Mesh segmentation and 3D object recognition

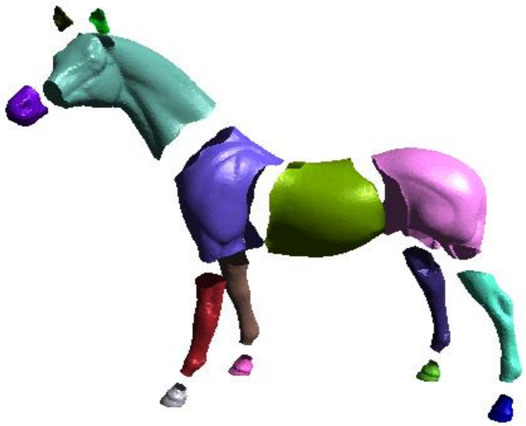
Florent Lafarge

Inria Sophia Antipolis - Mediterranee

What is mesh segmentation ?

What is mesh segmentation ?

- M = {V,E,F} is a mesh
- S is either V, E or F (usually F)
- A Segmentation is the set of sub-meshes induced by a partition of S into k disjoint subsets
- segmentation can also be called partitioning or clustering



Formulating a mesh segmentation problem

Usually specifies by two key elements:

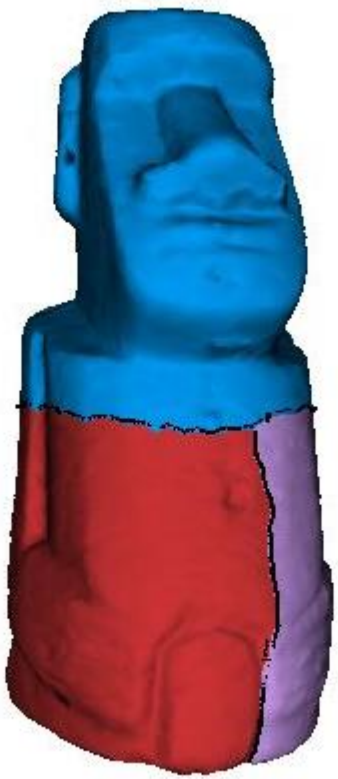
- a function measuring the quality of a partition, eventually under a set of constraints

- a mechanism for finding an optimal partition.

Contents

- Attributes and constraints
- Segmentation algorithms

How to measure the quality of a segmentation ?



What are we looking for ?

- Planar clusters?

What are we looking for ?

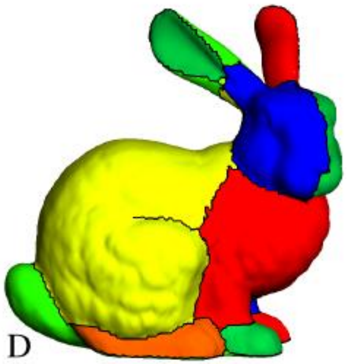
- Planar clusters?
- Smooth clusters?

What are we looking for ?

- Planar clusters?
- Smooth clusters?
- Round clusters?

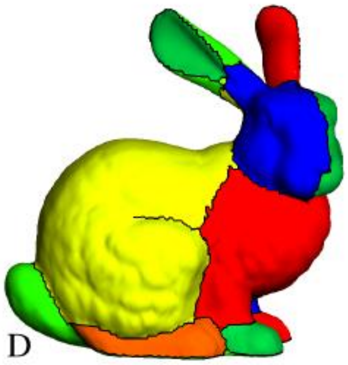
What are we looking for ?

- Planar clusters?
- Smooth clusters?
- Round clusters?
- Small/large clusters?



What are we looking for ?

- Planar clusters?
- Smooth clusters?
- Round clusters?
- Small/large clusters?
- Small number of clusters?



What are we looking for ?

- Planar clusters?
- Smooth clusters?
- Round clusters?
- Small/large clusters?
- Small number of clusters?
- Smooth boundary?

What are we looking for ?

- Planar clusters?
- Smooth clusters?
- Round clusters?
- Small/large clusters?
- Small number of clusters?
- Smooth boundary?
- structural clusters?
- …

Attributes and constraints

- Attributes:
  - criteria used to measure the quality of a partition with respect to the input mesh
  - attributes are usually embedded into some metrics

Attributes and constraints

- Attributes:
  - criteria used to measure the quality of a partition with respect to the input mesh
  - attributes are usually embedded into some metrics

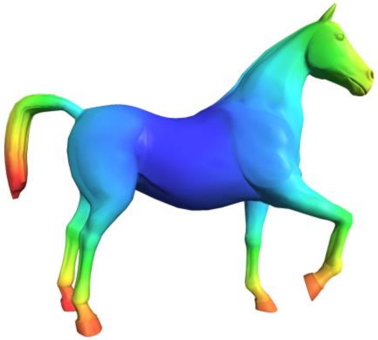
- Constraints:
  - cardinality, geometry or topology of the clusters
  - must be preserved (hard) or favored (soft)

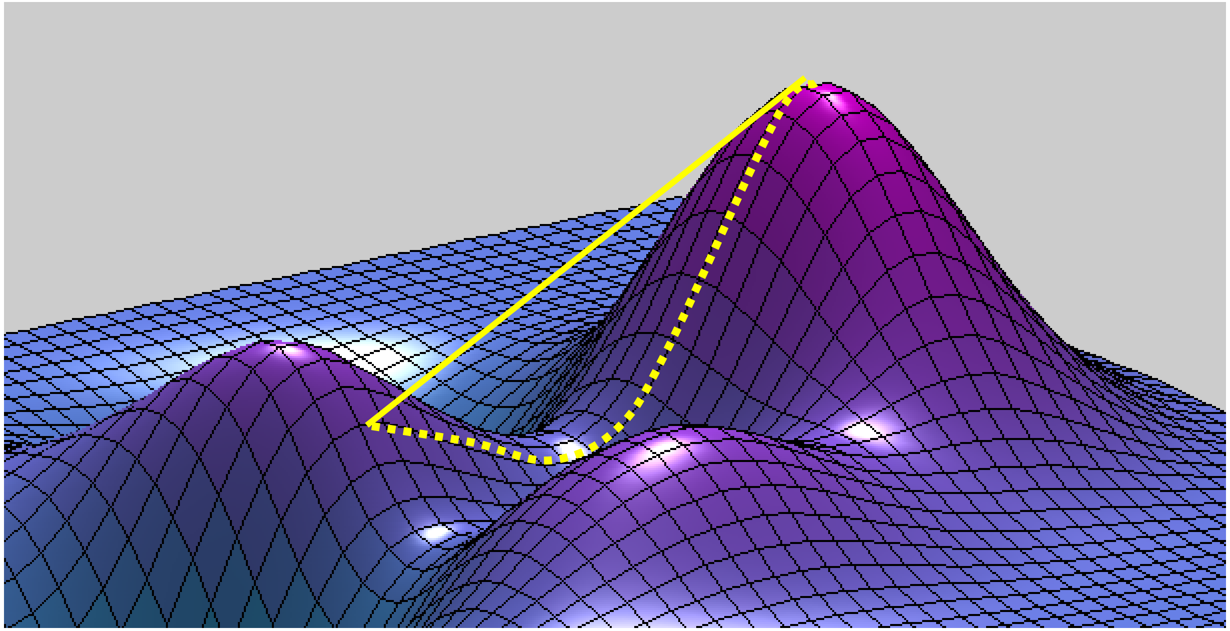
Attributes and constraints

- example of problems
  - without constraints $\boxed{\min_P f(p,\text{attributes})}$
  - with hard constraints $\boxed{\begin{array}{l}\min_P f(p,\text{attributes}) \\ \text{under } g(p)=0\end{array}}$
  - with soft constraints $\boxed{\min_P f(p,\text{attributes})-\alpha.g(p)}$

Attributes

- Distance and Geodesic distance





Attributes

- Distance and Geodesic distance
- Planarity, normal direction

Attributes

- Distance and Geodesic distance
- Planarity, normal direction
- Smoothness, curvature

Attributes

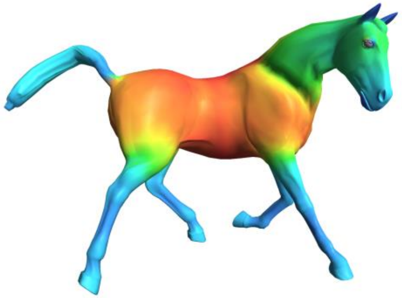
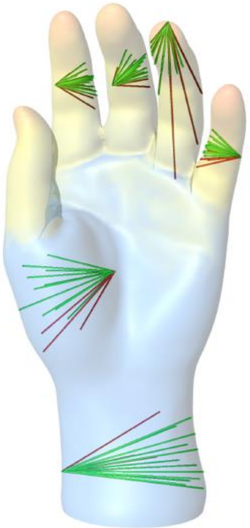
- Distance and Geodesic distance
- Planarity, normal direction
- Smoothness, curvature
- Distance to complex geometric primitives

