# Normalized Cuts for Two Clusters



# Normalized Cuts for Two Clusters

Symmetric similarity matrix  $W$

$$\text{Cut score } C(A, B) := \sum_{\substack{i \in A \\ j \in B}} w_{ij}$$

$$\text{Volume } V(A) := \sum_{i \in A} \sum_j w_{ij}$$

Normalized cut score

$$N(A, B) := C(A, B)(V(A)^{-1} + V(B)^{-1})$$

**“Normalized Cuts and Image Segmentation.”**

**Shi and Malik; PAMI 2000**

# Normalized Cuts

$$x_i := \begin{cases} V(A)^{-1} & \text{if } i \in A \\ -V(B)^{-1} & \text{if } i \in B \end{cases}$$

**On the board:**

$$x^\top Lx = \sum_{\substack{i \in A \\ j \in B}} w_{ij} (V(A)^{-1} + V(B)^{-1})^2$$

$$x^\top Dx = V(A)^{-1} + V(B)^{-1}$$

$$N(A, B) = \frac{x^\top Lx}{x^\top Dx}$$

$$x^\top D\mathbf{1} = 0$$

# Eigenvalue Problem

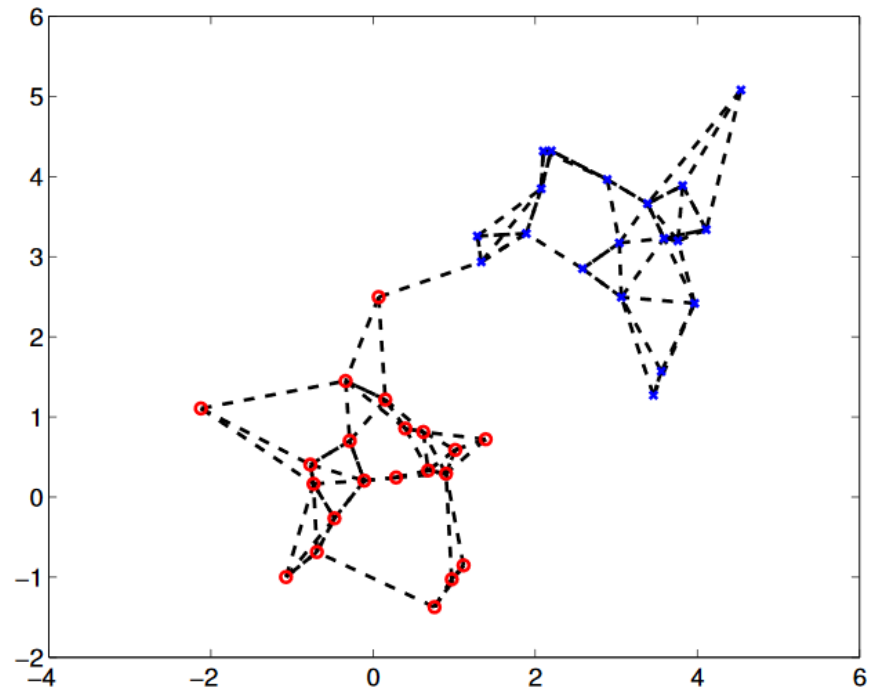
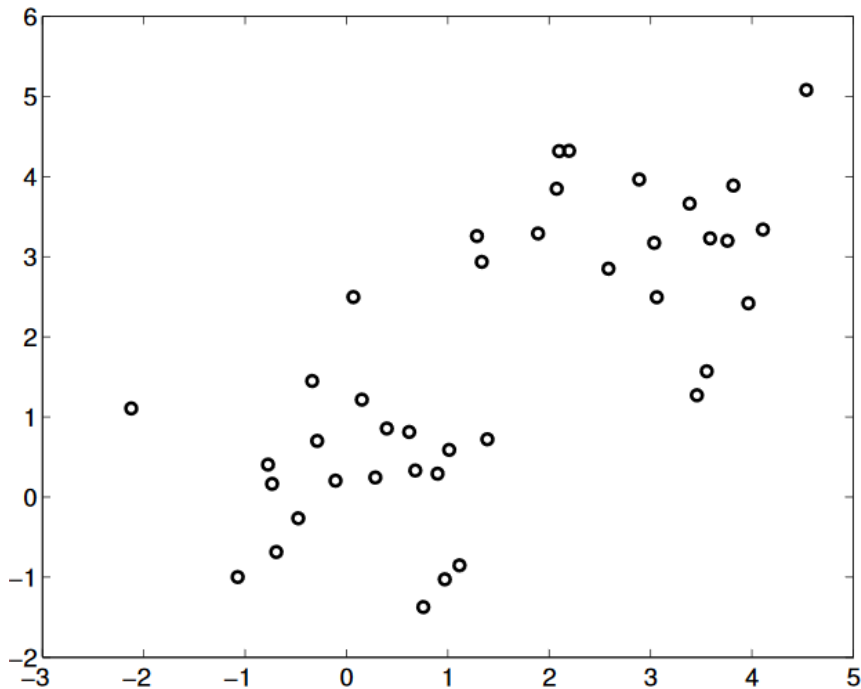
$$\begin{aligned} \min_x \quad & \frac{x^\top Lx}{x^\top Dx} \\ \text{s.t.} \quad & x^\top D\mathbf{1} = 0 \end{aligned}$$

