Clustering and Segmentation

Justin Solomon

6.838: Shape Analysis

Spring 2021

A Confusing Distinction

For "Customer Data and Engagement:"

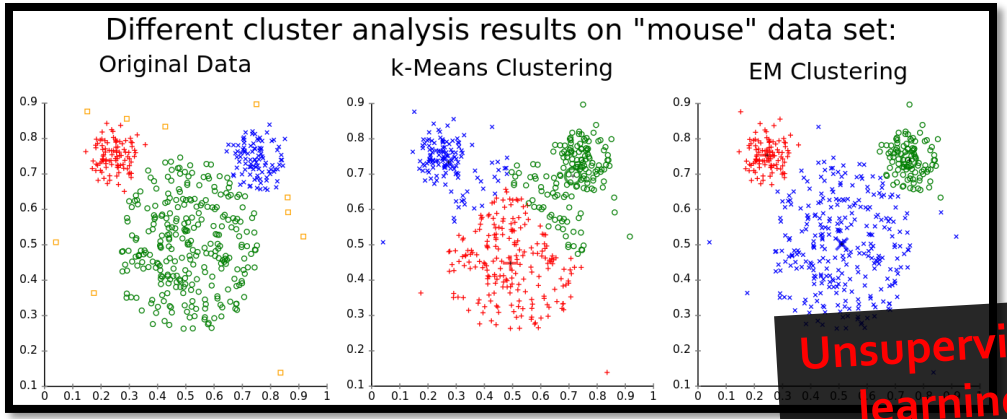
"**Segmenting** is the process of putting customers into groups based on similarities, and **clustering** is the process of finding similarities in customers so that they can be grouped, and therefore **segmented**."

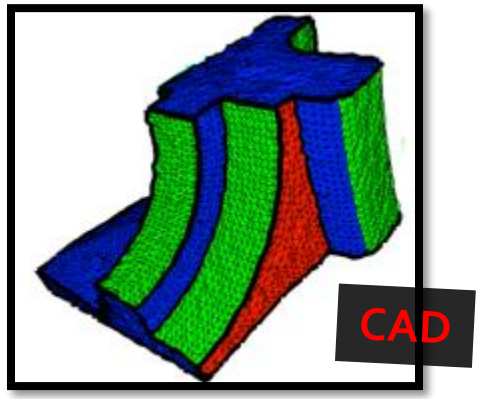
http://www2.agilone.com/blog/blog/segmentation-vs-clustering

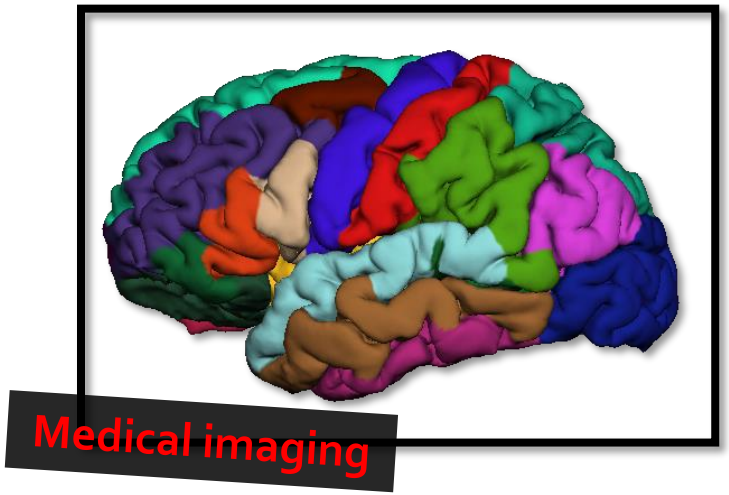
Our Objective

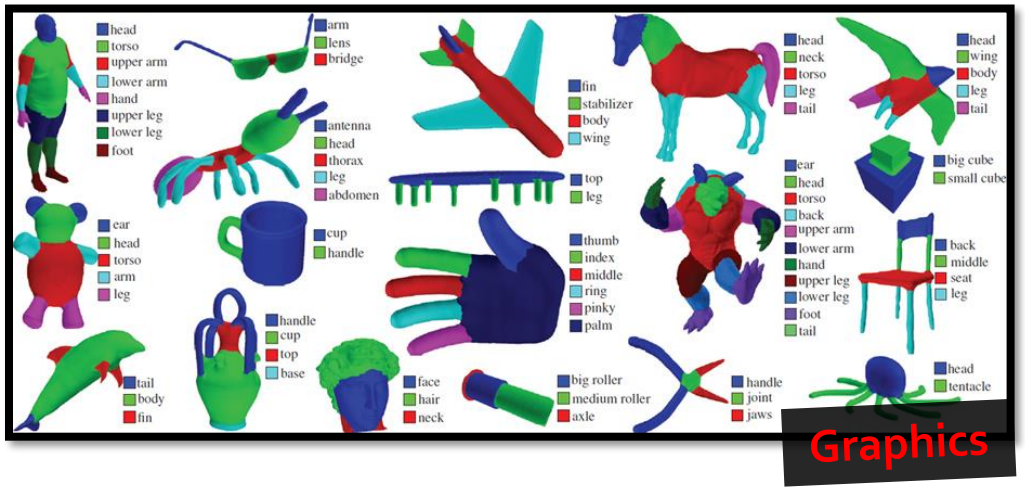
Divide a **geometric domain** into useful pieces.

Many Applications

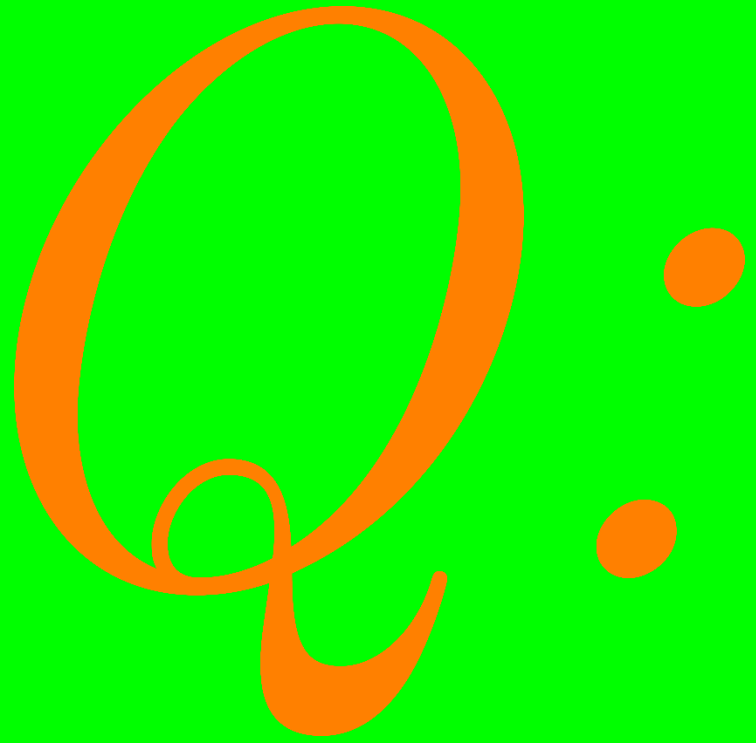








Images from Wikipedia, Christian Wolf (INSA de Lyon/LIRIS), Vangelis Kalogerakis



What is a good segmentation?

What is a Good Segmentation?

Application dependent!

- **Not an end in itself**
- **Unsolicited advice:**
  Be suspicious of general methods!

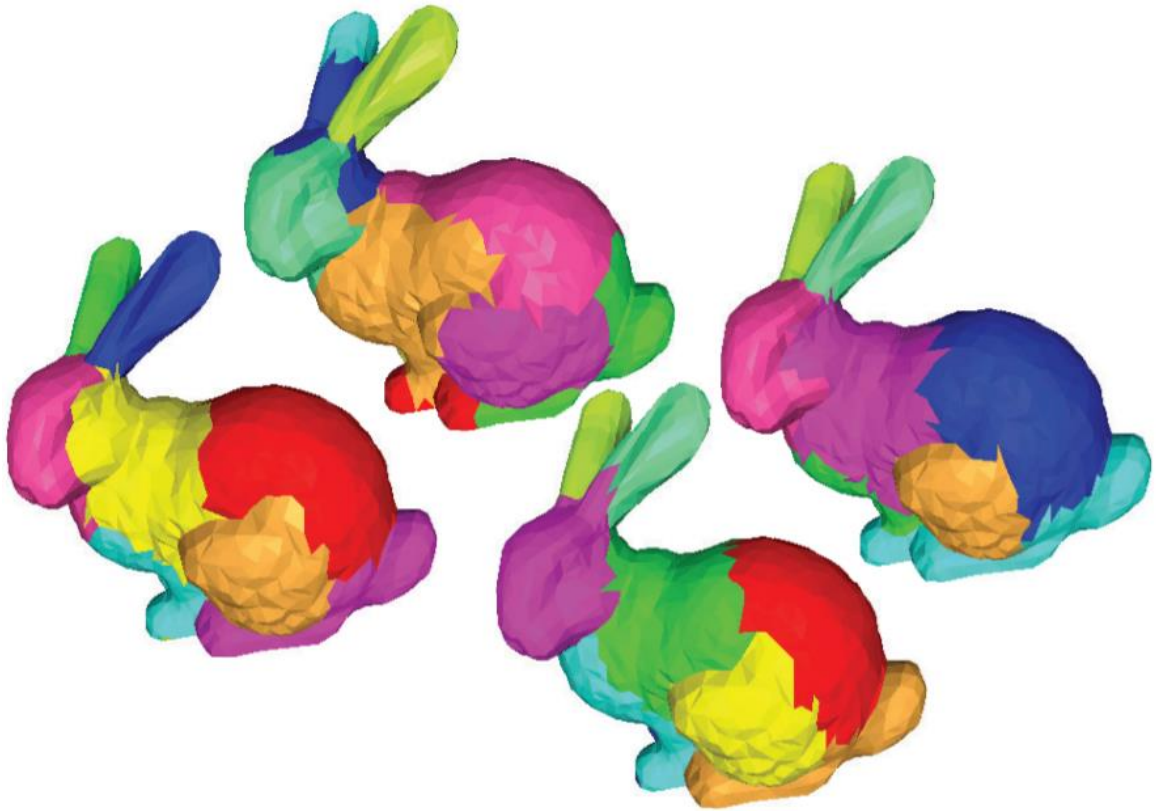


Image from "Randomized Cuts for 3D Mesh Analysis (Golovinskiy and Funkhouser)

According to Facebook

Aaron Hertzmann

October 9, 2018 ·

One bit of sloppy writing that has permeated the computer graphics and vision literature is the use of the word "semantic." Here's why I think that you should avoid using it, or, at least, use it very carefully.

"Semantic" is a pretentious weasel-word. The word "semantic" is used in a way that means almost nothing, which is ironic. However, it sounds like it's implying some sort of insight about AI or human intelligence. I think that researchers use it when they want to indicate that there's some high-level knowledge or context involved, but they're too lazy to be concrete about it.

Instead of using the word "semantic," I suggest thinking more concretely about what you really mean, and saying that instead. You will probably find that your paper is clearer. It's a bit of extra work, but clear writing takes work.

As an example, our SIGGRAPH 2010 paper was the first paper to apply learned "semantic labeling" to 3D surfaces. I insisted that we avoid using the word "semantic" as much as possible. Instead, we wrote that our method learns to label object parts, such as "hand" or "wheel", and that the labels can be chosen by a user. Saying that we learn to apply these labels is much clearer than saying that our labeling is "semantic" or that we label "semantic parts", whatever that means.

Other typical uses (I am making these examples up) is to say "We let the user group regions based on semantic concerns" or "The video can be broken into parts based on semantics." What do these sentences add?

Doug DeCarlo first pointed this issue out to me about a decade ago. He pointed out that "semantics" is the study of meaning, like dictionary definitions; how the phrase "I like it" means something different from "It likes me." Objects in images do not have meanings in the same way. A hand or wheel does not have a meaning. He said that reading this usage of "semantic" was like "nails on a chalkboard," and now I feel that way too.

There's a descriptivist argument one can make: our community's language naturally evolves over time. However, this doesn't license arbitrary misuse of language; we shouldn't use "plus" to mean "minus". Misusing technical terms from other fields can cause lots of problems. Doug said that using "semantic" in this way makes you sound stupid to, say, a computational linguist who might review your grant proposal.

Whenever I see the word "semantic," I think that the author hasn't thought carefully about what they mean, and is only using this pretentious word because they think it sounds cool. Avoid being that person.

I do make one exception: for better or for worse, the term "semantic labeling" has come to mean a specific task in vision and in graphics. So I think it's fair to use the term in this case: this is the name of a task, and one must use shortcuts in names. The problem is that the word "semantic" is used all over in lots of other contexts where it means very little.

Fredo Durand, Alec Jacobson and 41 others 22 Comments

Like Comment Share

Many Attempts to Standardize

A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation

F. Peraz

A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms

Roberto Tron René Vidal
Center for Imaging Science, Johns Hopkins University

Over the
their funda
enabling ta
many fields
the evolutio
mance and
this work w
ation metho
tion. The da
Segmentati
sequences,
object segm
blur and ap
by densely
truth segme
sive analys
proaches u
the spatial
silhouette c
uncover str
opening up

1. Introd

Over the past few
ing a scene containin
been proposed. How
tested on a handful o
been often tested on
the comparison of dif
In this paper, we com
gorithms for affine c
sequences of checker

1. Introduction

Motion segmentat
step for several appl
surveillance, tracking
nineties, these applic
several 2-D motion
niques aimed to sepa
into different regions
For example, a video

visceral

Identification, Localisation, and Segmentation

SEVENTH FRAMEWORK PROGRAMME

Call for participation

While a growing number of be
compare the performance of
automated organ segmentati
detection in images with restricte
no efforts have been made
benchmarking these and related
automated **identification and s**
bones, inner organs and relevar
visible in an image volume of th
trunk or even the whole body.

In order to gauge the current st
automated whole-body image
compare different methodological
are organizing the "VISCERAL
Localisation and Segmentation E
this purpose, **we are making a**
dataset of clinical whole-body M

A Benchmark for 3D Mesh Segmentation

Xiaobai Chen, Aleksey Golovinskiy, Thomas Funkhouser
Princeton University

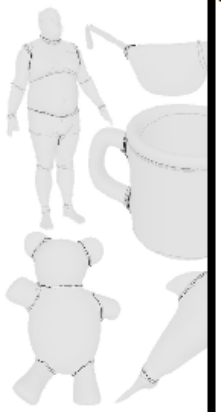


Figure 1: Composite imag

IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. 34, NO. 10, OCTOBER 2015 1993

The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)

Bjoern H. Menze*, A
Yuliya Burren, N
Marc-André Weber
Nicolas Cordier,
Christopher R. Du
Ben Glocker, P
Nigel M. John, End
Doina Precup, Steph
Lawrence Schwartz,

1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT 2019)

Performance Analysis and Benchmarking of Clustering Algorithms with gene datasets

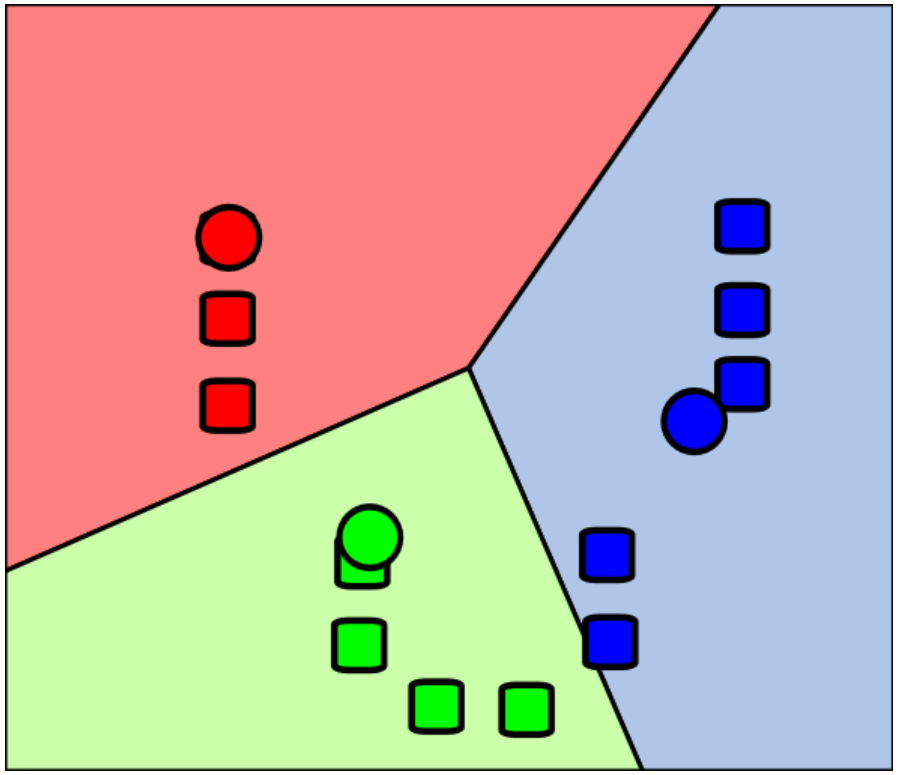
Our Approach

A few interesting geometric methods.