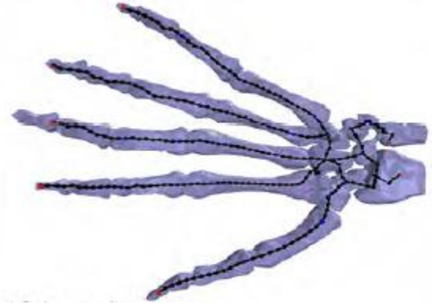
Attributes

- Distance and Geodesic distance
- Planarity, normal direction
- Smoothness, curvature
- Distance to complex geometric primitives
- Symmetry

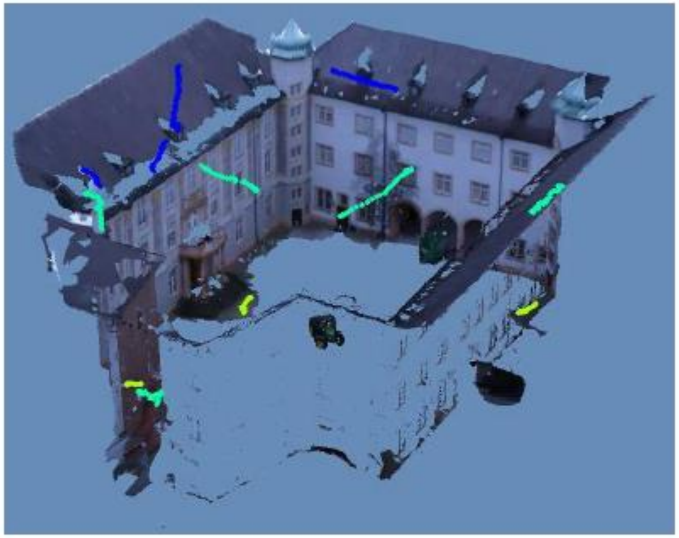
Attributes

- Distance and Geodesic distance
- Planarity, normal direction
- Smoothness, curvature
- Distance to complex geometric primitives
- Symmetry
- Medial Axis, Shape diameter

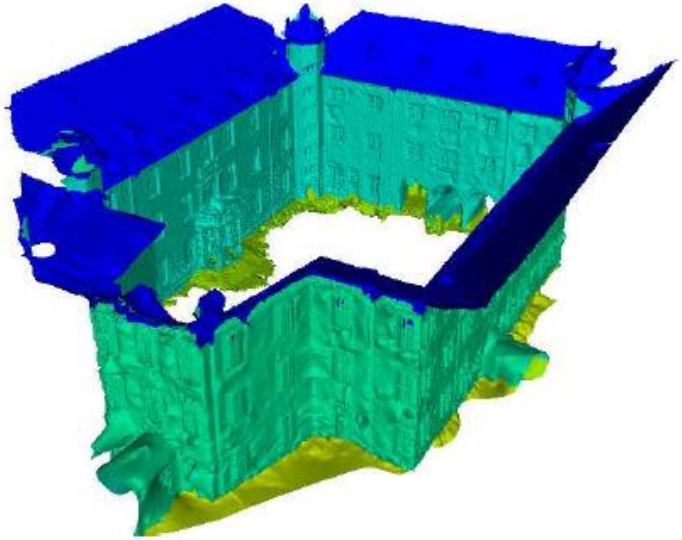





Attributes



- Distance and Geodesic distance
- Planarity, normal direction
- Smoothness, curvature
- Distance to complex geometric primitives
- Symmetry
- Medial Axis, Shape diameter
- Texture
- …

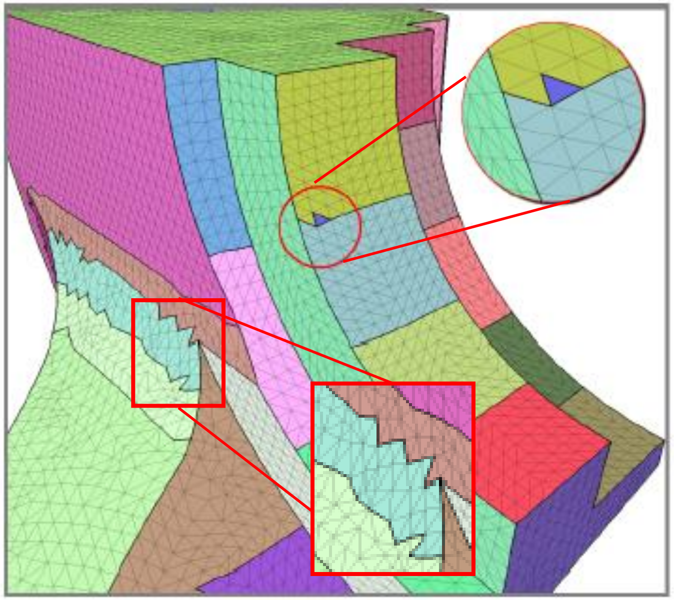


Constraints

- Cardinality
  Not too small/large or a given number of clusters
  Overall balanced partition, ..

- Geometry
  Size: area, diameter, radius
  Convexity, Roundness
  Boundary smoothness, ..

- Connectivity
  Boundary shape,
  Semantic constraints, ..



Contents

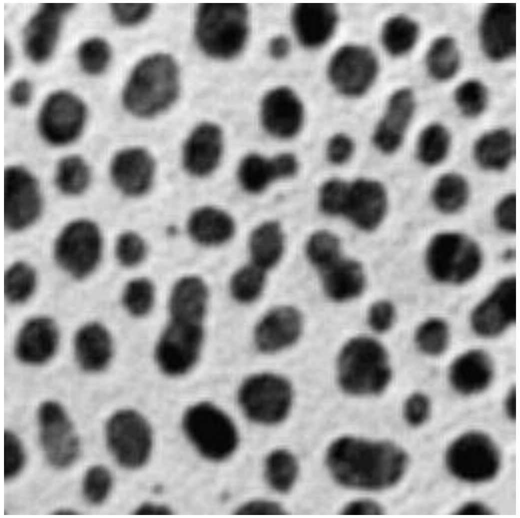
- Attributes and constraints
- Segmentation algorithms

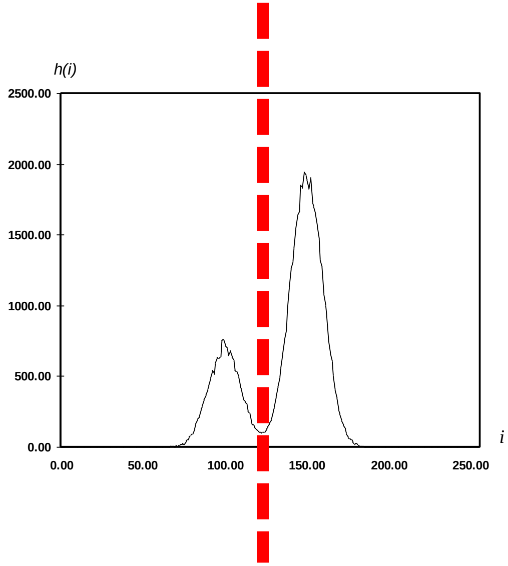
Segmentation algorithms

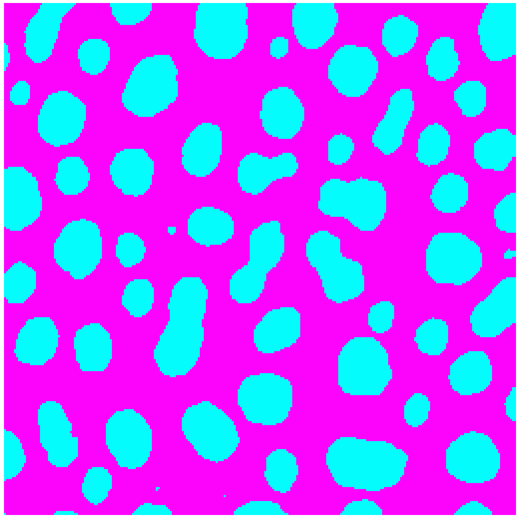
- Very large variety of «mechanisms» in the literature
- Common with image processing/computer vision
- Focus on some of them:
  - Thresholding
  - Region growing
  - Hierarchical partitioning
  - Markov Random Fields
  - …