*On the board:*

- **Relaxation of normalized cuts**
  - **Eigenvalue problem**



# Example on kNN Graph

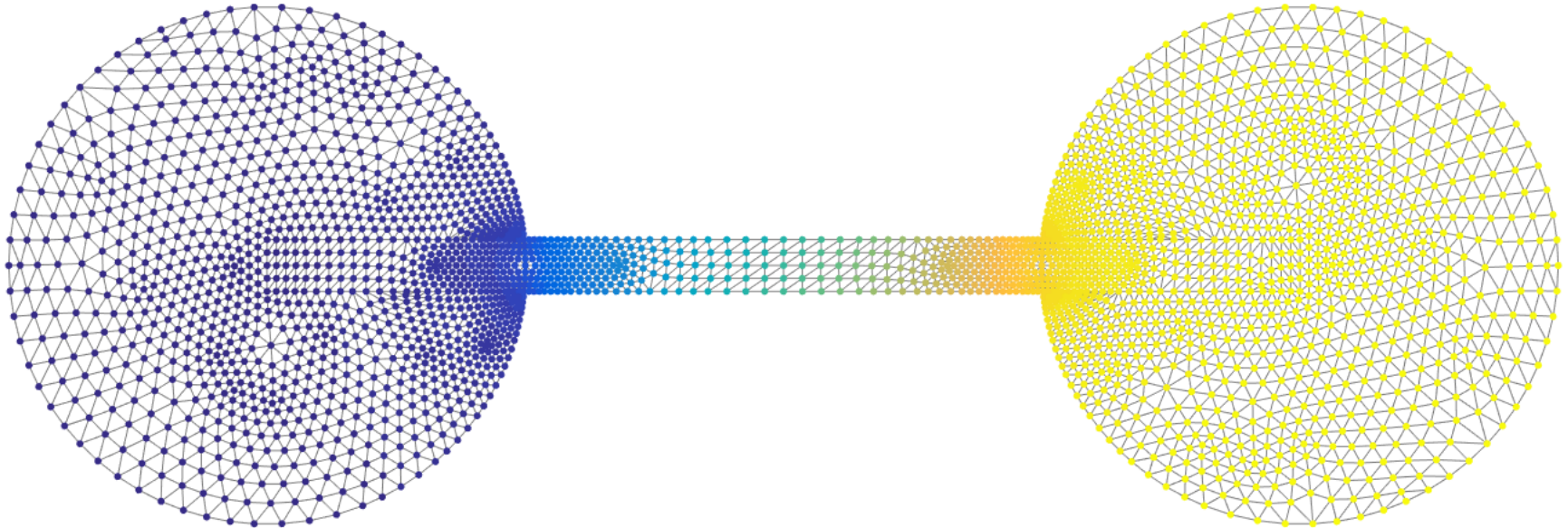


# For $\geq 2$ Clusters

- ***Recursive bi-partitioning*** (Hagen et al. 1991)
  - ***Analogy: Agglomerative clustering***
  - ***Potentially slow/unstable***
- ***Cluster multiple eigenvectors***
  - ***Analogy: k-means after dimension reduction***
  - ***More popular approach***