Geometry can come from:

- Embedding of a dataset
- Structure of a surface mesh
- Metric learning

Simplest Possible



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

https://upload.wikimedia.org/wikipedia/commons/d/d2/K_Means_Example_Step_4.svg

k-means clustering

Alternating Algorithm

$$\min_{S,\mu_i} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \mu_i\|_2^2$$

Initialization?

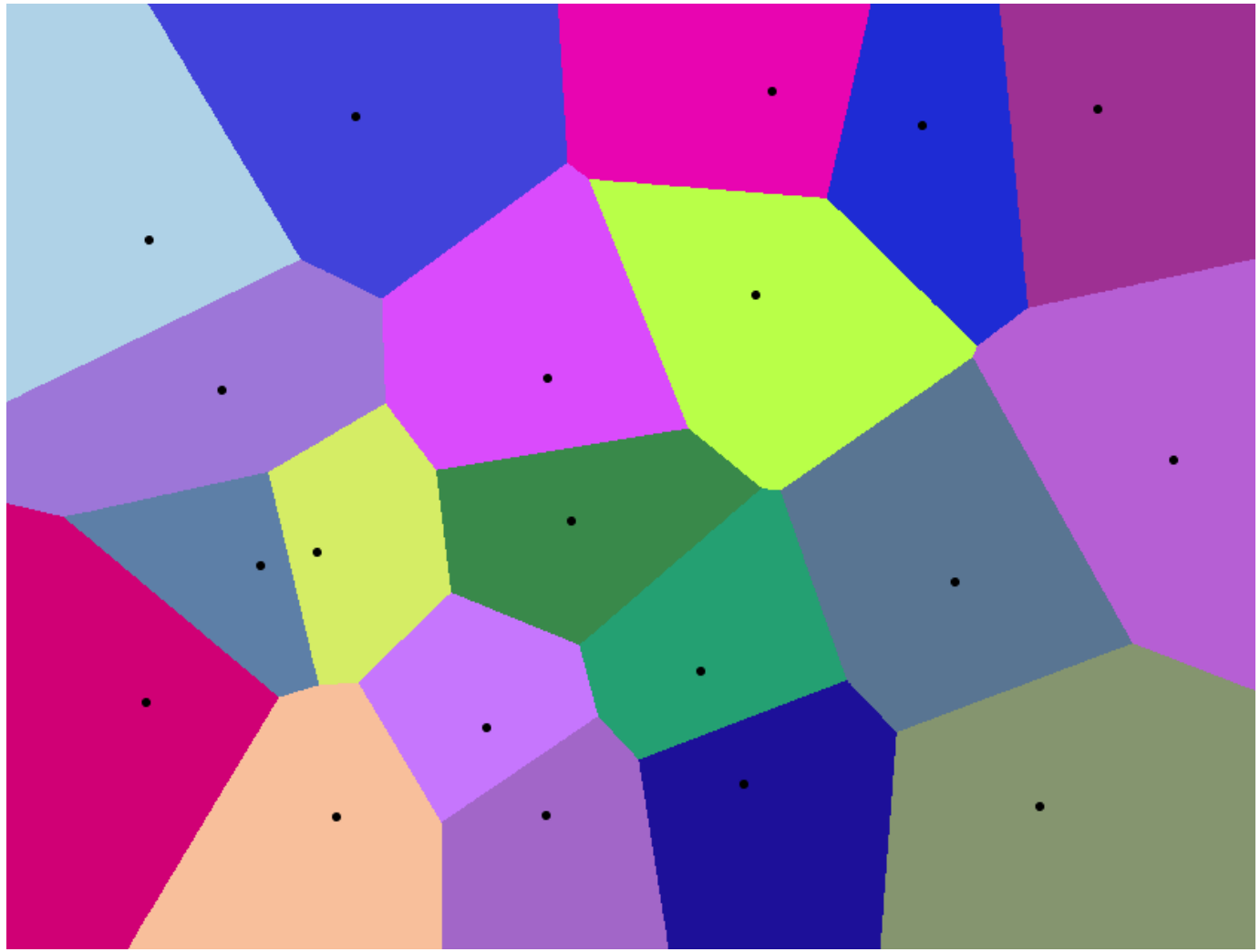
- **Assignment step (S): Voronoi cells**

$$S_i \leftarrow \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mu_i\|_2 \leq \|\mathbf{x} - \mu_j\|_2 \, \forall j \neq i\}$$

- **Update step (μ)**

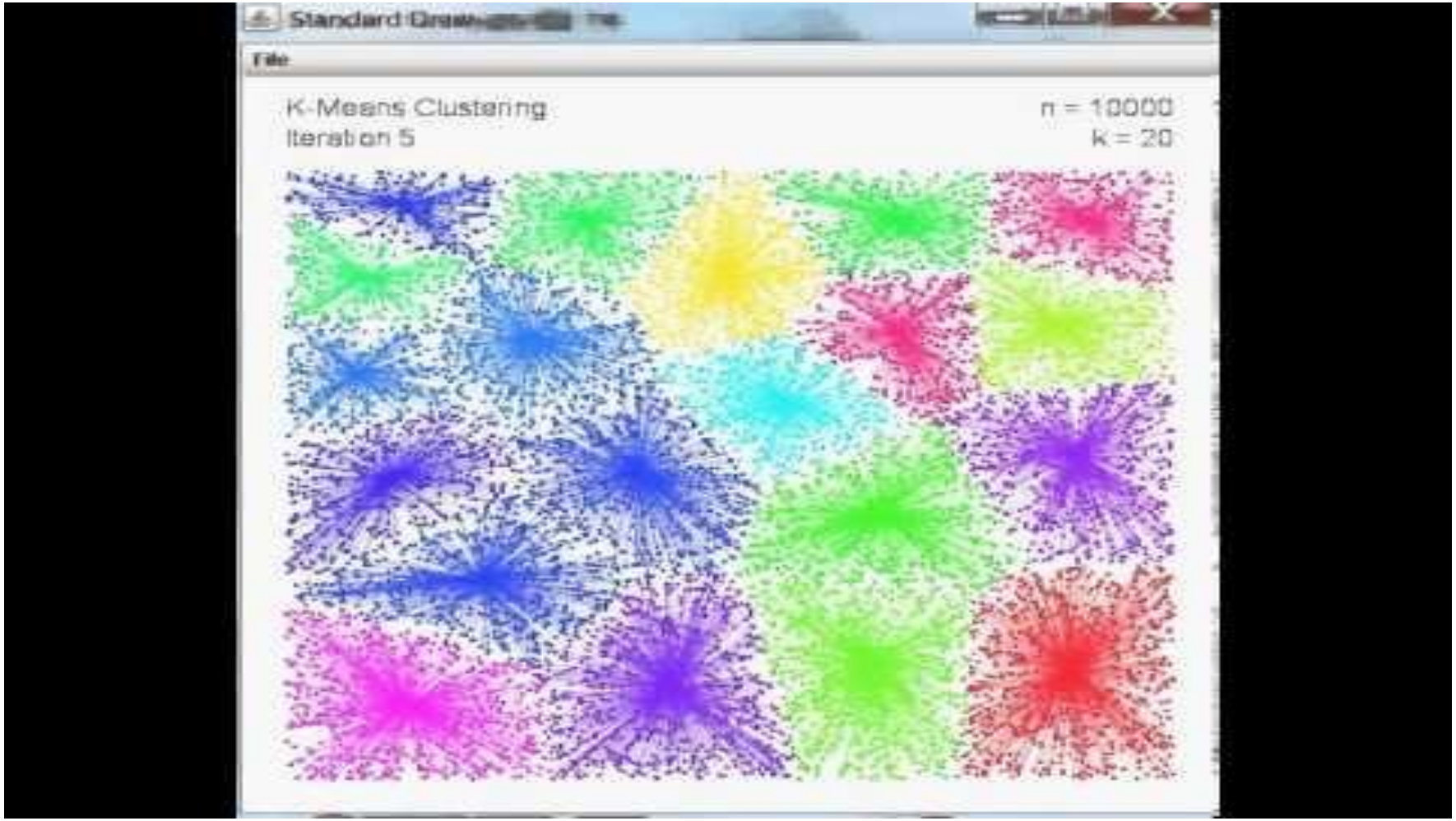
$$\mu_i \leftarrow \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x}$$

Voronoi Diagram



http://blog.alexbeutel.com/voronoi/v4.png

Example



K-Means++

1. Choose one center uniformly at random.
2. For each data point $\boldsymbol{x}$ not yet chosen, compute $D(\boldsymbol{x})$, the distance between x and the nearest center that has already been chosen.
3. Choose a new center randomly, with probability proportional to $D(\boldsymbol{x})^2$.
4. Repeat 1-3 until *k* centers are chosen.
5. Proceed with *k*-means clustering.

Theorem 1.1. *For any set of data points,* $E[\phi] \leq 8(\ln k + 2)\phi_{OPT}$.

k-means++: The Advantages of Careful Seeting (Arthur and Vassilvitskii 2007); pseudocode adapted from Wikipedia

Modern Seeding Algorithms: Many Options

Scalable K-Means++

Bahman Bahmani*†
Stanford University
Stanford, CA
bahman@stanford.edu

Benjamin Moseley*‡
University of Illinois
Urbana, IL
bmosele2@illinois.edu

Andrea Vattani*§
University of California
San Diego, CA
avattani@cs.ucsd.edu

Ravi Kumar
Yahoo! Research
Sunnyvale, CA
ravikumar@yahoo-inc.com

Sergei Vassilvitskii
Yahoo! Research
New York, NY
sergei@yahoo-inc.com

ABSTRACT

Over half a century old and showing no signs of aging, k-means remains one of the most popular data processing algorithms. As is well-known, a proper initialization of k-means is crucial for obtaining a good final solution. The recently proposed k-means++ initialization algorithm achieves this, obtaining an initial set of centers that is provably close to the optimum solution. A major downside of the k-means++ is its inherent sequential nature, which limits its applicability to massive data: one must make k passes over the data to find a good initial set of centers. In this work we show how to drastically reduce the number of passes needed to obtain, in parallel, a good initialization. This is unlike prevailing efforts on parallelizing k-means that have mostly focused on the post-initialization phases of k-means. We prove that our proposed initialization algorithm k-means|| obtains a nearly optimal solution after a logarithmic number of passes, and then show that in practice a constant number of passes suffices. Experimental evaluation on real-world large-scale data demonstrates that k-means|| outperforms k-means++ in both sequential and parallel settings.

1. INTRODUCTION

Clustering is a central problem in data management and

single method — k-means — remains the most popular clustering method; in fact, it was identified as one of the top 10 algorithms in data mining [34]. The advantage of k-means is its simplicity: starting with a set of randomly chosen initial centers, one repeatedly assigns each input point to its nearest center, and then recomputes the centers given the point assignment. This local search, called *Lloyd's* iteration, continues until the solution does not change between two consecutive rounds.

The k-means algorithm has maintained its popularity even as datasets have grown in size. Scaling k-means to massive data is relatively easy due to its simple iterative nature. Given a set of cluster centers, each point can independently decide which center is closest to it and, given an assignment of points to clusters, computing the optimum center can be done by simply averaging the points. Indeed parallel implementations of k-means are readily available (see, for example, cwiki.apache.org/MAHOUT/k-means-clustering.html).

From a theoretical standpoint, k-means is not a good clustering algorithm in terms of efficiency or quality: the running time can be exponential in the worst case [32, 4] and even though the final solution is locally optimal, it can be very far away from the global optimum (even under repeated random initializations). Nevertheless, in practice the speed and simplicity of k-means cannot be beat. Therefore, recent

Fast and Provably Good Seedings for k-Means

Olivier Bachem
Department of Computer Science
ETH Zurich
olivier.bachem@inf.ethz.ch

Mario Lucic
Department of Computer Science
ETH Zurich
lucic@inf.ethz.ch

S. Hamed Hassani
Department of Computer Science
ETH Zurich
hamed@inf.ethz.ch

Andreas Krause
Department of Computer Science
ETH Zurich
krausea@ethz.ch

Abstract

Seeding – the task of finding initial cluster centers – is critical in obtaining high-quality clusterings for k-Means. However, k-means++ seeding, the state of the art algorithm, does not scale well to massive datasets as it is inherently sequential and requires k full passes through the data. It was recently shown that Markov chain Monte Carlo sampling can be used to efficiently approximate the seeding step of k-means++. However, this result requires assumptions on the data generating distribution. We propose a simple yet fast seeding algorithm that produces *provably* good clusterings even *without assumptions* on the data. Our analysis shows that the algorithm allows for a favourable trade-off between solution quality and computational cost, speeding up k-means++ seeding by up to several orders of magnitude. We validate our theoretical results in extensive experiments on a

Issue: Choice of *k*

Informal intuition

"Elbow method:"

Look at percentage of variance explained by clusters (vague!)

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

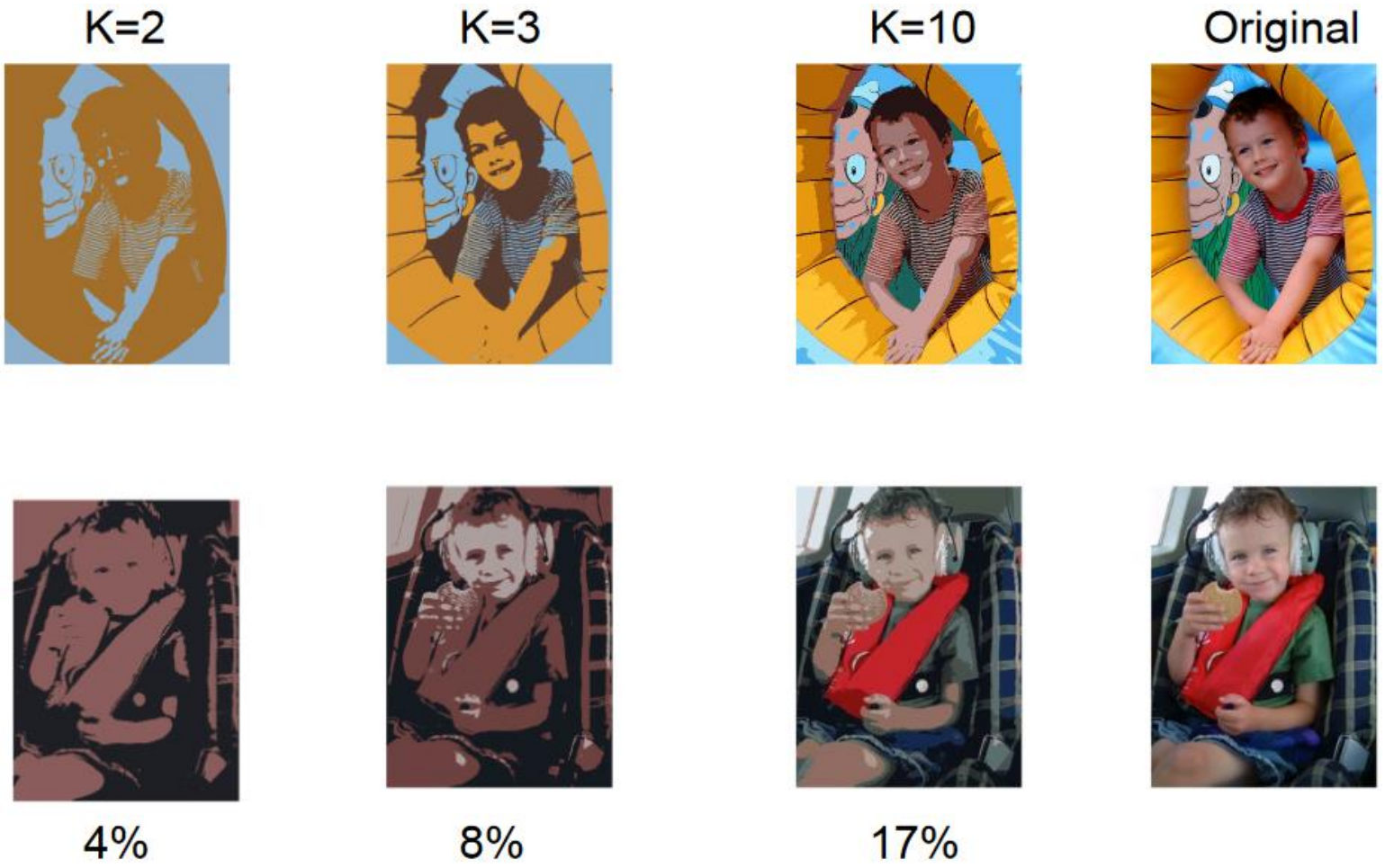
"Gap statistic"