Thresholding

- simple
- one parameter for a binary segmentation







Hysteresis thresholding

- idea: to keep the sites connected (favor large clusters)
- two parameters for a binary segmentation $t_l < t_h$
- scheme
  - threshold the sites with t_l
  - consider as cluster only the set of connected sites all $> t_l$ with at least one site $> t_h$

Region growing

- see primitive-based surface reconstruction slides

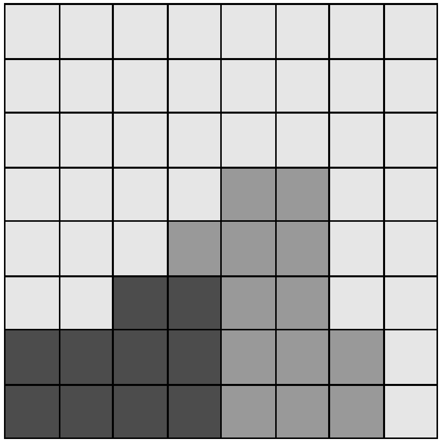
- Mechanism
  - while the mesh is not entirely segmented
    - choose an unlabeled site (a facet)
    - assign label k to its similar adjacent sites
    - iterate on the new assigned sites..
    - when propagation of cluster k finished, update k=k+1

hierarchical partitioning

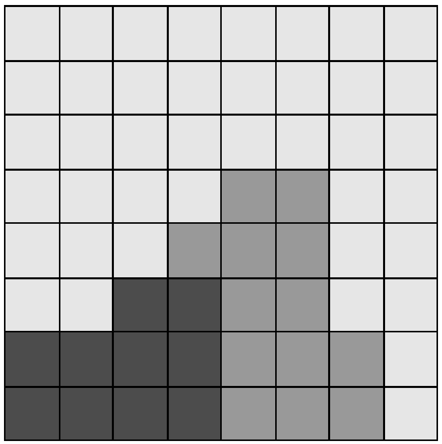
- start from one region representing the entire mesh

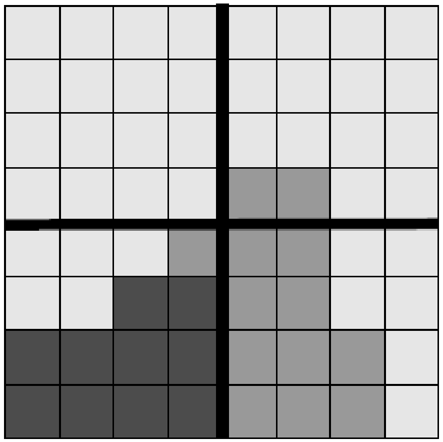
- divide into several regions when the criteria are not valid, and iterate on the new regions