*Recall:*

# Second-Smallest Eigenvector



$$Lx = \lambda x$$

**Used for  
graph  
partitioning**

**Fiedler vector (“algebraic connectivity”)**

# Back to the Laplacian

Computers & Graphics 33 (2009) 381–390



Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: [www.elsevier.com/locate/cag](http://www.elsevier.com/locate/cag)



Technical Section

## Discrete Laplace–Beltrami operators for shape analysis and segmentation

Martin Reuter<sup>a,b</sup>, Silvia Biasotti<sup>c,\*</sup>, Daniela Giorgi<sup>c</sup>, Giuseppe Patanè<sup>c</sup>, Michela Spagnuolo<sup>c</sup>

<sup>a</sup> Massachusetts Institute of Technology, Cambridge, MA, USA

<sup>b</sup> A.A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital, Harvard Medical School, Boston, MA, USA

<sup>c</sup> Istituto di Matematica Applicata e Tecnologie Informatiche - Consiglio Nazionale delle Ricerche, Genova, Italy

### ARTICLE INFO

#### Article history:

Received 12 December 2008

Received in revised form

2 March 2009

Accepted 3 March 2009

#### Keywords:

Laplace–Beltrami operator

Eigenfunctions

Nodal sets

Nodal domains

Shape analysis

Shape segmentation

### ABSTRACT

Shape analysis plays a pivotal role in a large number of applications, ranging from traditional geometry processing to more recent 3D content management. In this scenario, spectral methods are extremely promising as they provide a natural library of tools for shape analysis, intrinsically defined by the shape itself. In particular, the eigenfunctions of the Laplace–Beltrami operator yield a set of real-valued functions that provide interesting insights in the structure and morphology of the shape. In this paper, we first analyze different discretizations of the Laplace–Beltrami operator (geometric Laplacians, linear and cubic FEM operators) in terms of the correctness of their eigenfunctions with respect to the continuous case. We then present the family of segmentations induced by the nodal sets of the

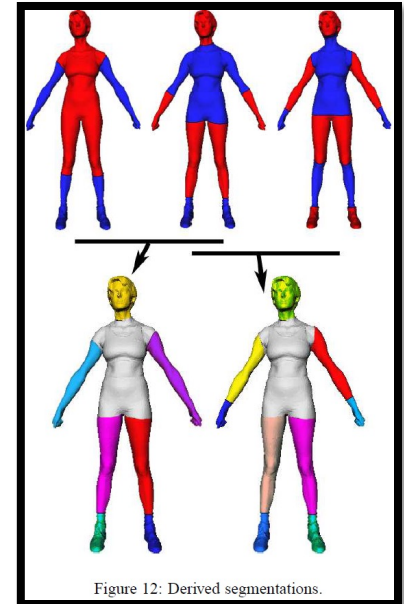


Figure 12: Derived segmentations.