$$D_i := \sum_{\mathbf{x},\mathbf{x}' \in S_i} \|\mathbf{x} - \mathbf{x}'\|_2^2$$

$$W_k := \sum_{i=1}^{k} \frac{1}{2n_i} D_i$$

$$\mathrm{Gap}_n(k) := \mathbb{E}_n^*[\log W_k] - \log W_k$$

Application to Color Space



http://cs.nyu.edu/~dsontag/courses/ml12/slides/lecture14.pdf

Can Apply to Features

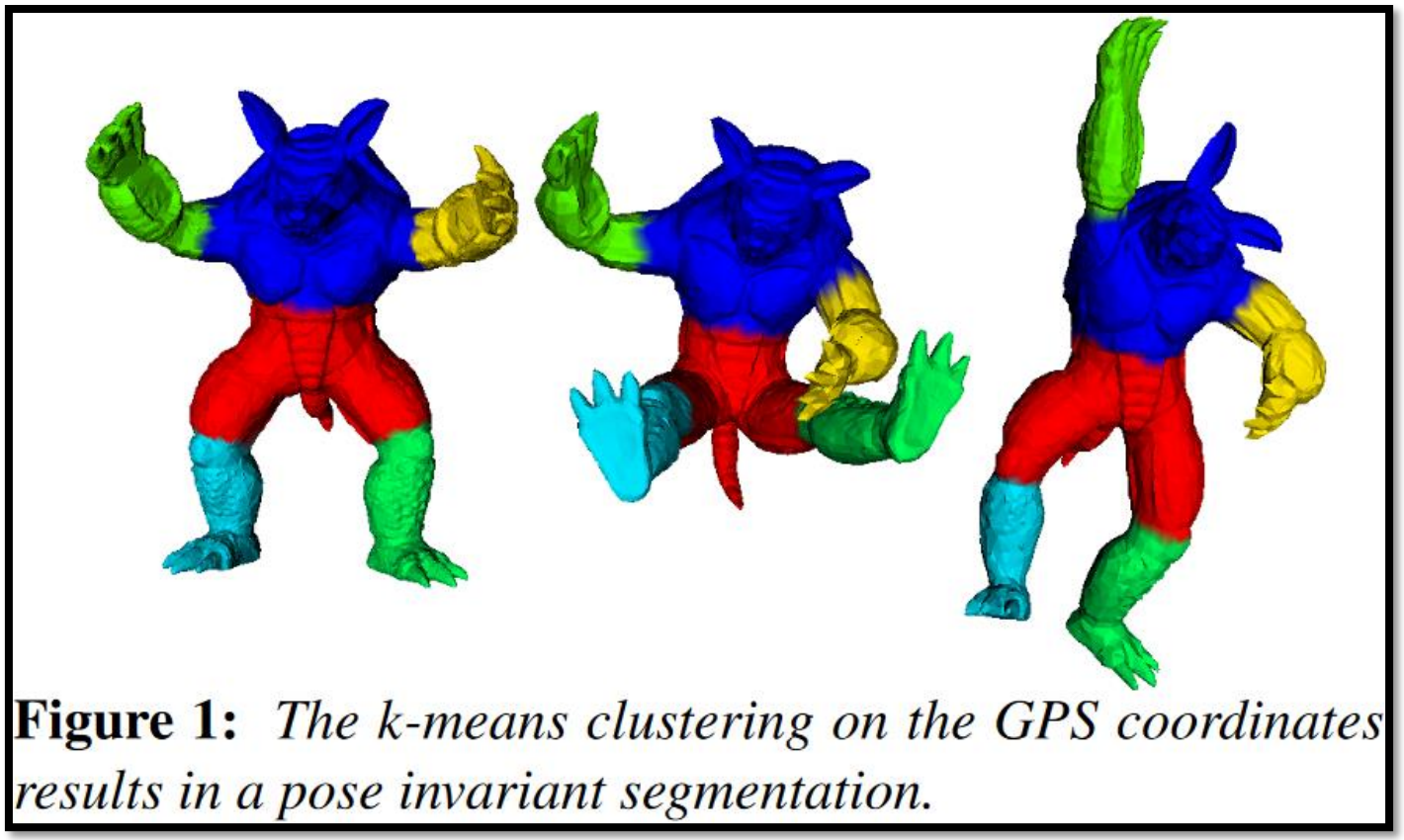
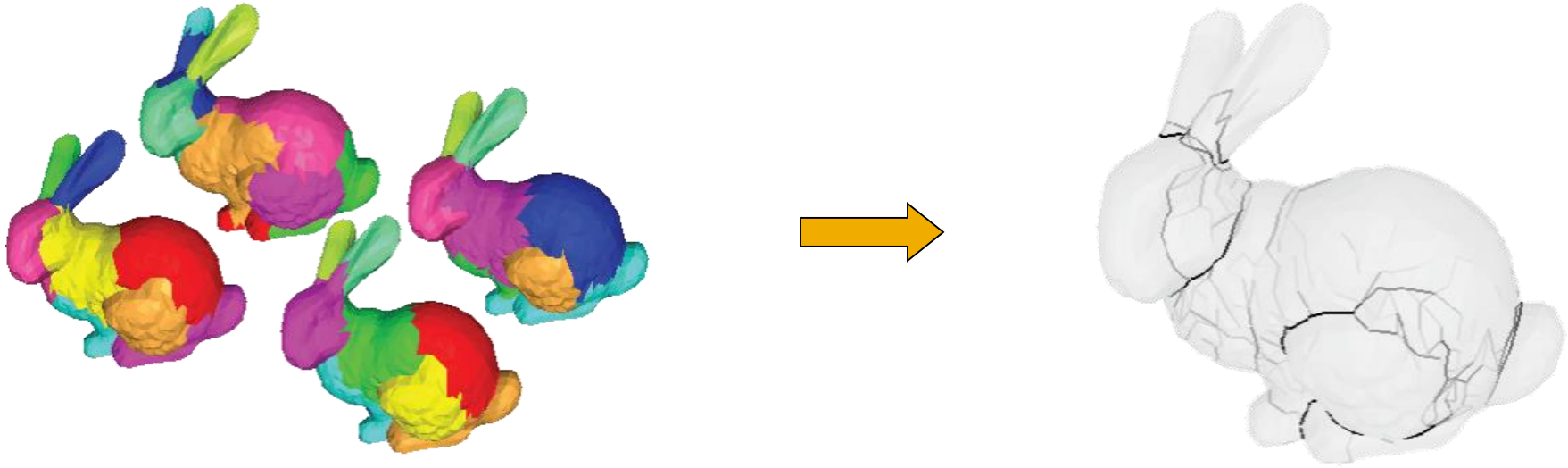


Figure 1: *The k-means clustering on the GPS coordinates results in a pose invariant segmentation.*

"Laplace-Beltrami Eigenfunctions for Deformation Invariant Shape Representation."

Rustamov; SGP 2007

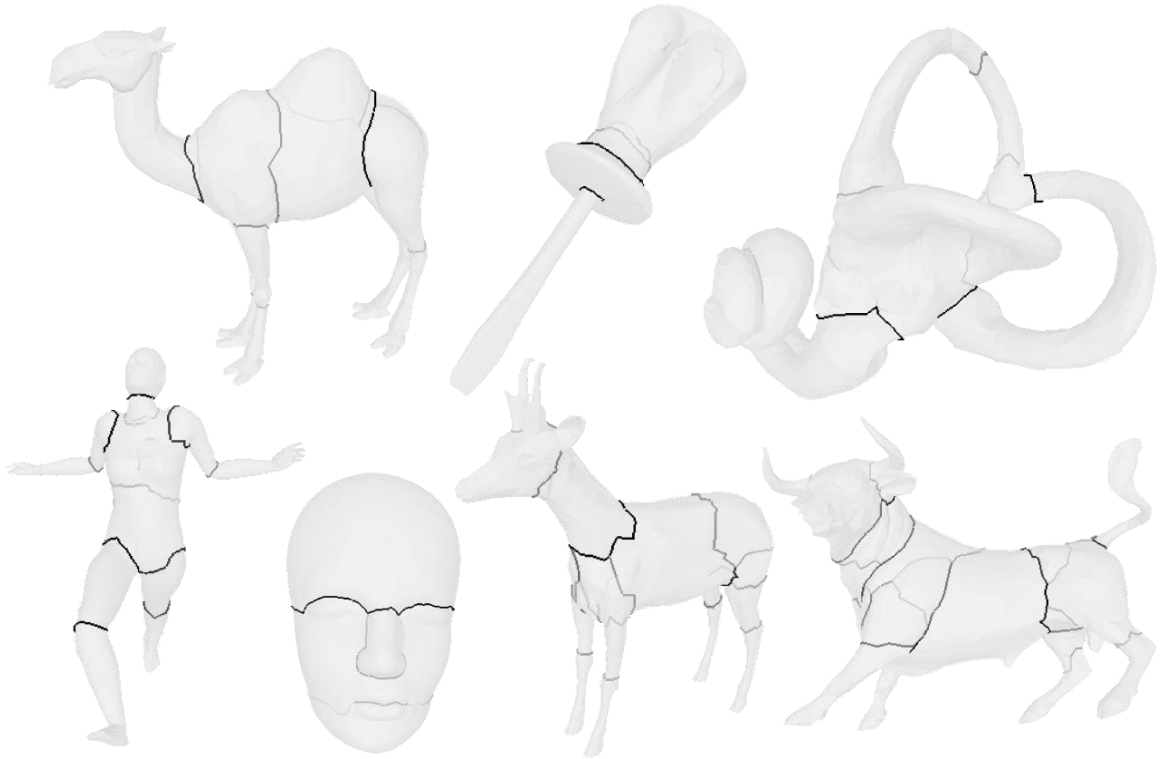
Dependence on Initial Guess



"Randomized Cuts for 3D Mesh Analysis."
Golovinskiy and Funkhouser; SIGGRAPH Asia 2008

Bug … or feature?

Dependence on Initial Guess



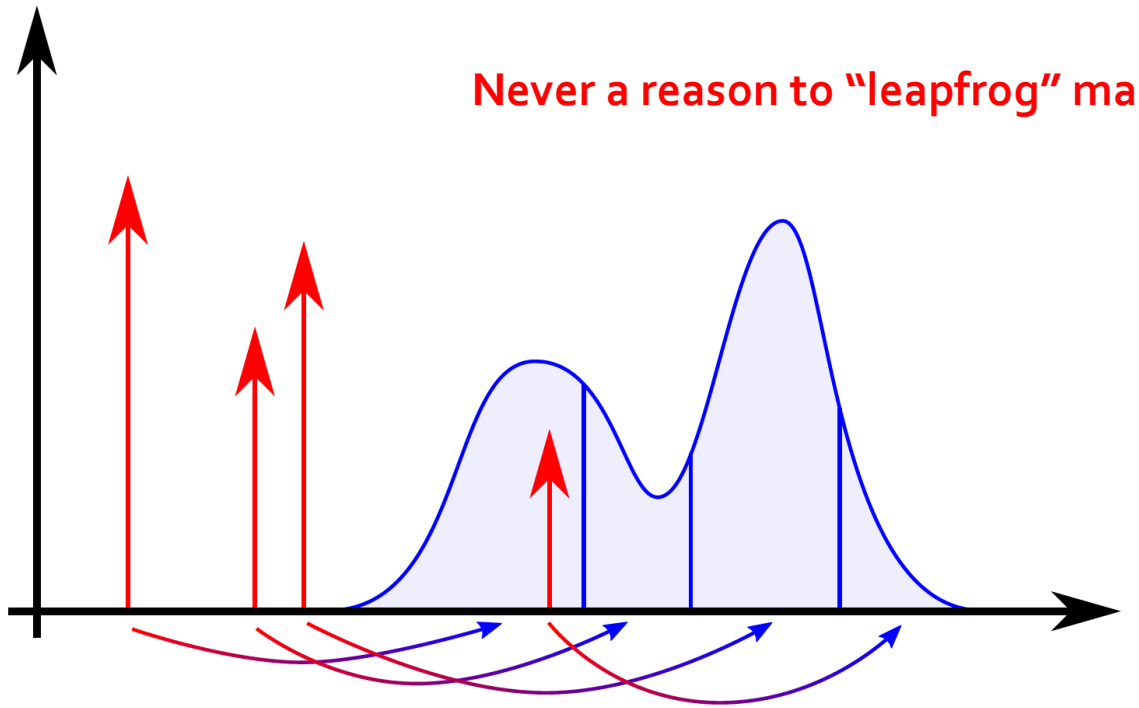
"Randomized Cuts for 3D Mesh Analysis."
Golovinskiy and Funkhouser; SIGGRAPH Asia 2008

Bug … or feature?

Link to past lecture:

Semidiscrete Transport

$$\mu_0 := \sum_{i=1}^{k_0} a_{0i}\delta_{\mathbf{x}_{0i}} \qquad \mu_1(S) := \int_S \rho_1(\mathbf{x})\, d\mathbf{x}$$



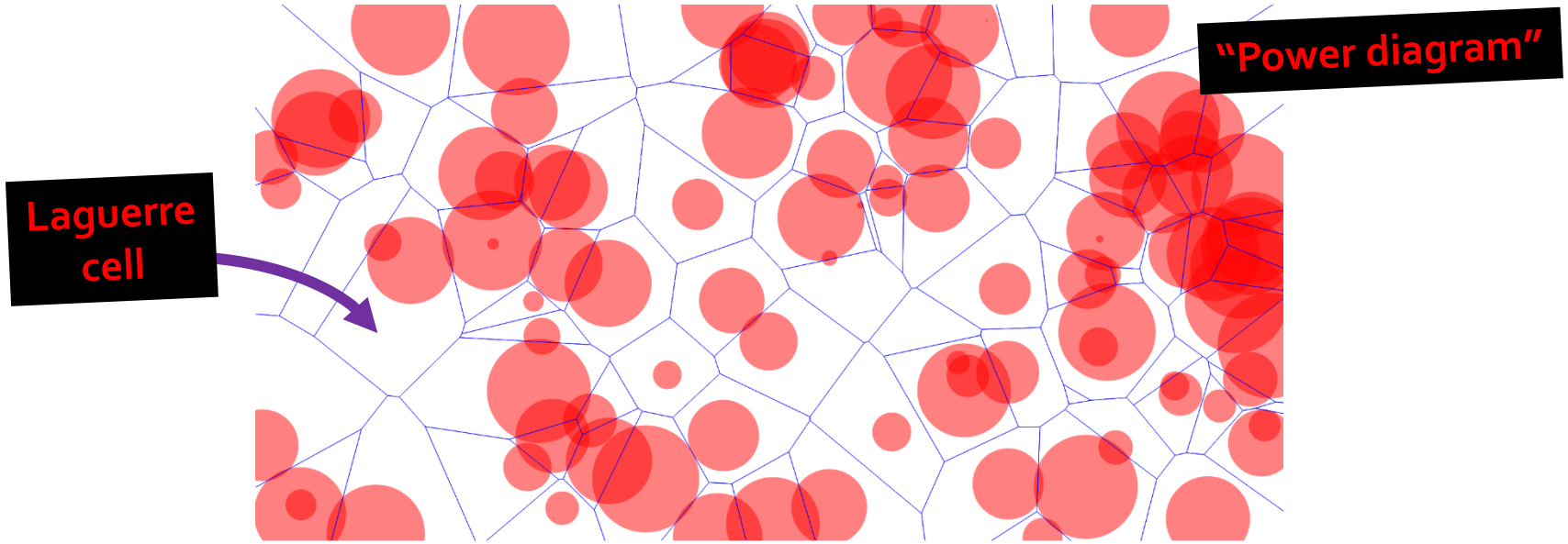
Never a reason to "leapfrog" mass!

Semidiscrete Transport

$$\mu_0 := \sum_{i=1}^{k} a_i \delta_{\mathbf{x}_i} \qquad \nu(S) := \int_S \rho(\mathbf{x})\, d\mathbf{x}$$

$$\mathcal{W}_2^2(\mu,\nu) = \sup_{\phi\in\mathbb{R}^k} \sum_i \left[a_i\phi_i + \int_{\mathrm{Lag}_\phi^c(\mathbf{x}_i)} \rho(\mathbf{y})[c(\mathbf{x}_i,\mathbf{y}) - \phi_i]\, dA(\mathbf{y}) \right]$$

$$\mathrm{Lag}_\phi^c(\mathbf{x}_i) := \{\mathbf{y}\in\mathbb{R}^n : c(\mathbf{x}_i,\mathbf{y}) - \phi_i \le c(\mathbf{x}_j,\mathbf{y}) - \phi_j \ \forall j \ne i\}$$



https://www.jasondavies.com/power-diagram/

Measure Quantization Problem (Balanced k-means)

Approximate by a discrete measure:

$$\frac{1}{k}\sum_{i=1}^{k}\delta_{\mathbf{x}_i}$$

Solve for *positions* of the points x_i.