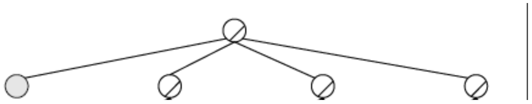
hierarchical partitioning



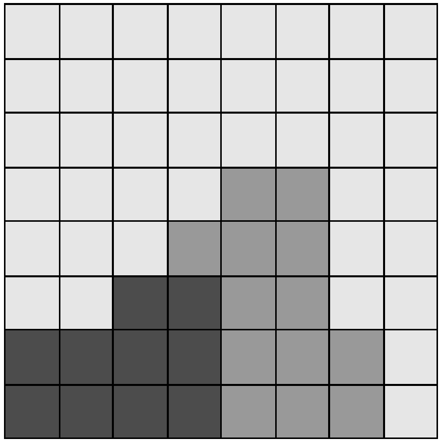
hierarchical partitioning

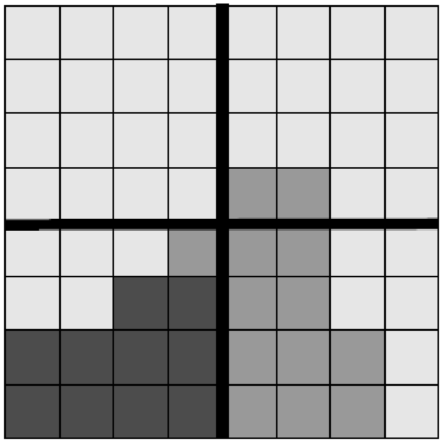


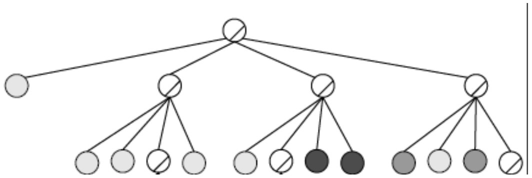


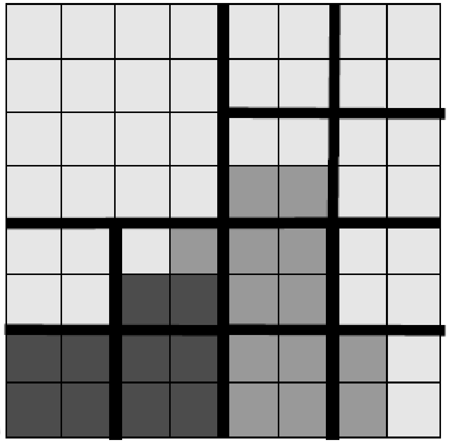


hierarchical partitioning

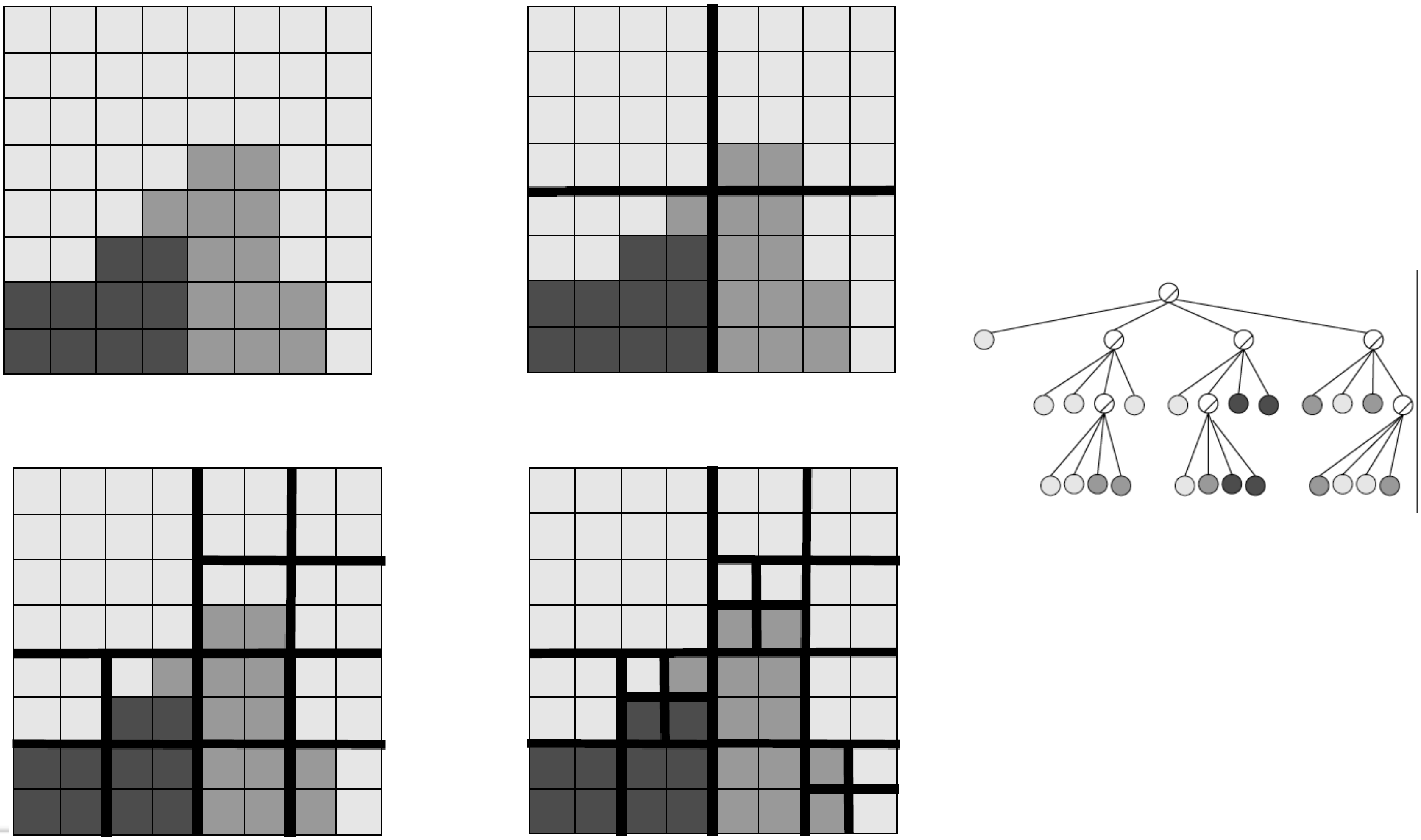








hierarchical partitioning



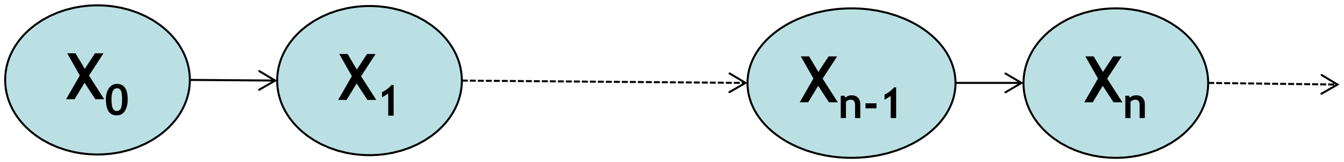
Markov Random Fields (MRF)

set of random variables having a Markov property described by an undirected graph

Markov property

- in 1D (Markov chain)

$$\Pr(X_{n+1}=x|X_1=x_1,X_2=x_2,\ldots,X_n=x_n)=\Pr(X_{n+1}=x|X_n=x_n).$$

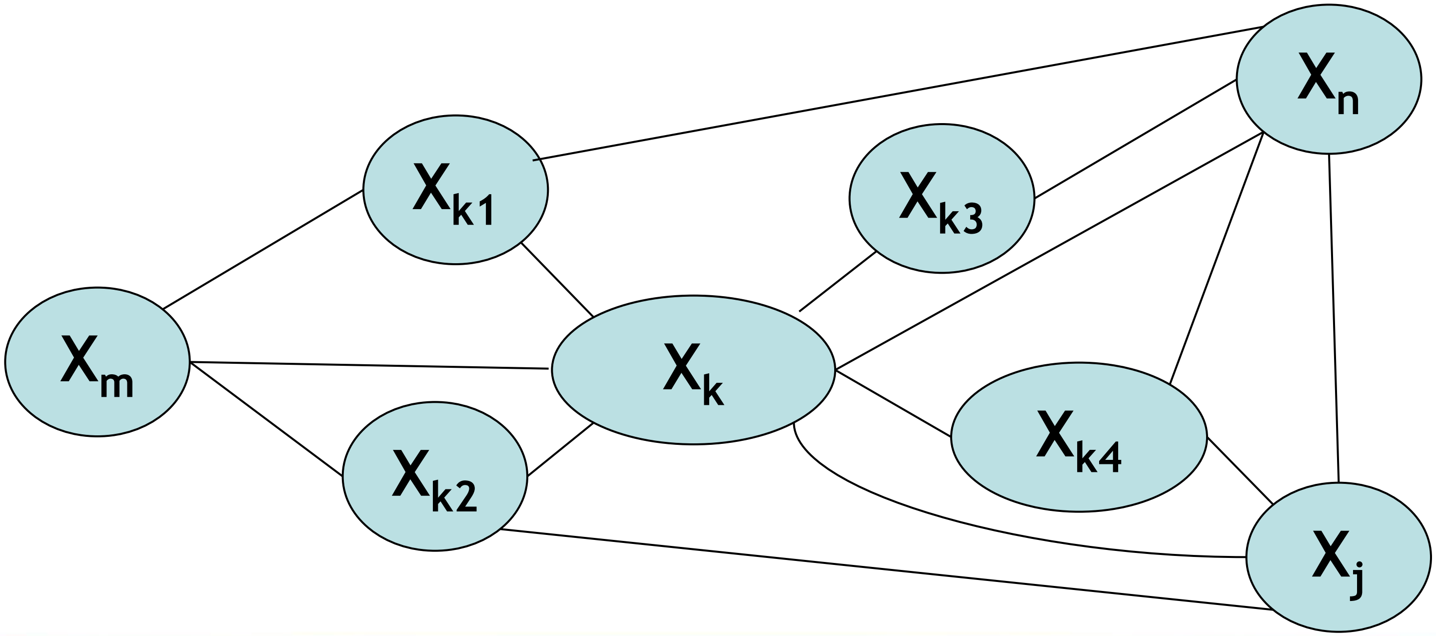


n usually corresponds to time

Markov property

- in 2D or on a manifold in 3D (Markov field)

$$P[\, X_k \,|\, X - \{Xk\} \,] = P[\, X_k \,|\, (X_{n(k)}) \,] \quad \text{with } n(k) \text{ neighbors of } k$$

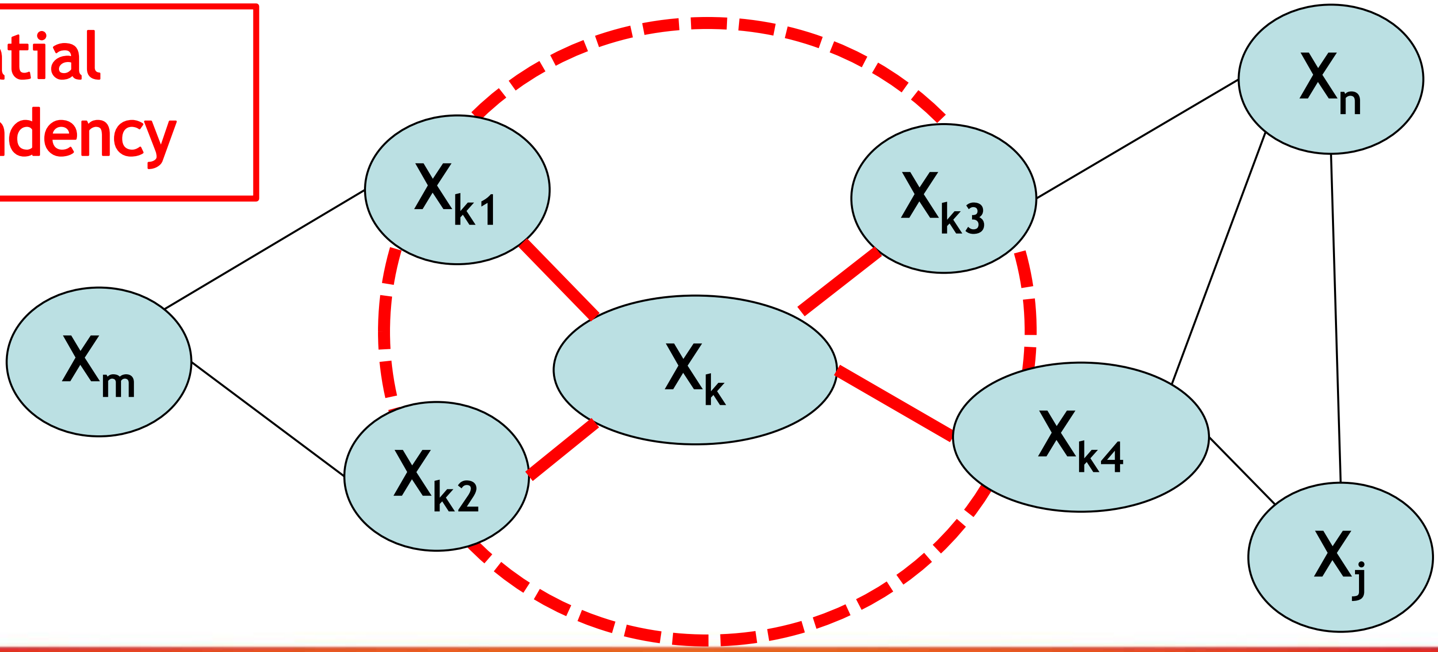


Markov property

- in 2D or on a manifold in 3D (Markov field)

$$P[\, X_k \mid X - \{Xk\} \,] = P[\, X_k \mid (X_{n(k)}) \,] \quad \text{with } n(k) \text{ neighbors of } k$$

Spatial dependency



Markov property

- notion of neighborhood

$V = \{V(s)/s \in S\}$ *est un système de voisinage si*

- $s \notin V(s)$
- $s \in V(t) \iff t \in V(s)$

De plus, on appelle **clique**, *un sous-ensemble* c *de* S *dont les sites sont voisins deux à deux. On note* C, *l'ensemble de toutes les cliques asociées à* (S, V).