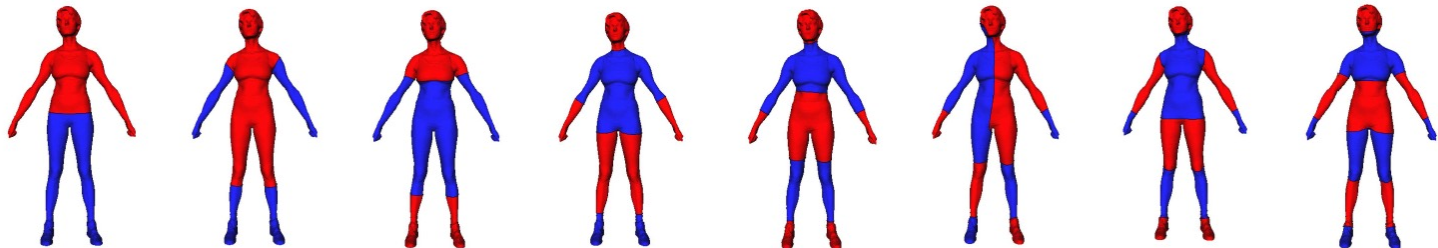


Figure 7: Segmentations induced by the nodal domains of some eigenfunctions selected among the first 15 eigenfunctions (in order of increasing eigenvalues). Blue regions correspond to regions where the eigenfunctions have negative values, while red regions correspond to positive values.

### 1. Introduction

Shape analysis and reasoning on point clouds is a large number of applications, ranging from traditional geometry processing to more recent 3D content management.

In the recent years, there has been a growing need to analyze and manage 3D content in many emerging web-based applications. A semantic analysis and segmentation able to capture a varied set of

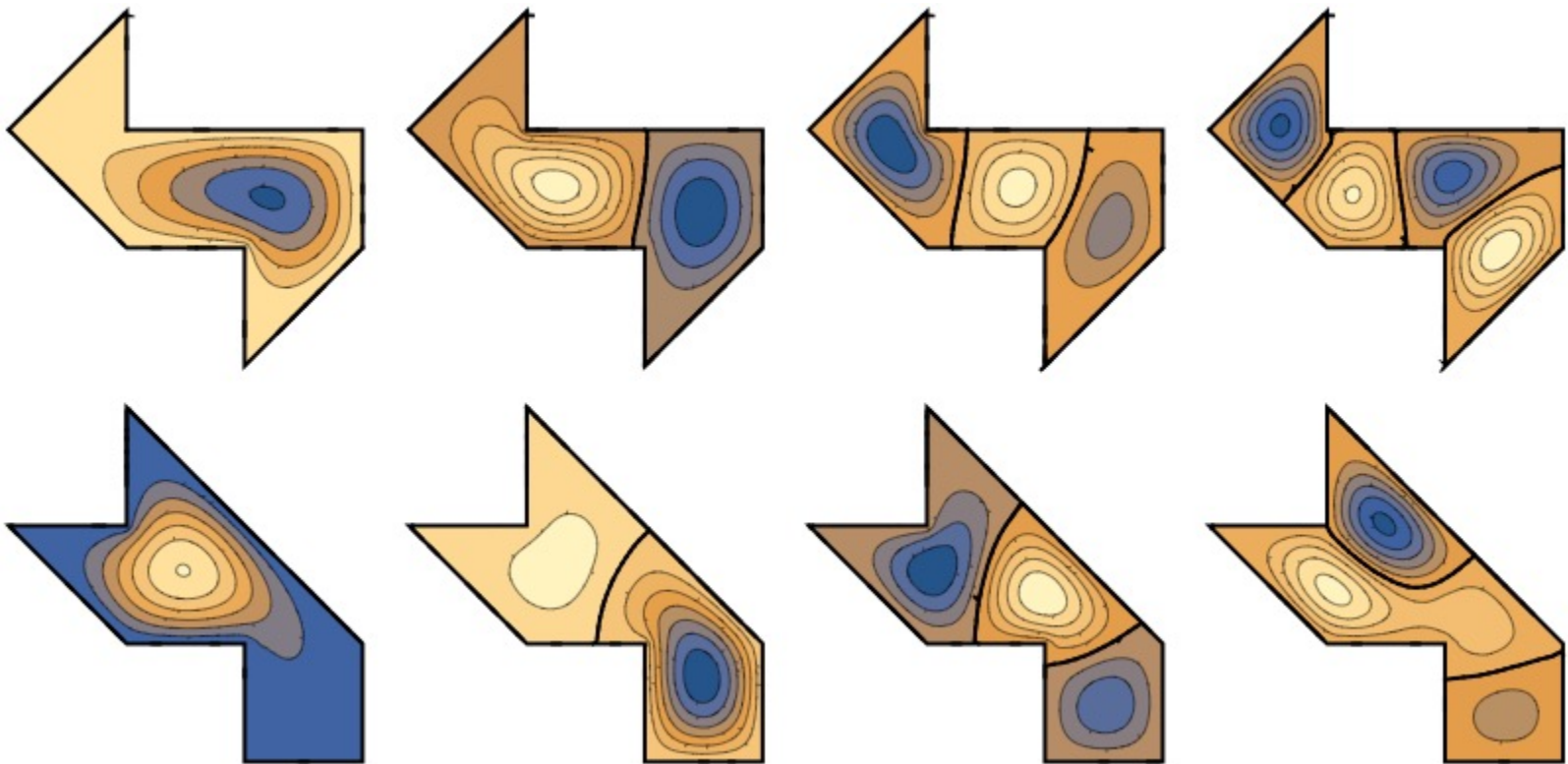
content in many emerging web-based applications. A semantic analysis and segmentation able to capture a varied set of