Kantorovich Dual Problem

$$F[f, x_1, \ldots, x_m] = \frac{1}{m} \sum_{i=1}^{m} f_i + \sum_{i=1}^{m} \int_{V_{x_i}} \left(d(x_i, y)^2 - f_i \right) \mathrm{d}\mu_2(y)$$

weights

Power diagram regions

Alternating algorithm:

1. Update weights

Stochastic gradient descent

$$\frac{\partial F}{\partial f_i} = \frac{1}{m} - \int_{V_{x_i}} \mathrm{d}\mu_2(y)$$

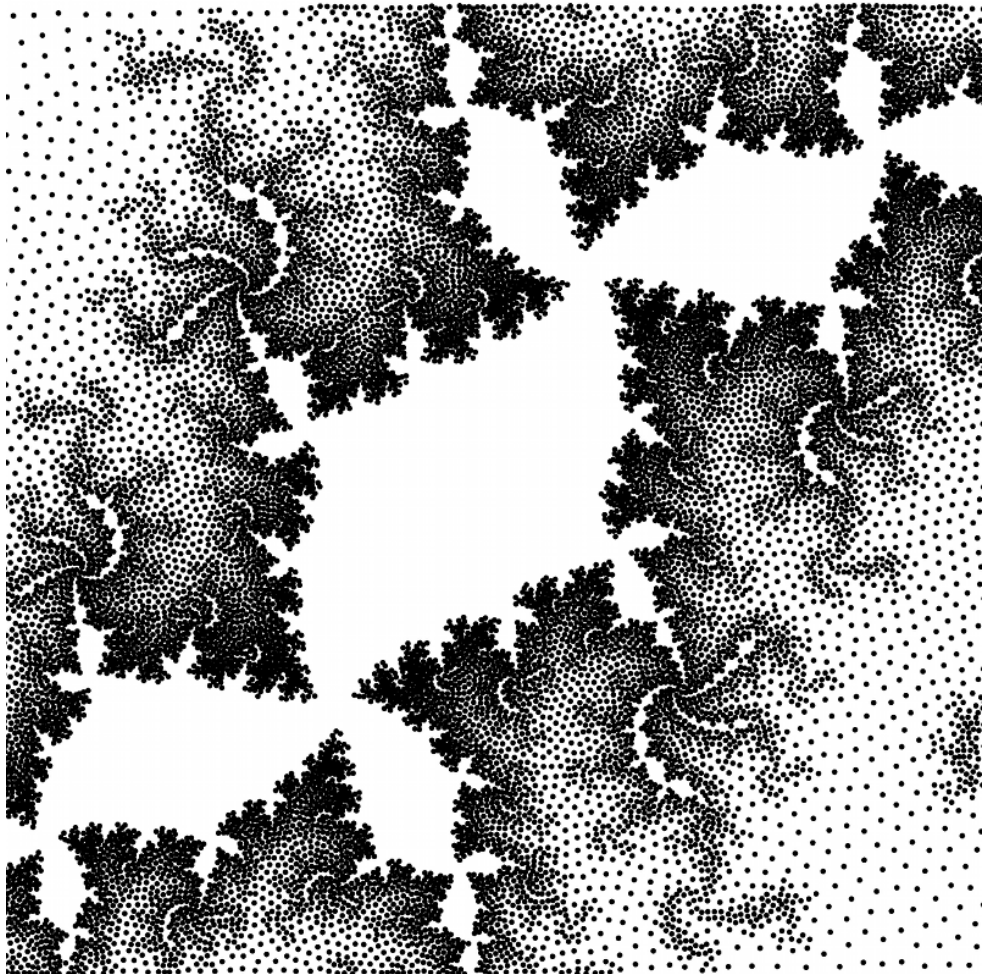
2. Update points

Fixed-point iteration

$$\frac{\partial F}{\partial x_i} = x_i \int_{V_{x_i}} \mathrm{d}\mu_2(y) - \int_{V_{x_i}} y \mathrm{d}\mu_2(y)$$

Claici, Chien, and Solomon. "Stochastic Wasserstein Barycenters." ICML 2018.

Related Algorithm for Low-Dimensional Problems



De Goes et al. "Blue Noise through Optimal Transport." SIGGRAPH Asia 2012.

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

$Q:$

What does it mean to average points in a metric space?

Fréchet Mean

aka
"Karcher mean"

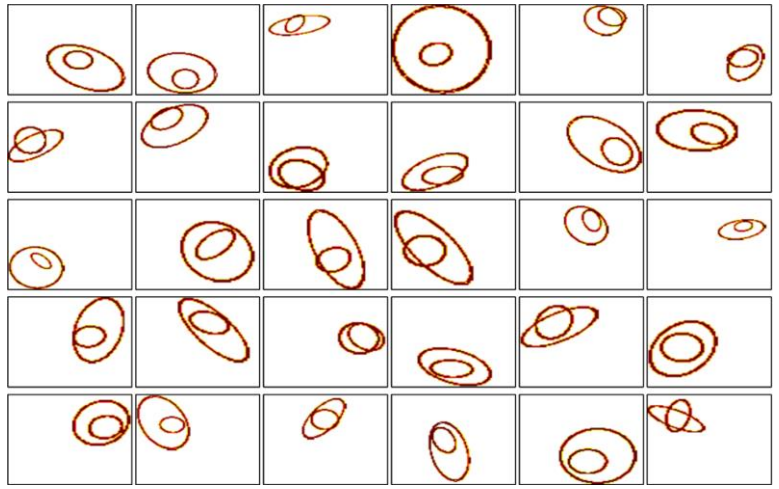
$$\arg\min_{p \in M} \underbrace{\sum_{i=1}^{N} d(p, x_i)^2}_{\text{"Fréchet variance"}}$$

Generalizes Euclidean notation of "mean."

$$\arg\min_{\mathbf{p}\in\mathbb{R}^n} \sum_{i=1}^{N} \|\mathbf{p}-\mathbf{x}_i\|_2^2$$

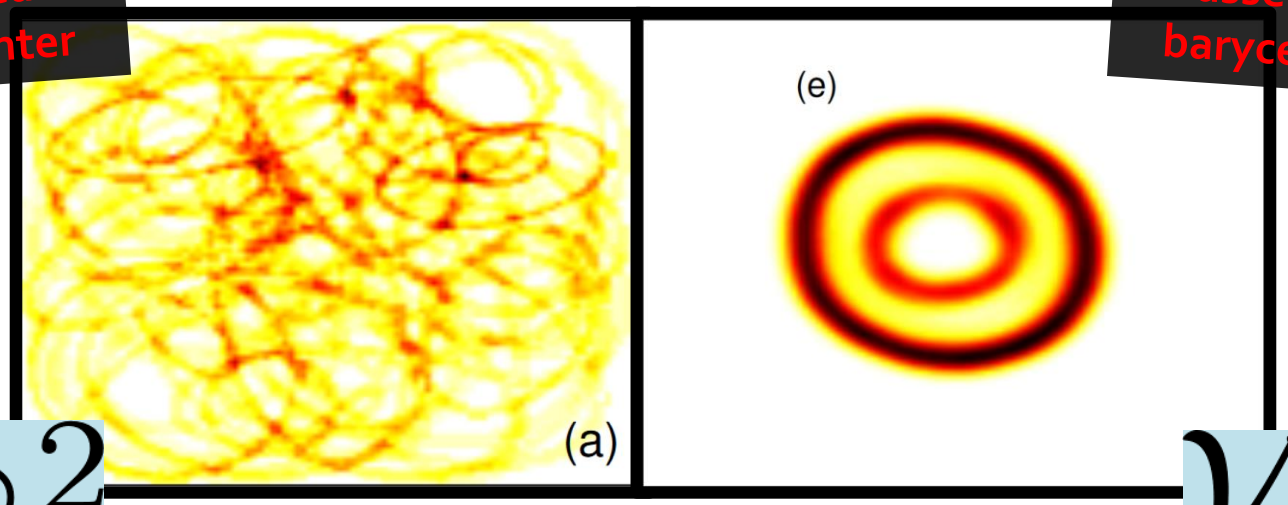
$\mathbb{R}_{++}$

Example from Past Lecture



Euclidean barycenter

Wasserstein barycenter



$\mathbb{R}^2$

W_2

"Fast Computation of Wasserstein Barycenters."
Cuturi and Doucet, ICML 2014

Application to Neural Networks

Differentiating through the Fréchet Mean

Aaron Lou [*1] **Isay Katsman** [*1] **Qingxuan Jiang** [*1] **Serge Belongie** [1] **Ser-Nam Lim** [2] **Christopher De Sa** [1]

Abstract

Recent advances in deep representation learning on Riemannian manifolds extend classical deep learning operations to better capture the geometry of the manifold. One possible extension is the Fréchet mean, the generalization of the Euclidean mean; however, it has been difficult to apply because it lacks a closed form with an easily computable derivative. In this paper, we show how to differentiate through the Fréchet mean for arbitrary Riemannian manifolds. Then, focusing on hyperbolic space, we derive explicit gradient expressions and a fast, accurate, and hyperparameter-free Fréchet mean solver. This fully integrates the Fréchet mean into the hyperbolic neural network pipeline. To demonstrate this integration, we present two case studies. First, we apply our Fréchet mean to the existing Hyperbolic Graph Convolutional Network, replacing its projected aggregation to obtain state-of-the-art results on datasets with high hyperbolicity. Second, to demonstrate the Fréchet mean's capacity to generalize Euclidean neural network operations, we develop a hyperbolic batch normalization method that gives an improvement parallel to the one observed in the Euclidean setting[1].

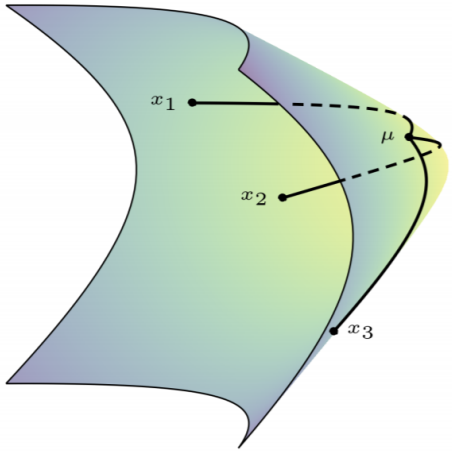


Figure 1. Depicted above is the Fréchet mean, μ, of three p
x_1, x_2, x_3 in the Lorentz model of hyperbolic space. A
can see, the Fréchet mean conforms with the geometry
hyperboloid and is vastly different from the standard Euc
mean.

space, in which distances grow exponentially as one n
away from the origin. Such a geometry is naturally equi
to embed trees, since if we embed the root of the tree
the origin and layers at successive radii, the geometry
perbolic space admits a natural hierarchical structure. More

Table 1. Summary of operations in the Poincaré ball model and the hyperboloid model ($K < 0$)

	Poincaré Ball	Hyperboloid
Manifold	$\mathbb{D}_K^n = \{x \in \mathbb{R}^n : \langle x, x\rangle_2 < -\frac{1}{K}\}$	$\mathbb{H}_K^n = \{x \in \mathbb{R}^{n+1} : \langle x, x\rangle_{\mathcal{L}} = \frac{1}{K}\}$
Metric	$g_x^{\mathbb{D}_K} = (\lambda_x^K)^2 g^{\mathbb{E}}$ where $\lambda_x^K = \frac{2}{1+K\|x\|_2^2}$ and $g^{\mathbb{E}} = I$	$g_x^{\mathbb{H}_K} = \eta$, where η is I except $\eta_{0,0} = -1$
Distance	$d_{\mathbb{D}}^K(x,y) = \frac{1}{\sqrt{\lvert K\rvert}} \cosh^{-1}\left(1 - \frac{2K\|x-y\|_2^2}{(1+K\|x\|_2^2)(1+K\|y\|_2^2)}\right)$	$d_{\mathbb{H}}^K(x,y) = \frac{1}{\sqrt{\lvert K\rvert}} \cosh^{-1}(K\langle x, y\rangle_{\mathcal{L}})$
Exp map	$\exp_x^K(v) = x \oplus_K \left(\tanh\left(\sqrt{\lvert K\rvert}\frac{\lambda_x^K\|v\|_2}{2}\right)\frac{v}{\sqrt{\lvert K\rvert}\|v\|_2}\right)$	$\exp_x^K(v) = \cosh(\sqrt{\lvert K\rvert}\|\|v\|\|_{\mathcal{L}})x + v\frac{\sinh(\sqrt{\lvert K\rvert}\|\|v\|\|_{\mathcal{L}})}{\sqrt{\lvert K\rvert}\|\|v\|\|_{\mathcal{L}}}$
Log map	$\log_x^K(y) = \frac{2}{\sqrt{\lvert K\rvert}\lambda_x^K}\tanh^{-1}(\sqrt{\lvert K\rvert}\|\|-x\oplus_K y\|_2)\frac{-x\oplus_K y}{\|-x\oplus_K y\|_2}$	$\log_x^K(y) = \frac{\cosh^{-1}(K\langle x,y\rangle_{\mathcal{L}})}{\sinh(\cosh^{-1}(K\langle x,y\rangle_{\mathcal{L}}))}(y - K\langle x,y\rangle_{\mathcal{L}}x)$
Transport	$PT_{x\to y}^K(v) = \frac{\lambda_x^K}{\lambda_y^K}\operatorname{gyr}[y,-x]v$	$PT_{x\to y}^K(v) = v - \frac{K\langle y,v\rangle_{\mathcal{L}}}{1+K\langle x,y\rangle_{\mathcal{L}}}(x+y)$

Table 2. Summary of hyperbolic counterparts of Euclidean operations in neural networks

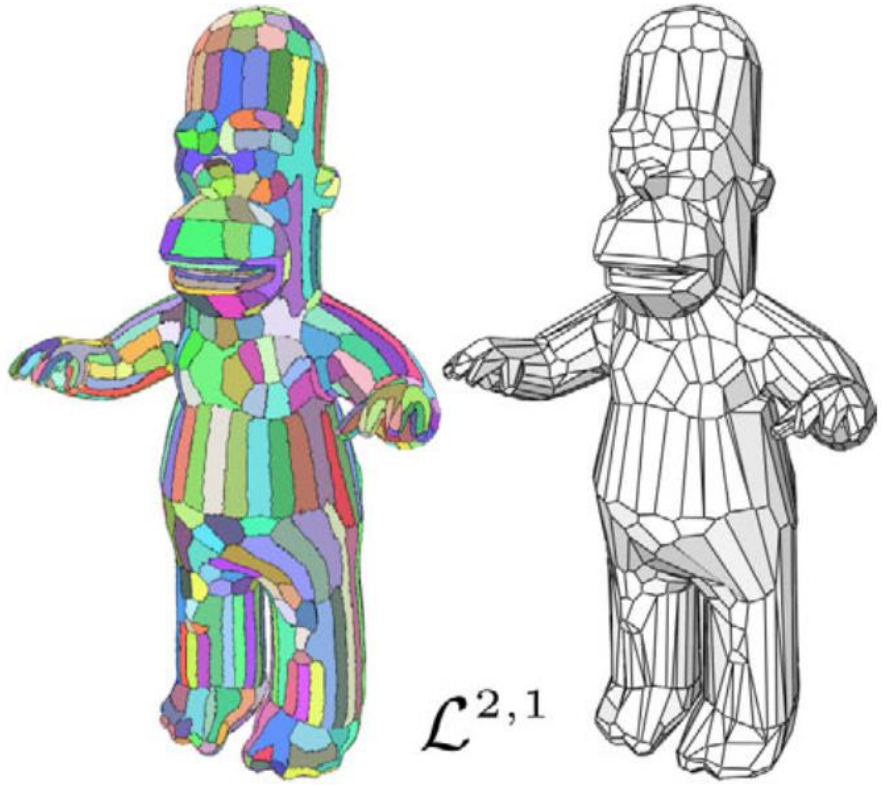
Operation	Formula
Matrix-vector multiplication	$A \otimes^K x = \exp_0^K(A\log_0^K(x))$
Bias translation	$x \oplus^K b = \exp_x(PT_{0\to x}^K(b))$
Activation function	$\sigma^{K_1,K_2}(x) = \exp_0^{K_1}(\sigma(\log_0^{K_2}(x)))$

Finding Nicely-Shaped Regions on a Surface

Lloyd's Algorithm

Alternate between

1. Fitting primitive parameters
2. Assign points to patches



"Variational Shape Approximation."
Cohen-Steiner, Alliez, and Desbrun; SIGGRAPH 2004