- a MRF is always associated to a neighborhood system defining the dependency between graph nodes

Markov property

- Gibbs energy (Hammersley-Clifford theorem)

Soit X *un champ de Markov associé à l'espace probabilisé* $(\Omega, \mathcal{F}, P)$, *et tel que* $\forall x \in \Omega, P(X = x) > 0$.
Alors, la distribution $P(X)$ *de ce champ est une distribution de Gibbs :*

$$P(X) = \frac{\exp -U(X)}{Z}$$

où $U(X) = \sum_{c \in C} V_c(X_s, s \in c)$ *et* $Z = \sum_{X \in \Omega} \exp -U(X)$

Markov property

- **Why is the markovian property important ?**

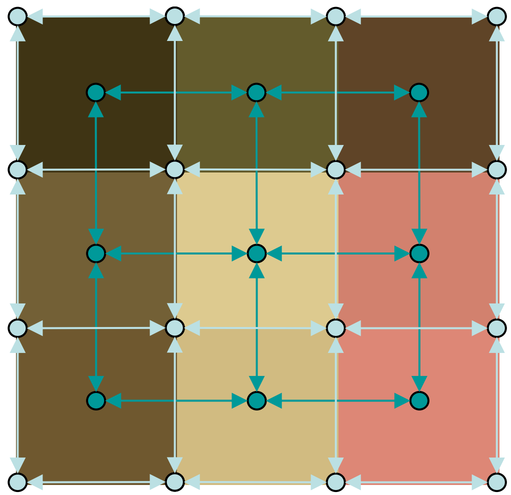
  - graph with 1M nodes
  - if each node is adjacent to every other nodes:
    1M*(999,999)/2 edges ~ 500 G edges

  - each random variable cannot be dependent to all the other ones
    complexity needs to be reduced by spatial considerations

Markov Random Fields (MRF)

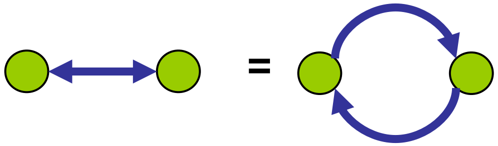
▪ set of random variables having a Markov property described by an undirected graph

▪ for an image, two common graphs



nodes = pixel centers
edges = adjacent pixels (4-connexity)

nodes = pixel corners
edges = pixel borders



Markov Random Fields (MRF)

- set of random variables having a Markov property described by an undirected graph

- for a mesh:

  Graph nodes = vertices & graph edges = edges

  Graph nodes = facets & graph edges = edges

  Graph nodes = edges & graph edges = vertices

Bayesian formulation

Let y, the data (attributes)
x, the label

we want to model the probability of having x knowing y

$$\Pr(X = x \ / \ Y = y) = \frac{\Pr(Y = y \ / \ X = x) \ . \ \Pr(X = x)}{\Pr(Y = y)}$$ **Bayes law**

$$\underbrace{\Pr(X = x \ / \ Y = y)}_{\text{probabilité a posteriori de } x} \propto \underbrace{\Pr(Y = y \ / \ X = x)}_{\text{formation des observations}} \ . \ \underbrace{\Pr(X = x)}_{\text{a priori}}$$

- Standard assumption: the conditional independence of the observation:

$$\Pr(X|Y) = \prod_{p \in D} \Pr\left(x_p \middle| I_p\right)$$

From probability to energy

- data term : local dependency hypothesis
- regularization : soft constraints

$$U(l) = \sum_{i \in V} D_i(l_i) + \beta \sum_{\{i,j\} \in E} V_{ij}(l_i, l_j)$$

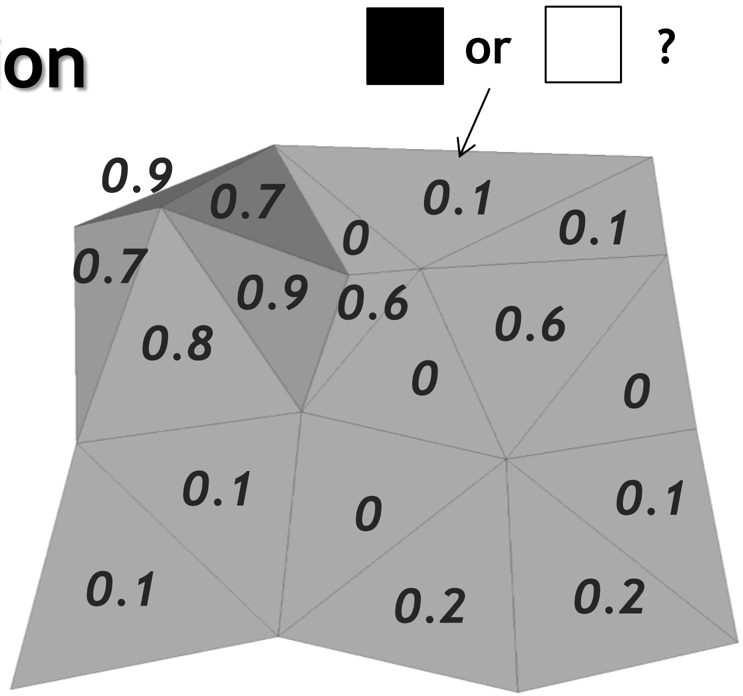
Optimal configuration

We search for the label configuration x that maximizes P(X=x /Y=y)

$$\rightarrow \quad x^{*} = \arg\max_{x} \Pr(X=x|Y=y)$$

$$= \arg\min_{x} U(x)$$

exercise: binary segmentation



Graph structure

Graph nodes = facets

Graph edges = common edges

Attributes on facet: *[0,1]* (y)

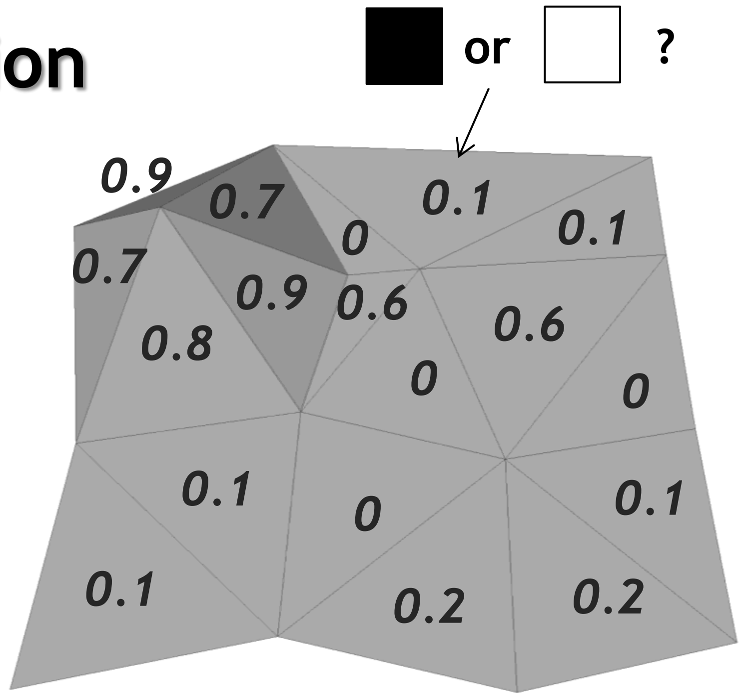
labels: {*white*, *black*} (l)

Energy: $U(l) = \sum_{i \in V} D_i(l_i) + \beta \sum_{\{i,j\} \in E} V_{ij}(l_i, l_j)$

with $D_i(l_i) = \begin{cases} y_i & \text{if } l_i = \text{'white'} \\ 1 - y_i & \text{otherwise} \end{cases}$

$$V_{i,j}(l_i, l_j) = \begin{cases} 0 & \text{if } \; l_i = l_j \\ 1 & \text{otherwise} \end{cases}$$

exercise: binary segmentation



Graph structure

Graph nodes = facets
Graph edges = common edges

Attributes on facet: *[0,1]* (y)

labels: {*white*, *black*} (l)