# ***Nodal domain***

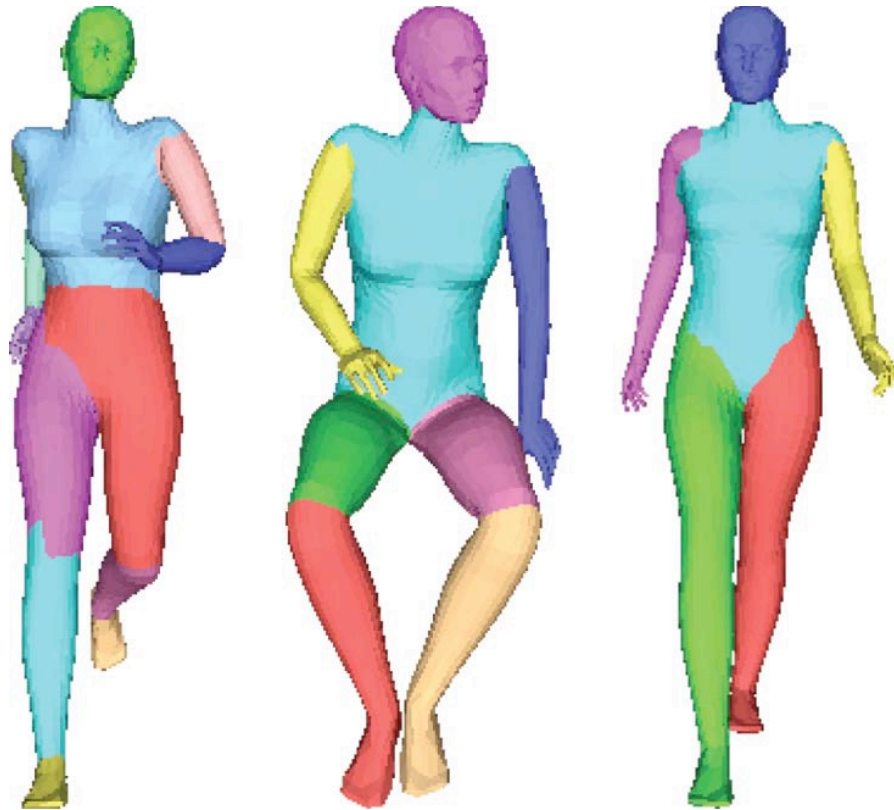
A connected region  
where a Laplacian  
eigenfunction has  
constant sign