Shape Collections

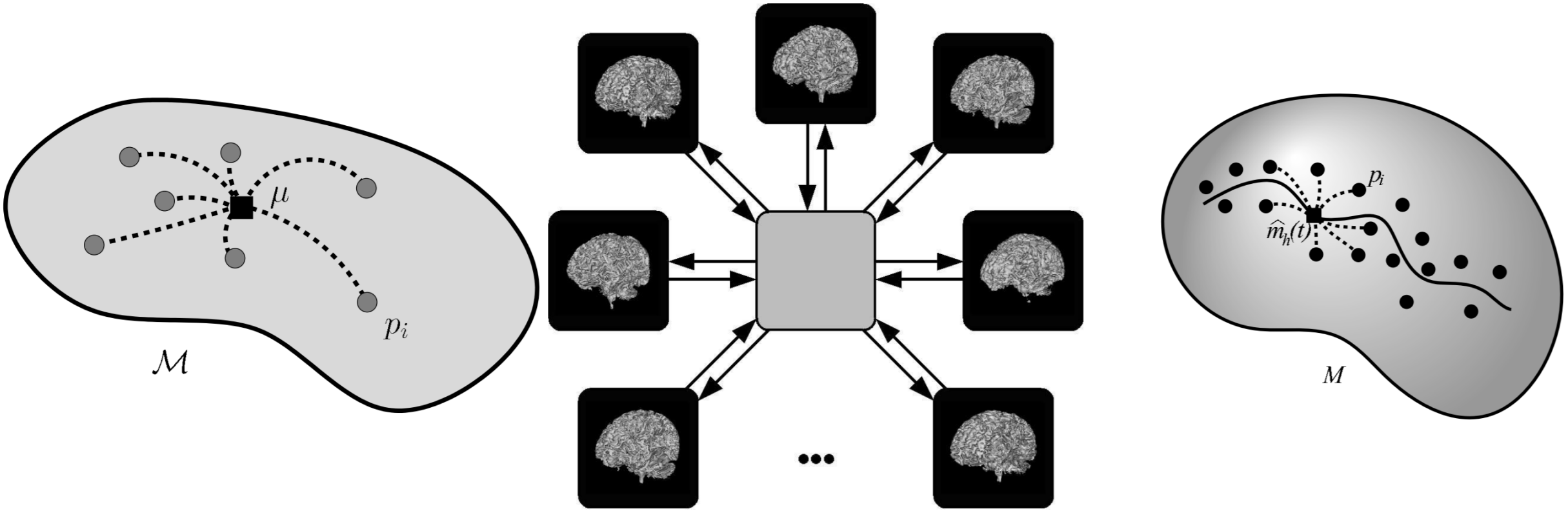


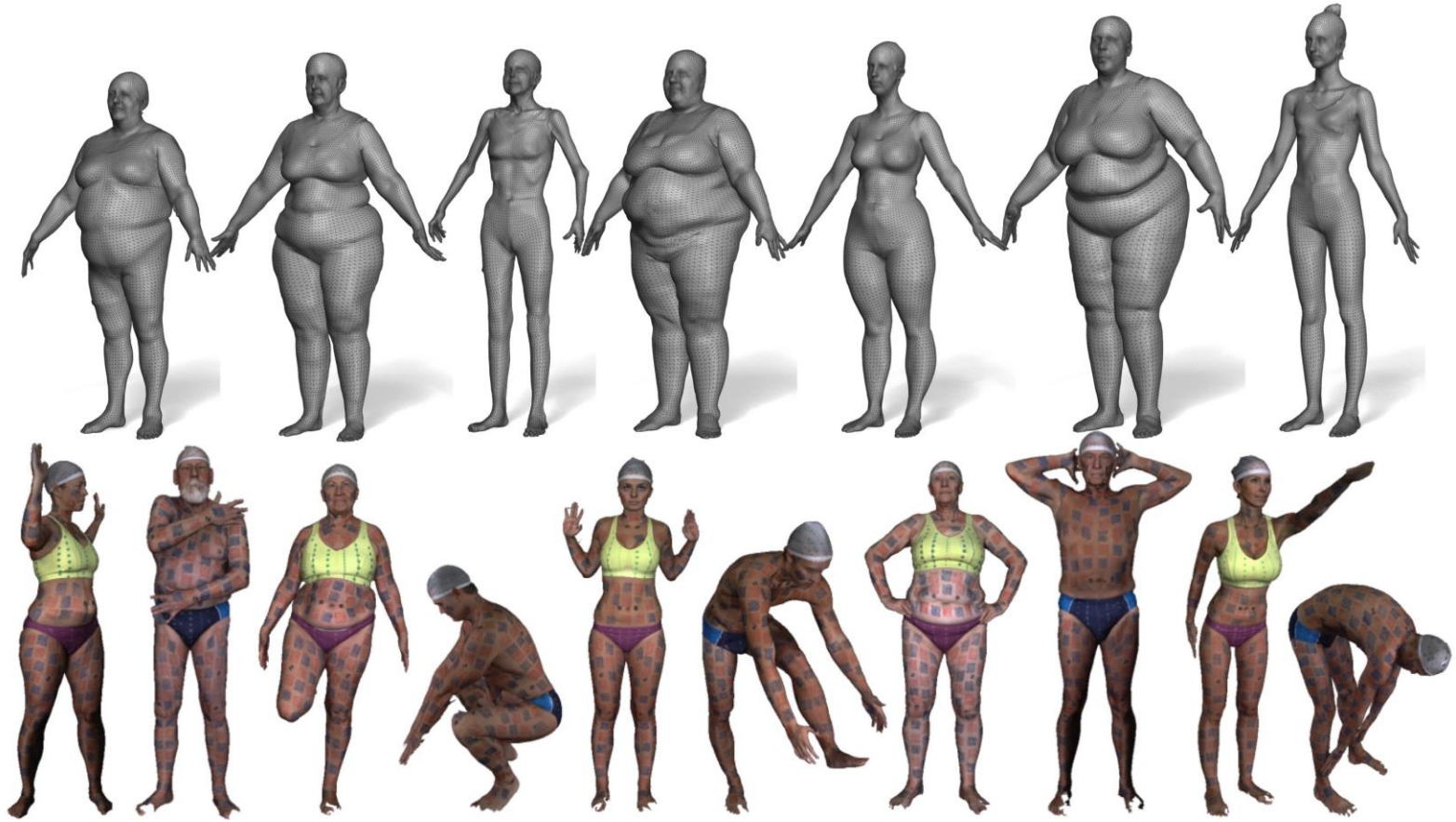
Image from "Medical Image Analysis via Fréchet Means of Diffeomorphisms (Davis 2008)

k-Medioids

- 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

Example Task



https://ps.is.tuebingen.mpg.de/research_projects/3d-mesh-registration

Clustering in a shape collection

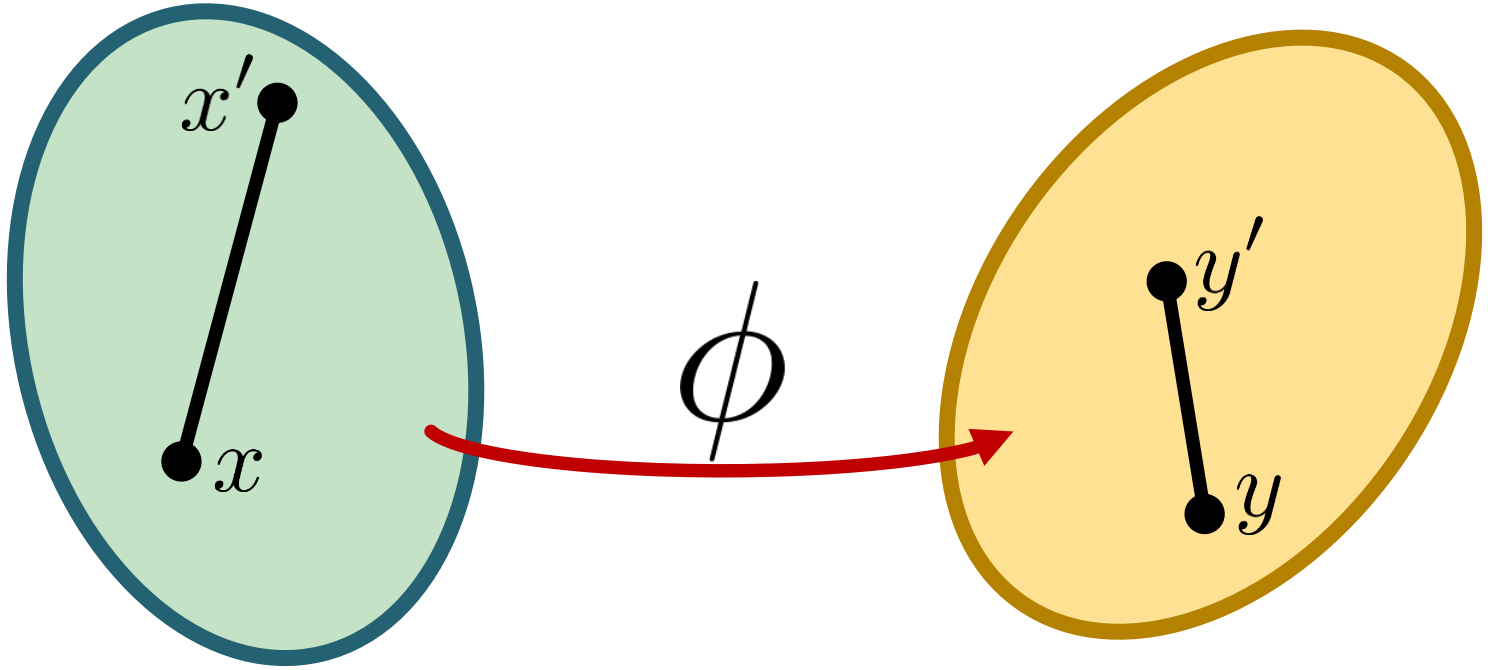
Gromov-Hausdorff Distance

Distance between metric spaces X, Y

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

Best map

Worst distortion



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[1†]

[1]Department of Mathematics, Stanford University, California, USA.

Abstract

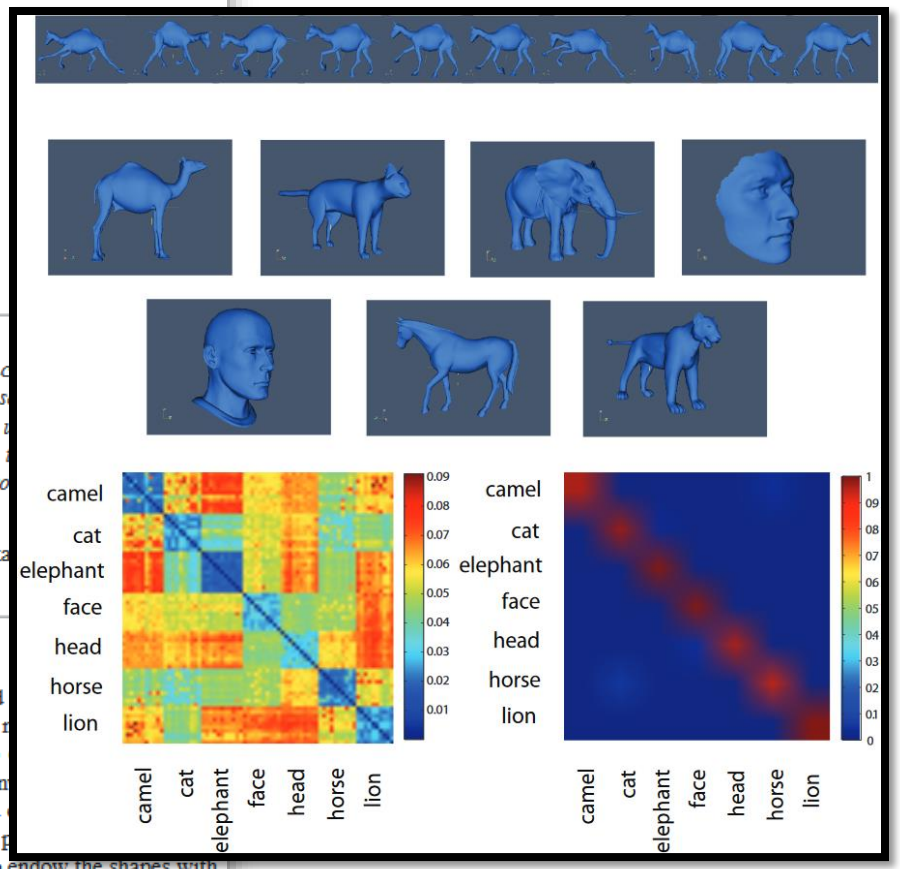
It is the purpose of this paper to propose and discuss certain modifications of the ideas conc Hausdorff distances in order to tackle the problems of shape matching and comparison. Thes render these distances more amenable to practical computations without sacrificing theoretical u second goal of this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

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

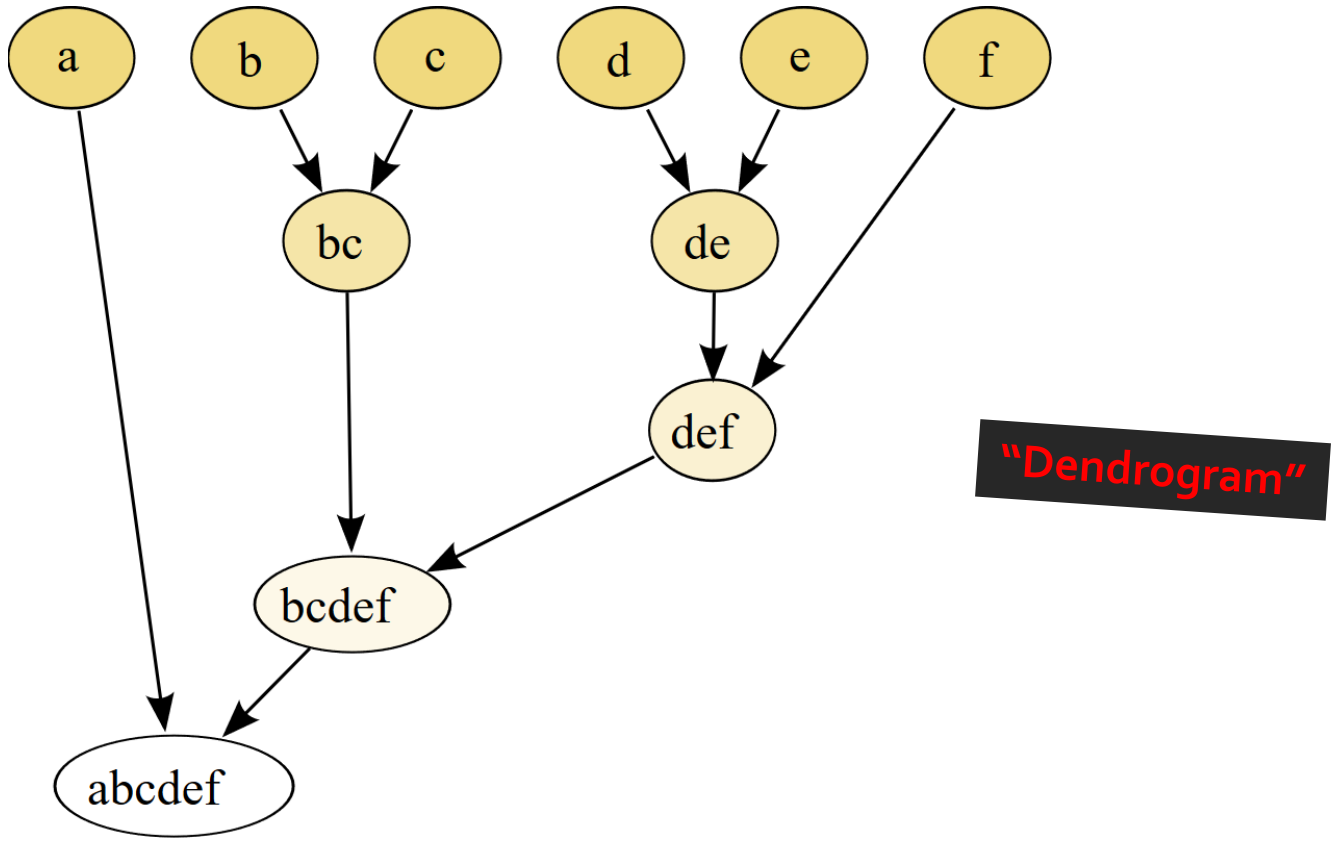
1. Introduction

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 notion of distance compares the full r tained in the shapes, as opposed to only compare simple (incomplete) in shapes will be declared *equal* if and ric. This means that the invariance p coded by the metrics one chooses to endow the shapes with.