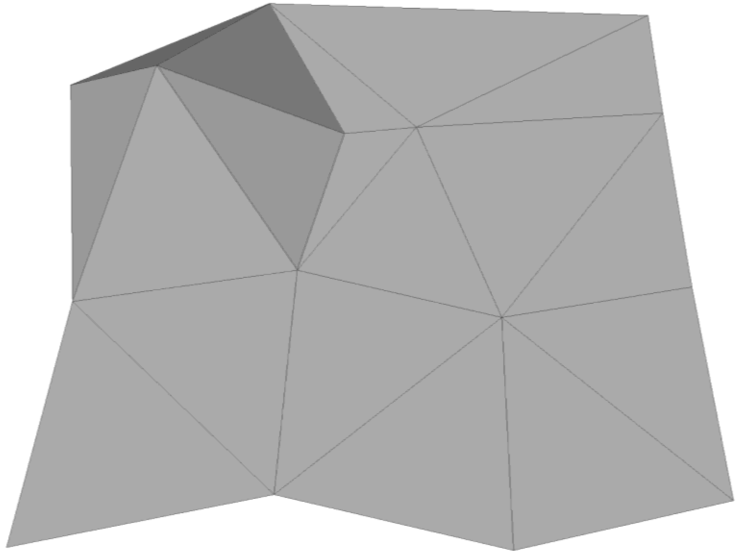
Energy: $U(l) = \sum_{i \in V} D_i(l_i) + \beta \sum_{\{i,j\} \in E} V_{ij}(l_i, l_j)$

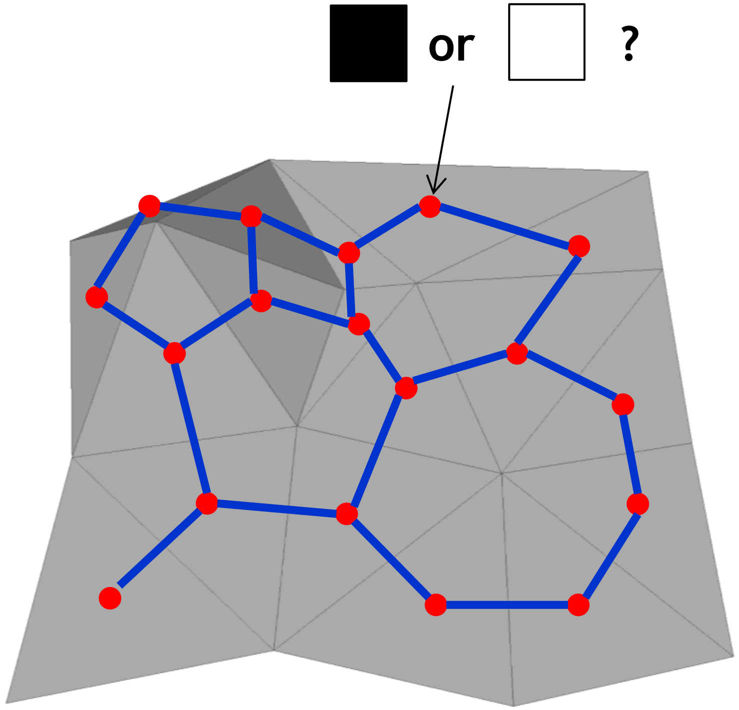
with $D_i(l_i) = \begin{cases} y_i & \text{if } l_i = \text{'white'} \\ 1 - y_i & \text{otherwise} \end{cases}$

$V_{i,j}(l_i, l_j) = \begin{cases} 0 & \text{if } l_i = l_j \\ 1 & \text{otherwise} \end{cases}$

Q1: what is the optimal configuration l if β =0 ? What is its energy ?
Q2: what is the optimal configuration if $\beta \rightarrow \inf$?
Q3: what are the other possible optimal configurations in function of β ?

exercise: binary segmentation





Finding the optimal configuration of labels

Graph-cut based approches

fast but limitations on energy formulation

Monte Carlo sampling

slow but no limitation

Example: mesh segmentation with principal curvature attributes & soft geometric constraints

Multi-label energy model of the form

$$U(l) = \sum_{i \in V} D_i(l_i) + \beta \sum_{\{i,j\} \in E} V_{ij}(l_i, l_j)$$

with V, set of vertices of the input mesh

E, set of edges in the mesh

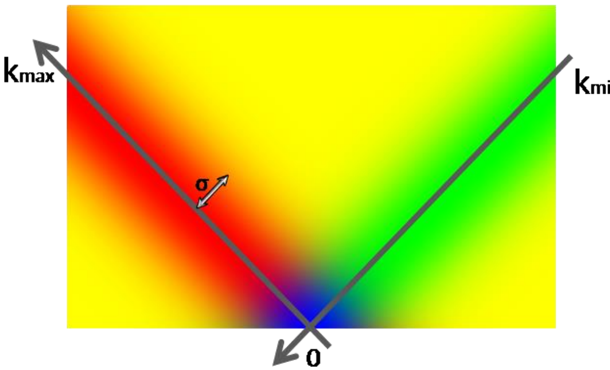
l_i, the label of the vertex i among : *planar* (1), *developable convex* (2), *developable concave* (3) and *non developable* (4)

Data term

$$D_i(l_i) = 1 - Pr(l_i | k_{min}^{(i)}, k_{max}^{(i)})$$

with

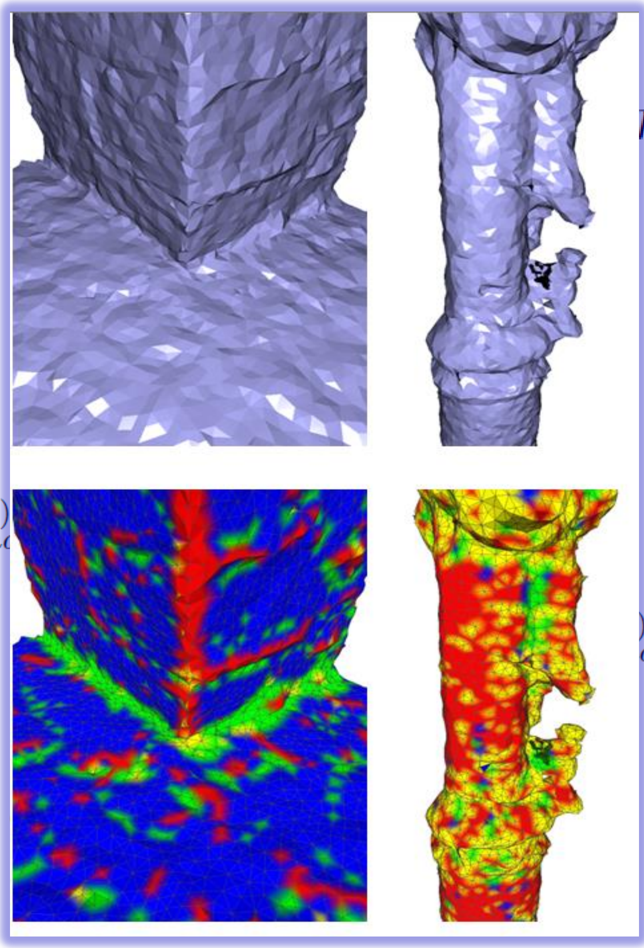
$$Pr(l_i | k_{min}^{(i)}, k_{max}^{(i)}) = \begin{cases} G_\sigma(k_{min}^{(i)}) G_\sigma(k_{max}^{(i)}) & \text{if } l_i = 1 \\ G_\sigma(k_{min}^{(i)})(1 - G_\sigma(k_{max}^{(i)})) & \text{if } l_i = 2 \\ (1 - G_\sigma(k_{min}^{(i)})) G_\sigma(k_{max}^{(i)}) & \text{if } l_i = 3 \\ (1 - G_\sigma(k_{min}^{(i)}))(1 - G_\sigma(k_{max}^{(i)})) & \text{if } l_i = 4 \end{cases}$$