# Courant's Theorem

**The  $k$ -th Laplacian eigenfunction has  
at most  $k$  nodal domains.**



# Issue



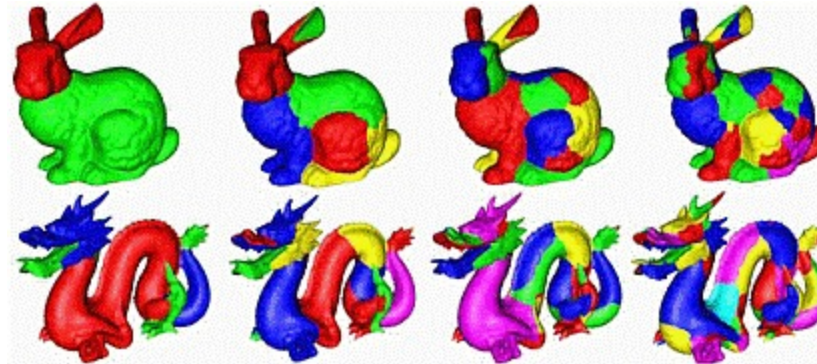
*From  
randomized  
cuts*

Image courtesy Q. Huang

***Inconsistent!***

# What's meaningful?

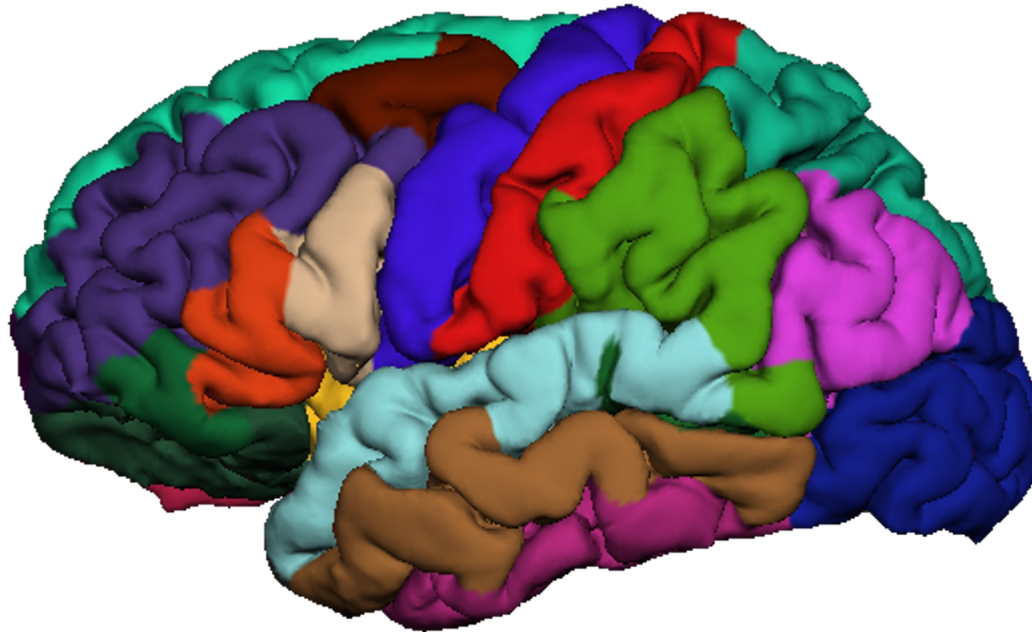
- ***Semantically?***
- Flat/Developable?
  - Convex?
  - Voronoi-like?