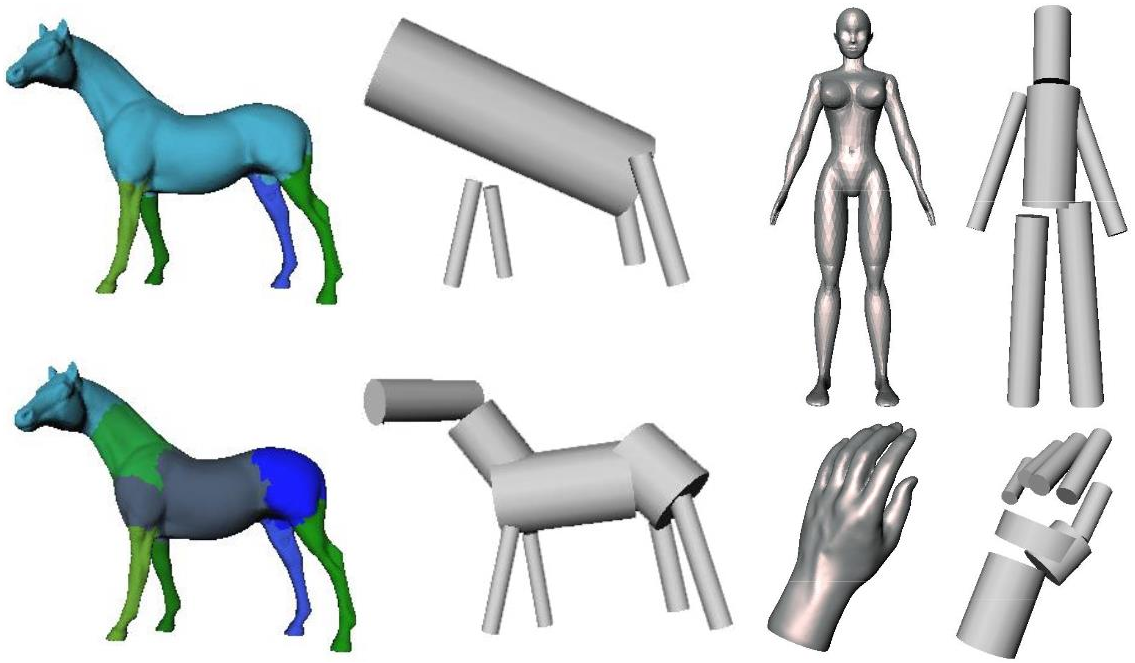
Agglomerative Clustering



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

Related Technique: Flood Fill

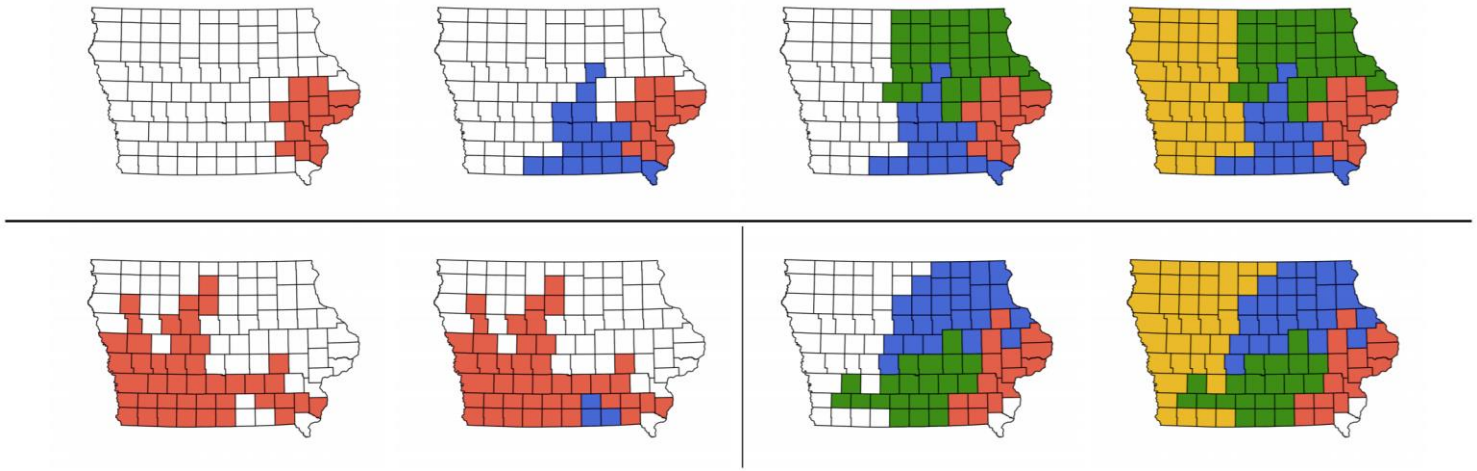
Region Growing Algorithm

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

Flood Fill and Agglomerative Clustering in Redistricting



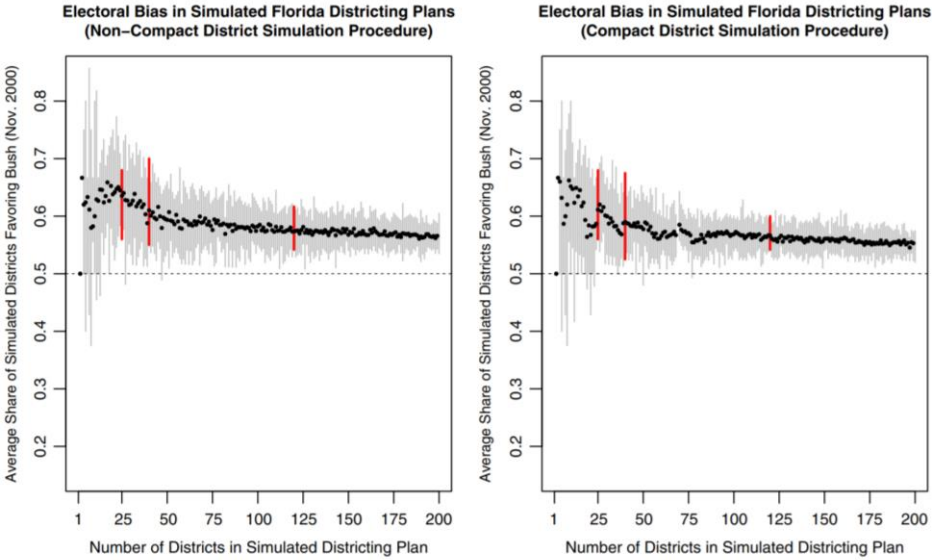


Figure 4. Republican electoral bias in simulated Florida districting plans.

Note: Black dots indicate the average share of simulated districts that have pro-Bush majorities in the simulated plans. Gray bars depict the entire range of pro-Bush district shares that were observed across all simulations for each given legislature size. Red bars depict the range of simulated outcomes for legislatures of 25 districts (Florida's Congressional Delegation), 40 districts (the Florida State Senate), and 120 districts (the Florida State House).

Images from "Redistricting Algorithms" (Becker and Solomon) and
"Unintentional Gerrymandering: Political Geography and Electoral Bias in Legislatures" (Chen and Rodden, 2013)

Clustering in Feature Space

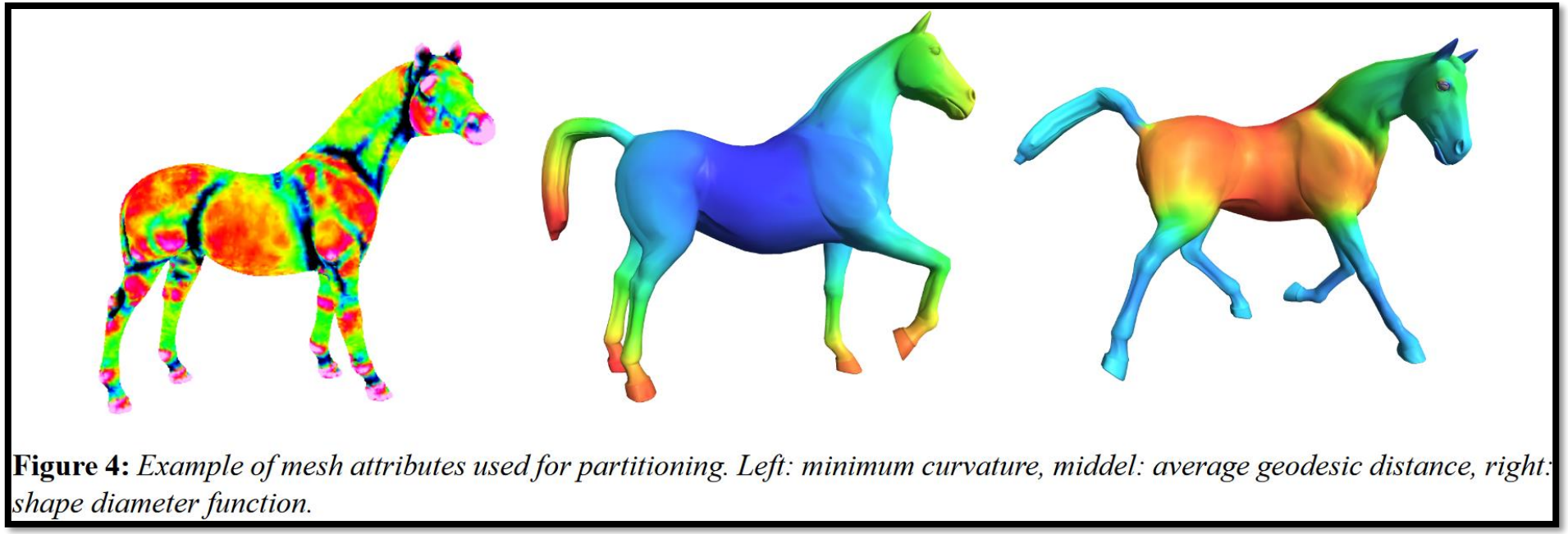
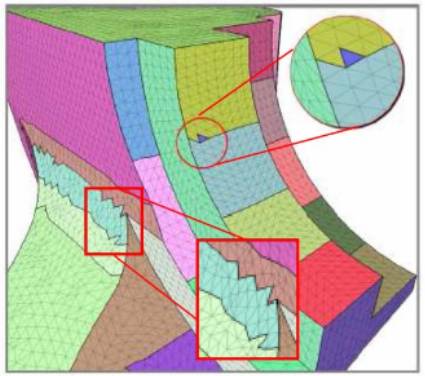


Figure 4: *Example of mesh attributes used for partitioning. Left: minimum curvature, middel: average geodesic distance, right: shape diameter function.*

"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.

Global Optimality Unlikely

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

The planar *k*-means problem is NP-hard☆

Meena Mahajan[a,*], Prajakta Nimbhorkar[a], Kasturi Varadarajan[b]

[a] *The Institute of Mathematical Sciences, Chennai 600 113, India*

[b] *The University of Iowa, Iowa City, IA 52242-1419, USA*

ARTICLE INFO

Keywords:
Clustering
k-means
Planar graphs
NP-hardness

ABSTRACT

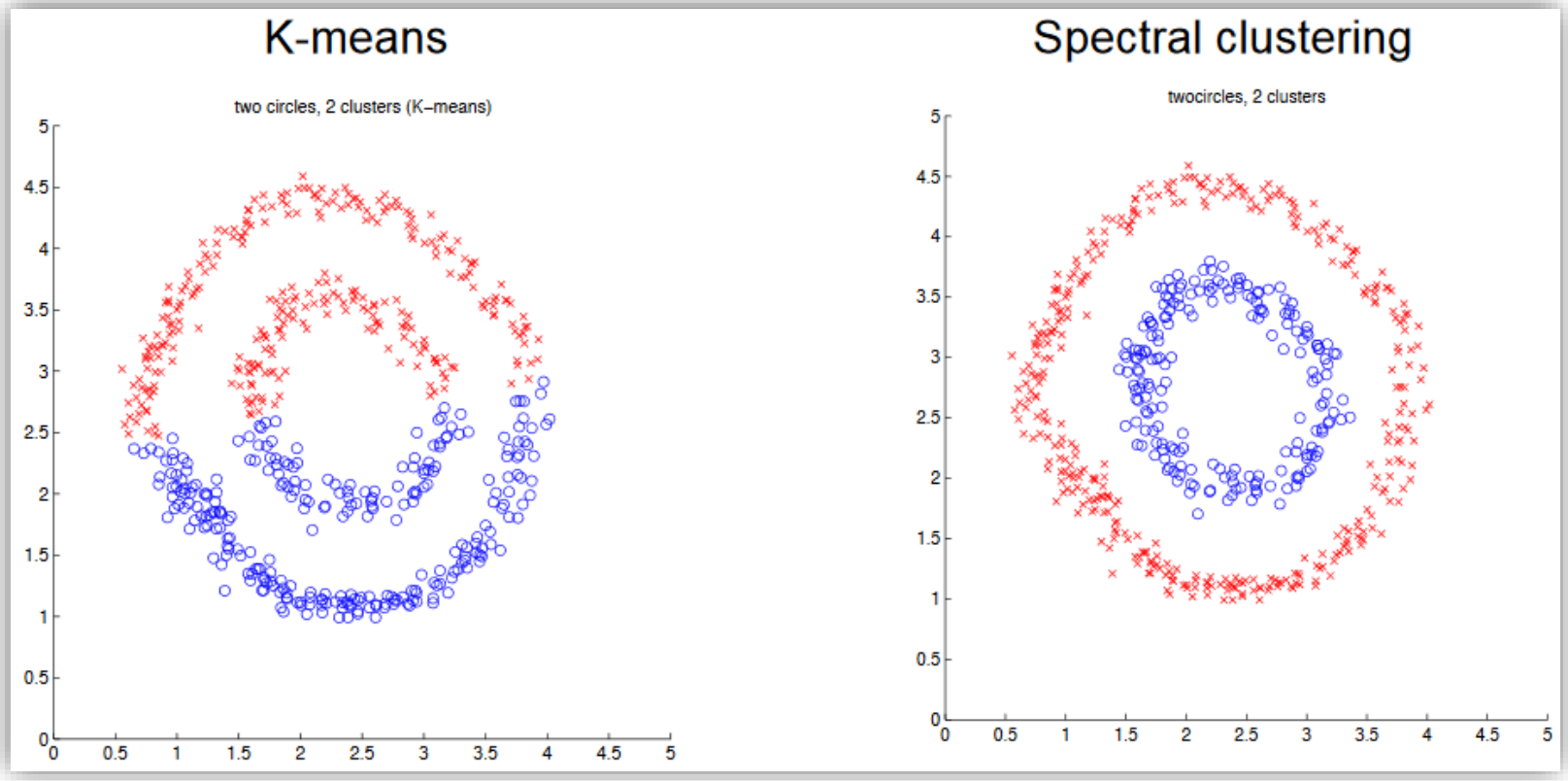
In the *k*-means problem, we are given a finite set S of points in $\Re^m$, and integer $k \geq 1$, and we want to find k points (centers) so as to minimize the sum of the square of the Euclidean distance of each point in S to its nearest center. We show that this well-known problem is NP-hard even for instances in the plane, answering an open question posed by Dasgupta (2007) [7].

1. Introduction

In the *k*-means problem, we are given a finite set S of points in $\Re^m$, and integer $k \geq 1$, and we want to find k points (centers) so as to minimize the sum of the square of the Euclidean distance of each point in S to its nearest center. This is a well-known and popular clustering problem that has also received a lot of attention from the algorithms community.