$$G_\sigma(k) = \exp(-k^2 / 2\sigma^2)$$

Data term

$$r(l_i|k_{min}^{(i)}, k_{max}^{(i)})$$



with

$$Pr(l_i|k_{min}^{(i)}, k_{m}$$

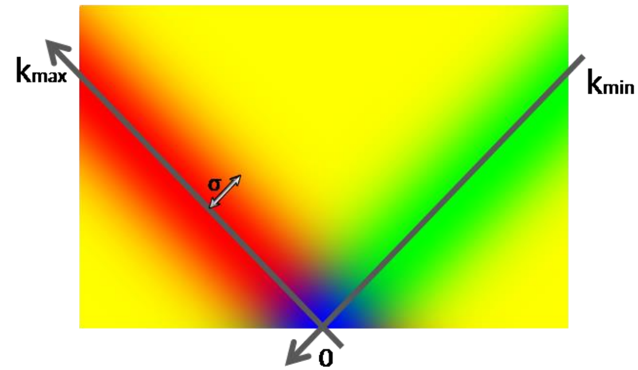
$${}_{ax}))$$

if $l_i = 1$

if $l_i = 2$

if $l_i = 3$

if $l_i = 4$

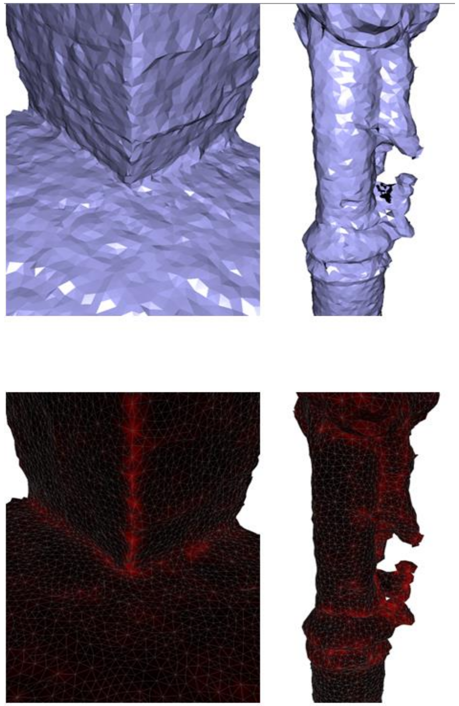


$$G_\sigma(k) \ =$$

Soft constraints

Label smoothness

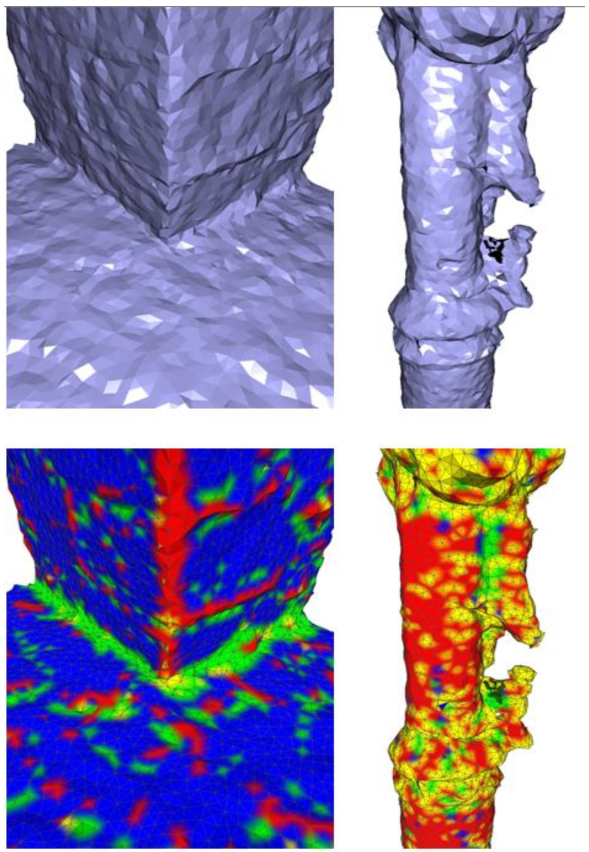
Edge preservation

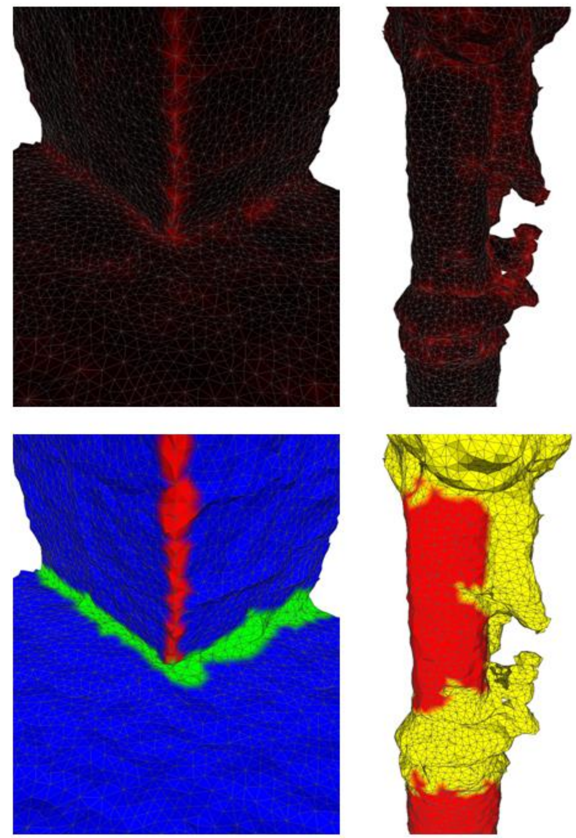


$$V_{ij}(l_i, l_j) = \begin{cases} 1 & \text{if } l_i \neq l_j \\ \min(1, a\|\mathbf{W_i} - \mathbf{W_j}\|_2) & \text{otherwise} \end{cases}$$

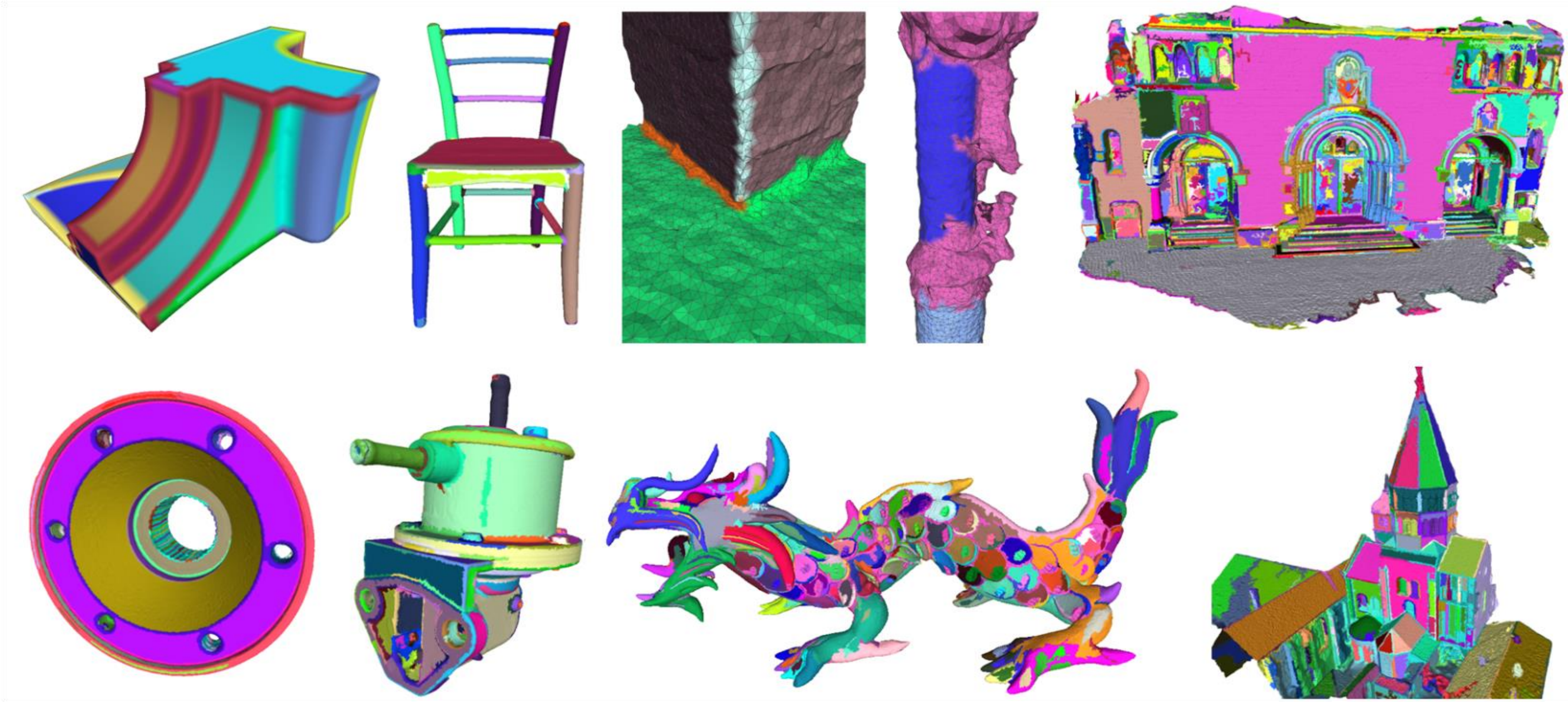
with $\mathbf{W} = \begin{pmatrix} k_{min}.\mathbf{w_{min}} \\ k_{max}.\mathbf{w_{max}} \end{pmatrix}$