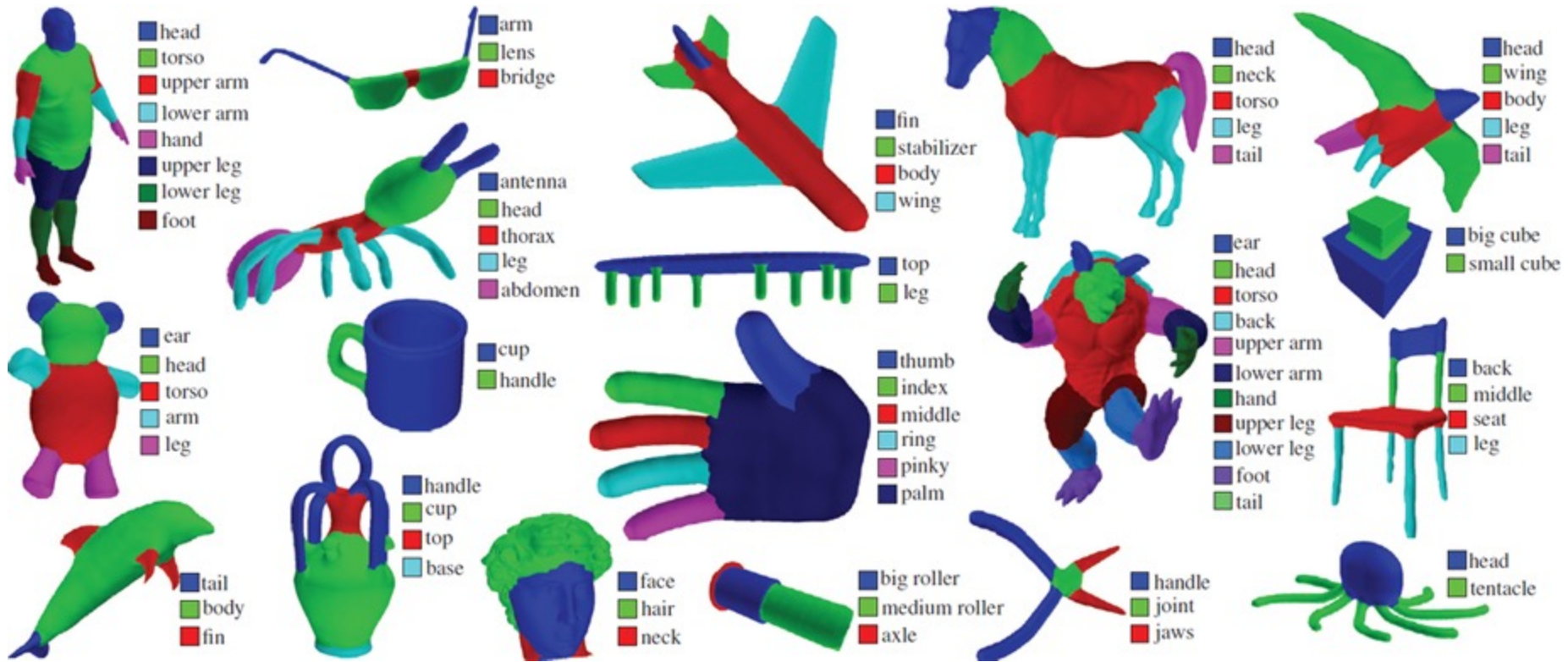
# Obvious Counterexample

<http://www.erflow.eu/brain-segmentation-science-case>



***Shape provides only a clue***

# Supervised Learning

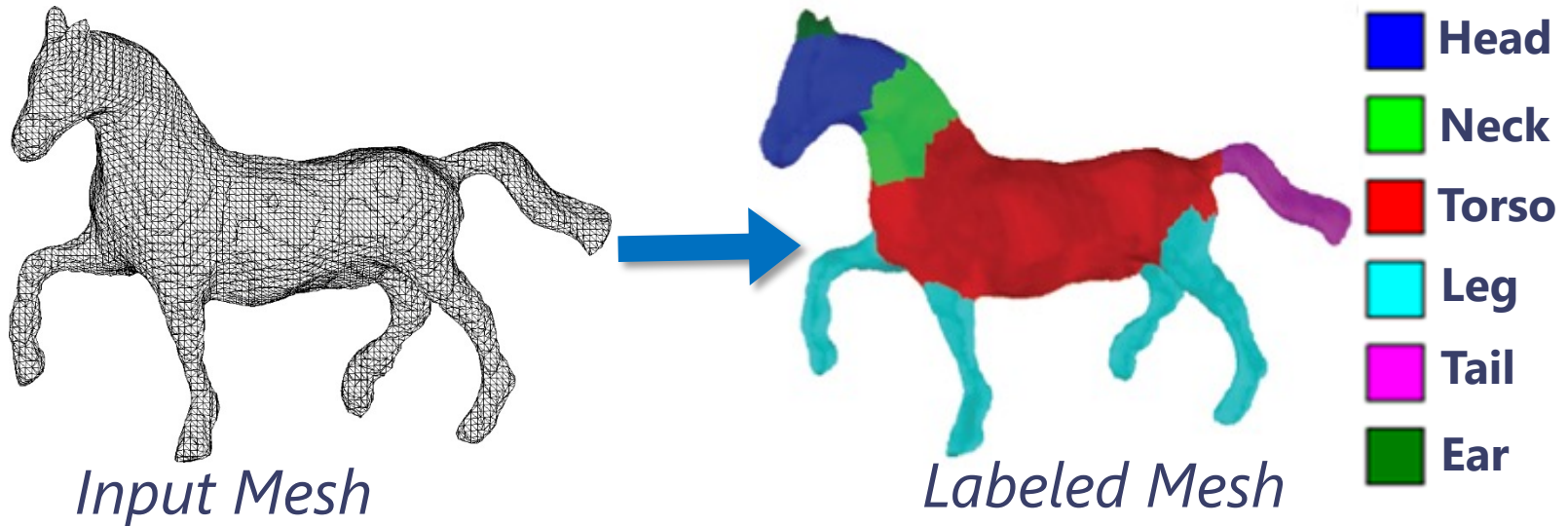


*“Learning 3D Mesh Segmentation and Labeling.”*

*Kalogerakis, Hertzmann, and Singh; SIGGRAPH 2010*

**Use example data to help**

# Conditional Random Field



$$c^* := \arg \min_c \left[ \sum_i \alpha_i E_1(c_i; x_i) + \sum_{ij} \ell_{ij} E_2(c_i, c_j; y_{ij}) \right]$$

**Unary descriptor term**

**Binary label compatibility term**

# Before Someone Asks

## 3D Shape Segmentation with Projective Convolutional Networks

Evangelos Kalogerakis<sup>1</sup>

Melinos Averkiou<sup>2</sup>

Subhransu Maji<sup>1</sup>

Siddhartha Chaudhuri<sup>3</sup>

<sup>1</sup>University of Massachusetts Amherst

<sup>2</sup>University of Cyprus

<sup>3</sup>IIT Bombay

### Abstract

*This paper introduces a deep architecture for segmenting 3D objects into their labeled semantic parts. Our architecture combines image-based Fully Convolutional Networks (FCNs) and surface-based Conditional Random Fields (CRFs) to yield coherent segmentations of 3D shapes. The image-based FCNs are used for efficient view-based reasoning about 3D object parts. Through a special projection layer, FCN outputs are effectively aggregated across multiple views and scales, then are projected onto the 3D object surfaces. Finally, a surface-based CRF combines the projected outputs with geometric consistency cues to yield coherent segmentations. The whole architecture (multi-view FCNs and CRF) is trained end-to-end. Our approach significantly outperforms the existing state-of-the-art methods in the currently largest segmentation benchmark (ShapeNet). Finally, we demonstrate promising segmentation results on noisy 3D shapes acquired from consumer-grade depth cameras.*