Spectral Clustering

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



- Rough notion of optimality
- Assembles local relationships

Normalized Cuts for Two Cuts

Symmetric similarity matrix W

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

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

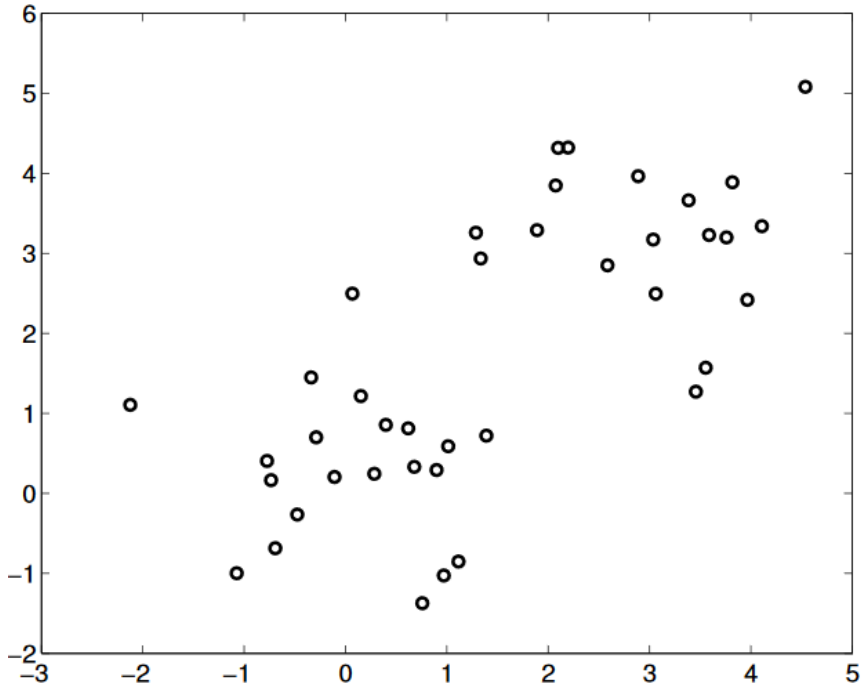
$$x_i := \begin{cases} V(A)^{-1} & \text{if } i \in A \\ -V(B)^{-1} & \text{if } i \in B \end{cases}$$

Eigenvalue Problem

$$\min_{\mathbf{x}} \frac{\mathbf{x}^\top L\mathbf{x}}{\mathbf{x}^\top D\mathbf{x}}$$
$$\text{s.t. } \mathbf{x}^\top D\mathbf{1} = 0$$

$$L\mathbf{x} = \lambda D\mathbf{x}$$

Example on kNN Graph



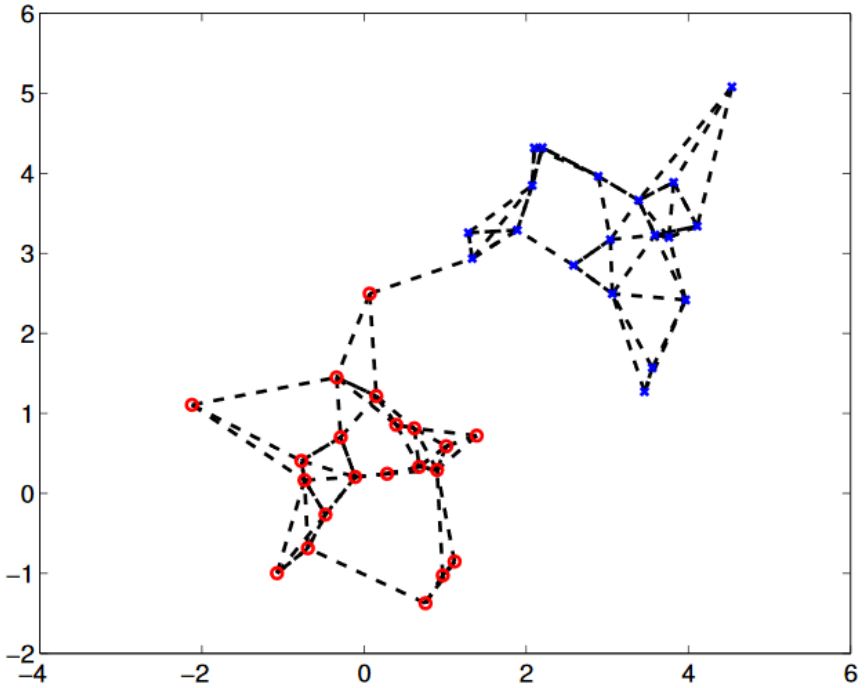


Image courtesy D. Sontag

For ≥ 2 Clusters

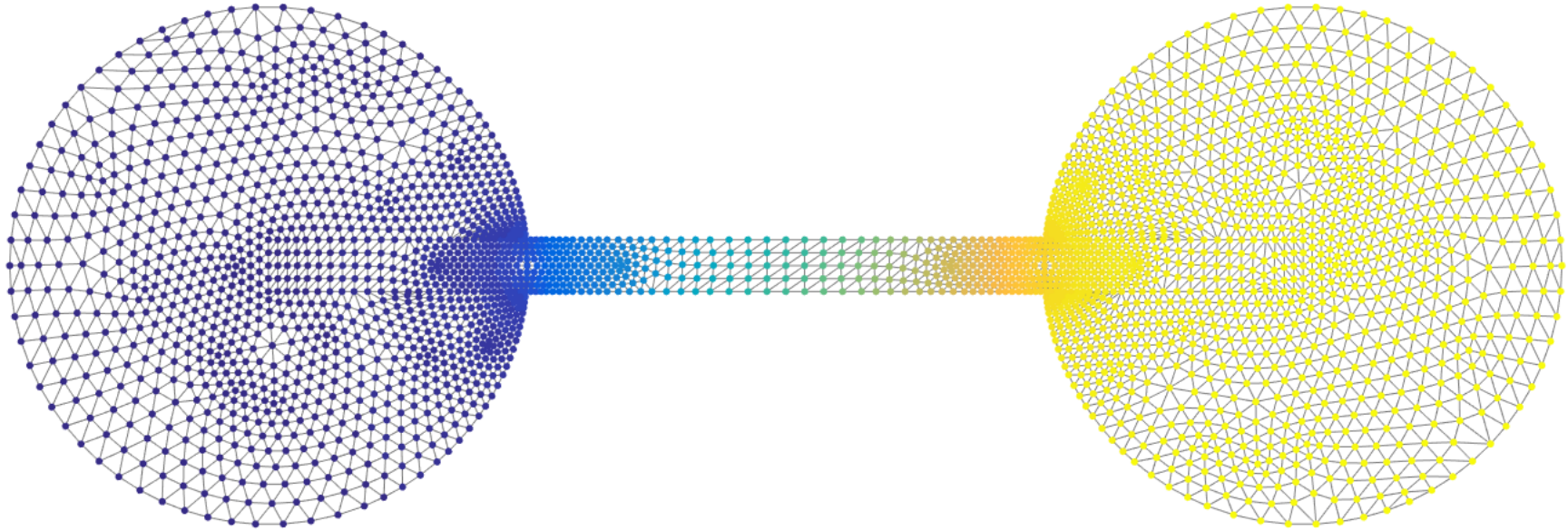
- **Recursive bi-partitioning (Hagen et al. 1991)**
  - **Analogy: Agglomerative clustering**
  - **Potentially slow/unstable**

- **Cluster multiple eigenvectors**
  - **Analogy: *k*-means after dimension reduction**
  - **More popular approach**

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

Recall:

Second-Smallest Eigenvector



$$L\mathbf{x} = \lambda\mathbf{x}$$

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

Technical Section

Discrete Laplace–Beltrami operators for shape analysis and segmentation

Martin Reuter [a,b], Silvia Biasotti [c,*], Daniela Giorgi [c], Giuseppe Patanè [c], Michela Spagnuolo [c]

[a] Massachusetts Institute of Technology, Cambridge, MA, USA
[b] A.A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital, Harvard Medical School, Boston, MA, USA
[c] 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 segmentati

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

1. Introductio

Shape ana
reasoning on p
large number
processing to
In the rece
the need to a
objects, in orde
content in many emerging web-based applications. A semantic

analysis and segmentation able to capture a varied set of

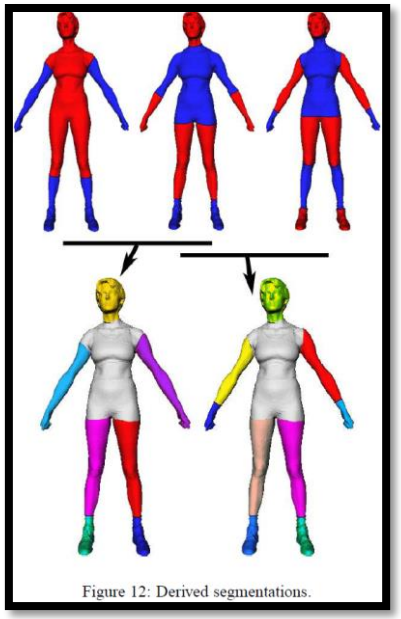


Figure 12: Derived segmentations.

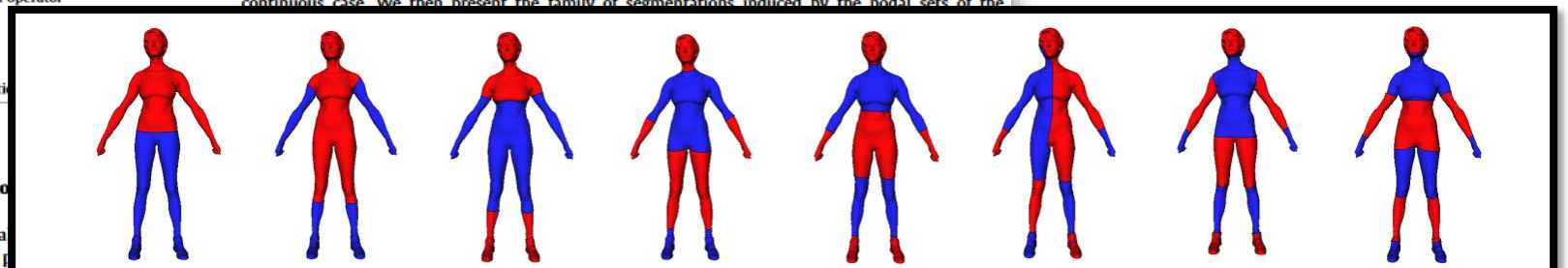
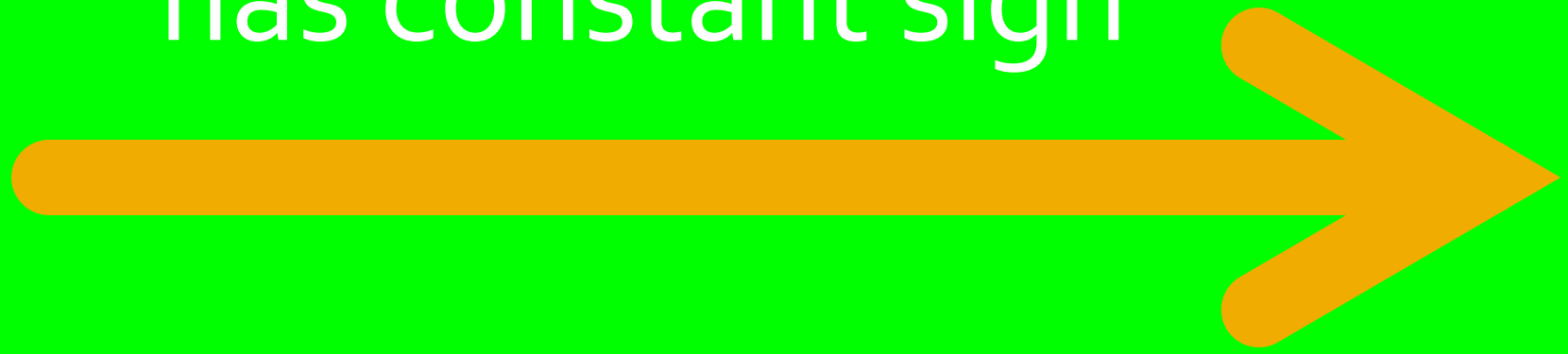


Figure 7: Segmentations induced by the nodal domains of some eigenfunctions selected among the first 15 eigenfunctions (in order of increasing eigenvalues). Blue regions correspond to regions where the eigenfunctions have negative values, while red regions correspond to positive values.

Nodal domain

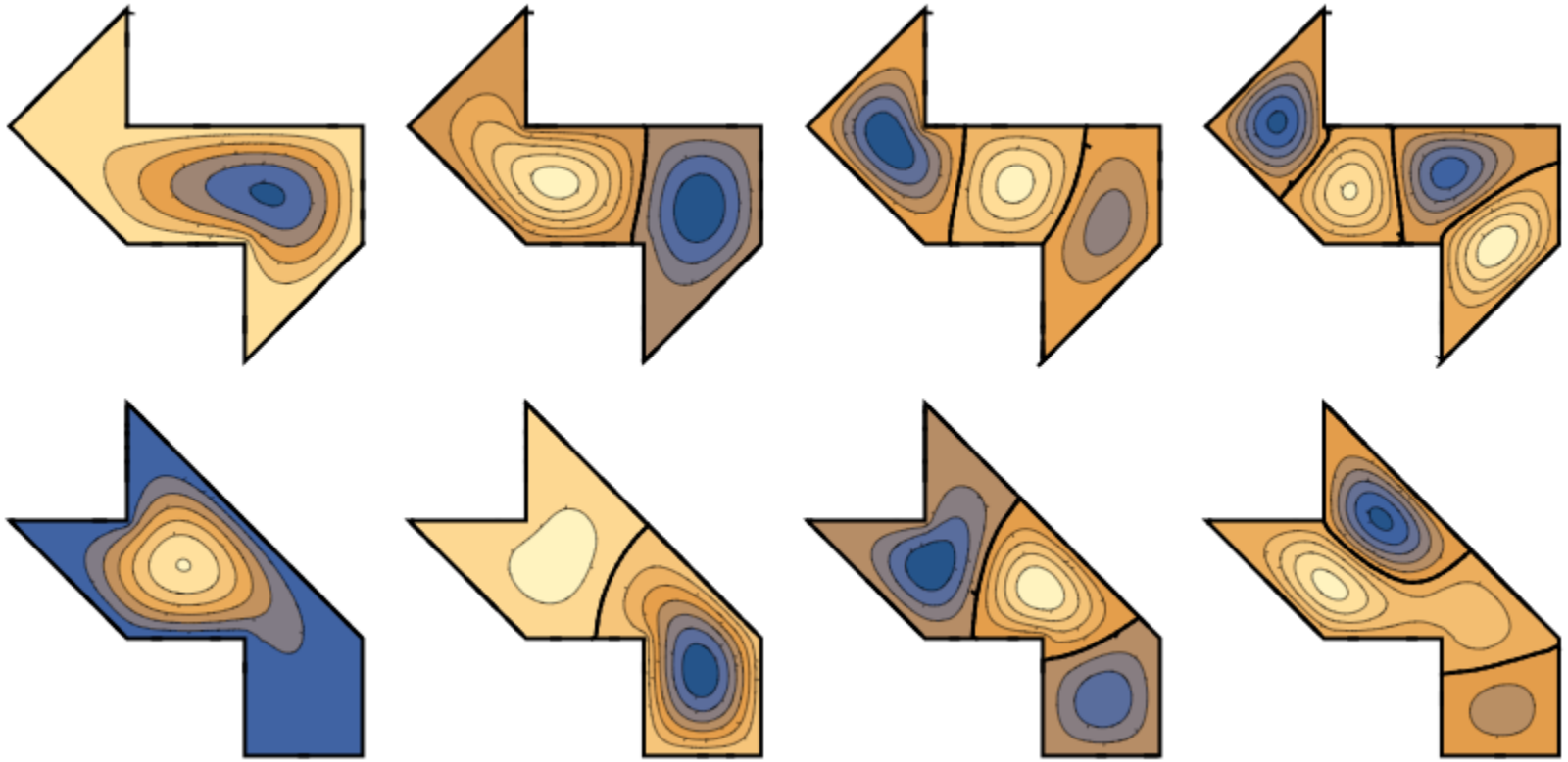
[**nohd**-l doh-**meyn**]:

A connected region where a Laplacian eigenfunction has constant sign



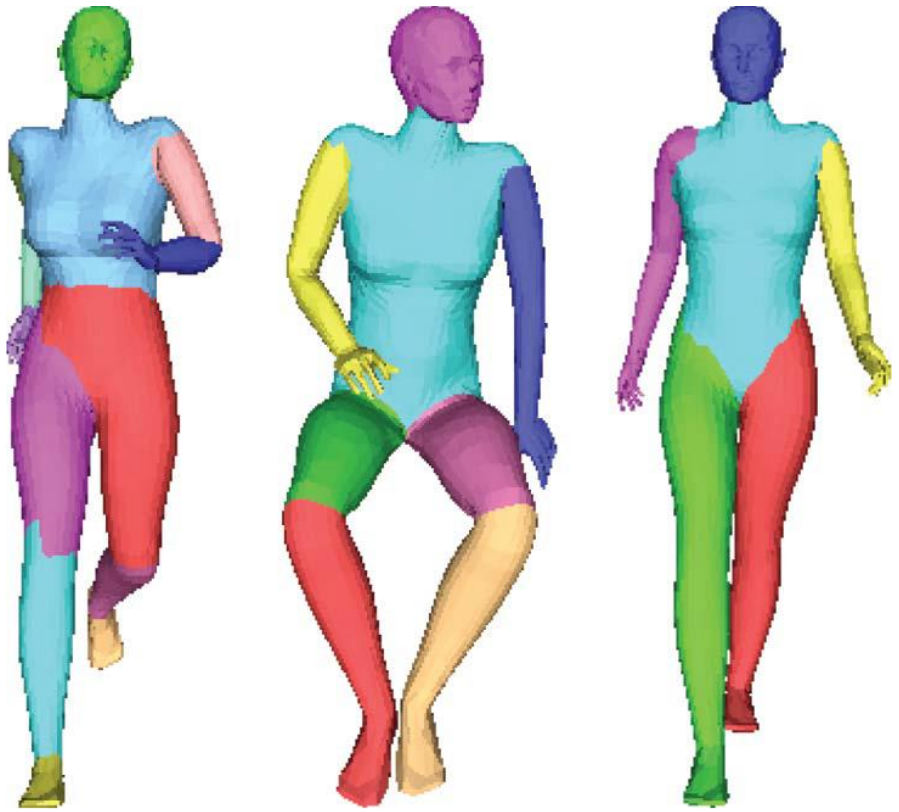
Courant's Theorem

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



https://i.stack.imgur.com/JJIFP.png

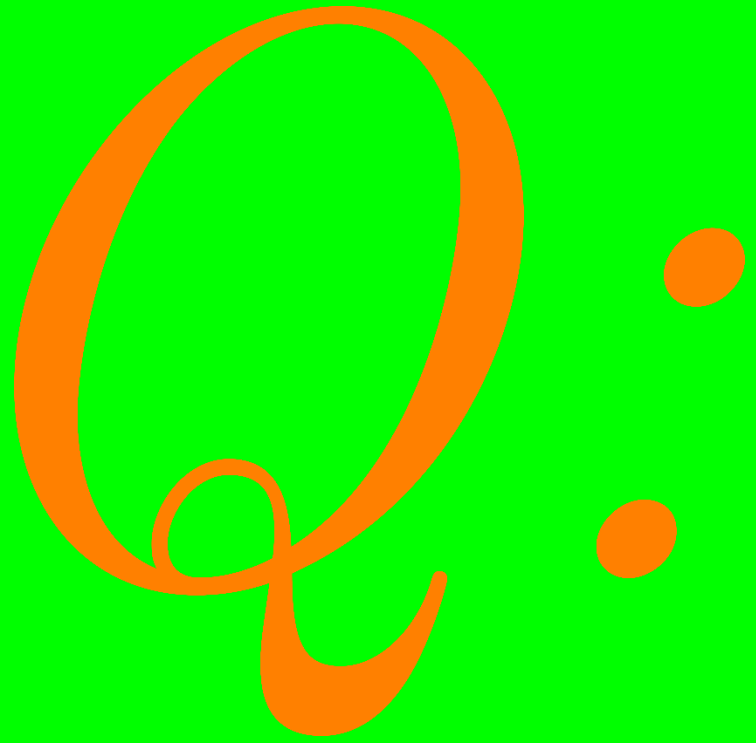
Issue



From randomized cuts

Image courtesy Q. Huang

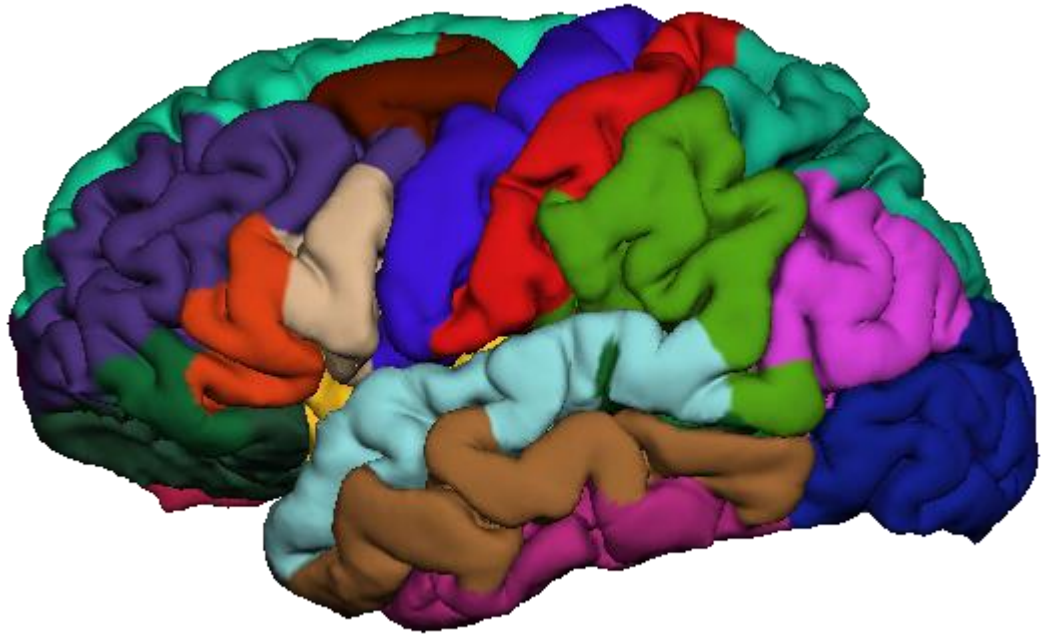
Inconsistent!



Is segmentation a purely geometric problem?

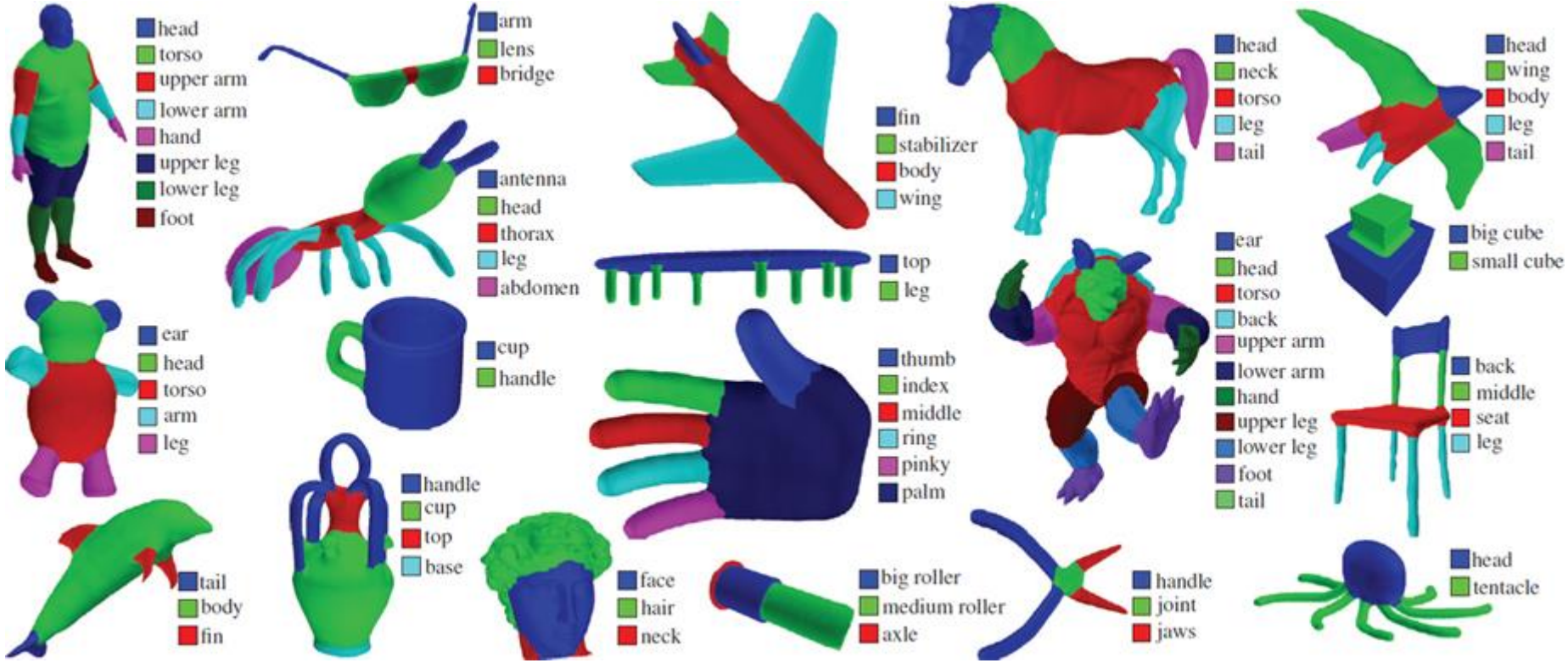
Obvious Counterexample

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



Shape provides only a clue

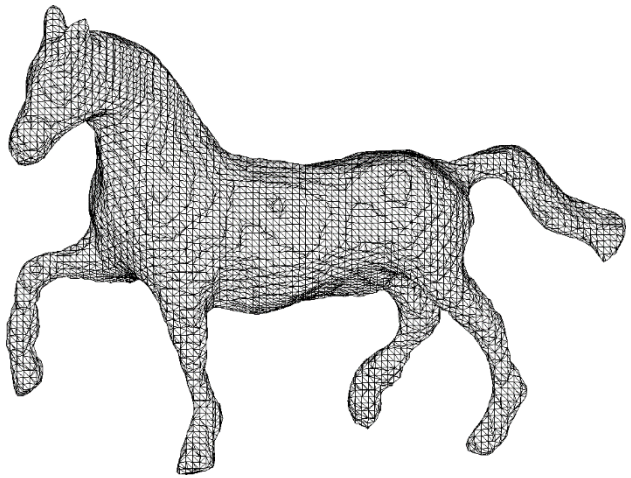
Supervised Learning



"Learning 3D Mesh Segmentation and Labeling."
Kalogerakis, Hertzmann, and Singh; SIGGRAPH 2010

Use example data to help

Conditional Random Field



Input Mesh

Labeled Mesh

- Head
- Neck
- Torso
- Leg
- Tail
- Ear

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

Unary descriptor term

Binary label compatibility term

Preview: Supervised Learning

More later…

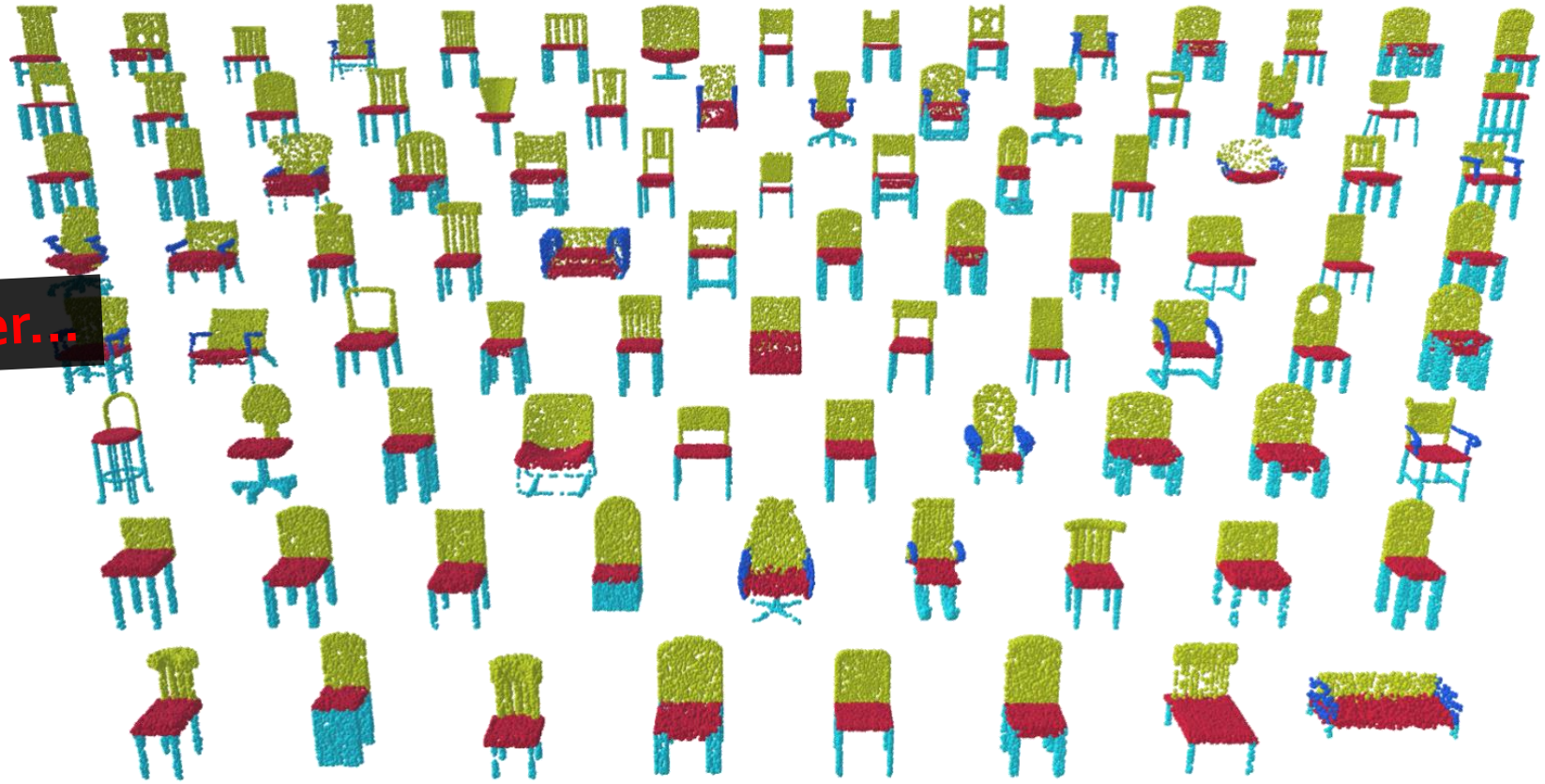
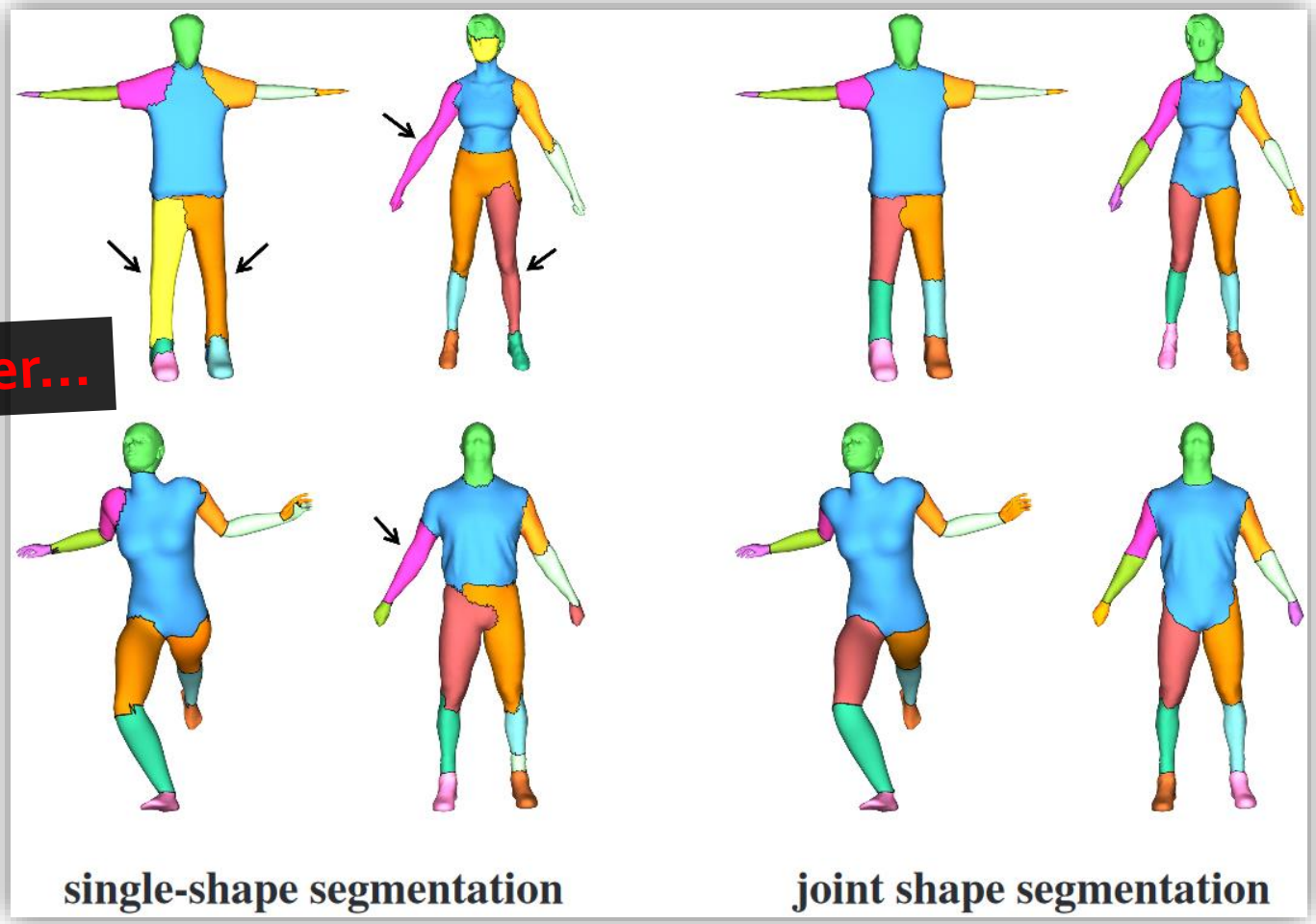


Image from "Dynamic Graph CNN for Learning on Point Clouds" (Wang et al. 2019)

Preview: Unsupervised Learning

More later…



"Joint Shape Segmentation with Linear Programming."
Huang, Koltun, and Guibas; SIGGRAPH Asia 2011

Clustering and Segmentation

Justin Solomon

6.838: Shape Analysis

Spring 2021