### 1. Introduction

In recent years there has been an explosion of 3D shape data on the web. In addition to the increasing number of community-curated CAD models, depth sensors deployed on a wide range of platforms are able to acquire 3D geometric representations of objects in the form of polygon

The shape segmentation task, while fundamental, is challenging because of the variety and ambiguity of shape parts that must be assigned the same semantic label; because accurately detecting boundaries between parts can involve extremely subtle cues; because local and global features must be jointly examined; and because the analysis must be robust to noise and undersampling.

We propose a deep architecture for segmenting and labeling 3D shapes that simply and effectively addresses these challenges, and significantly outperforms prior methods. The key insights of our technique are to repurpose image-based deep networks for view-based reasoning, and aggregate their outputs onto the surface representation of the shape in a geometrically consistent manner. We make no geometric or topological assumptions about the shape, nor exploit any hand-tuned geometric descriptors.

Our view-based approach is motivated by the success of deep networks on image segmentation tasks. Using rendered shapes lets us initialize our network with layers that have been trained on large image datasets, allowing better generalization. Since images depict shapes of photographed objects (along with texture), we expect such pre-trained layers to already encode some information about parts and their relationships. Recent work on view-based 3D shape classification [43, 35] and RGB-D recognition [13, 42] have shown the benefits of transferring learned representations

arXiv:1612.02808v1 [cs.CV] 8 Dec 2016

CVPR 2017