Optimization



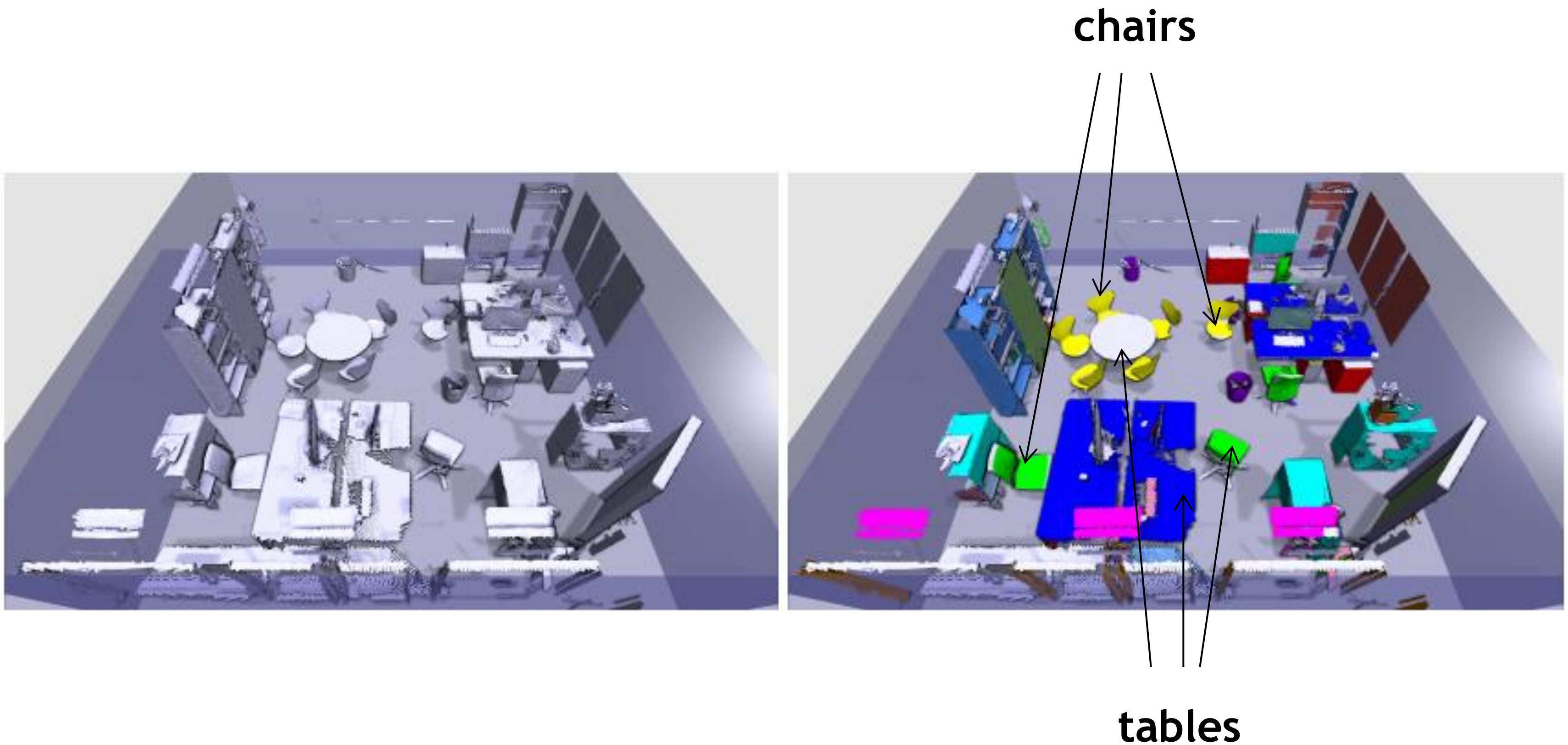


Some results



What is object recognition?

What is object recognition?



What is object recognition?

Input: CAD databases (eg 3D warehouse)
or point clouds (eg kinect, laser)

Assumption: objects have been segmented. We want to find the nature of each object

Contents

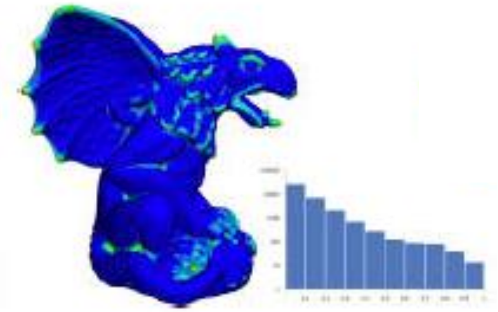
- Feature extraction (local and/or global)
- Classification

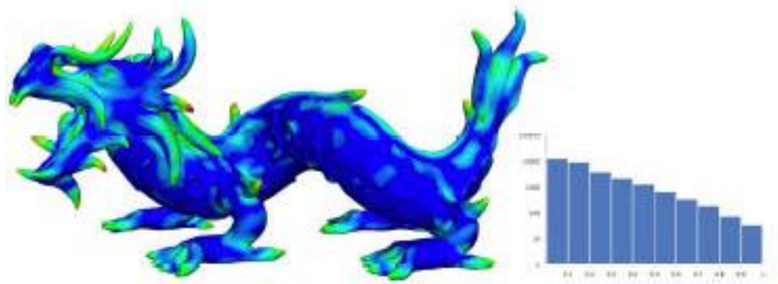
Local features (one data point = one feature)

- Distance and Geodesic distance
- Planarity, normal direction
- Smoothness, curvature
- Distance to complex geometric primitives
- Symmetry
- Medial Axis, Shape diameter
- Texture
- …

Global features (one object = one feature)

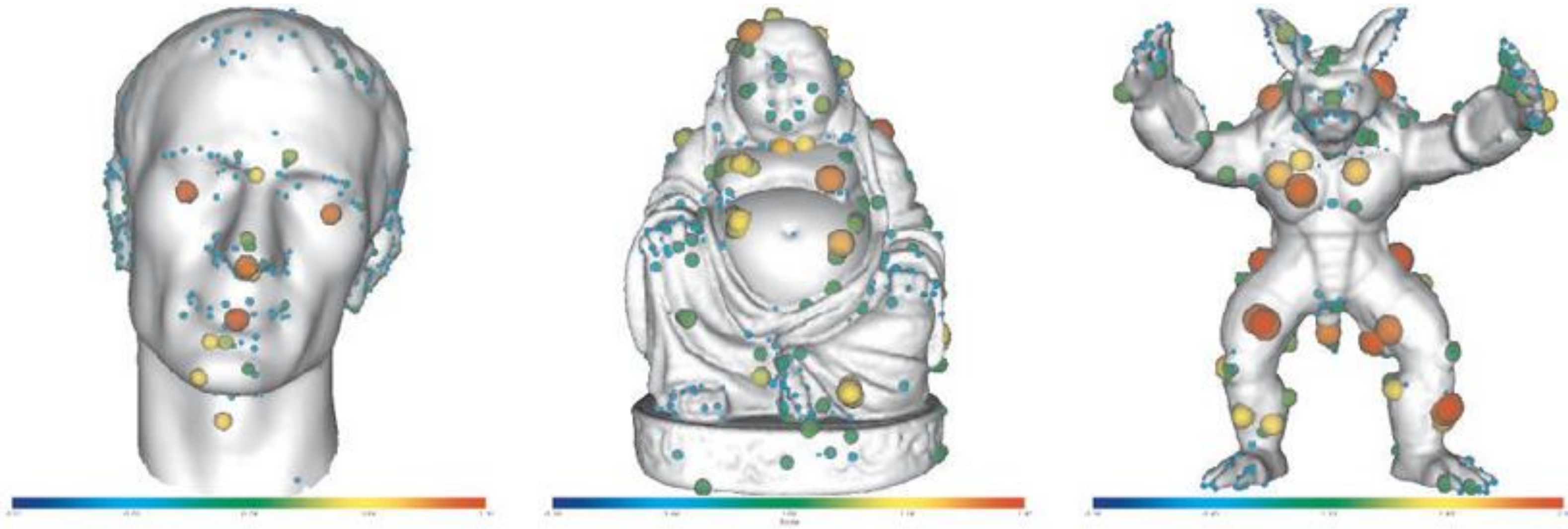
- distribution measures of local features (eg histograms)





Keypoint features (one object= several keypoints)

- Local features detected at some specific locations of the object



Contents

- Feature extraction (local and/or global)
- Classification

Machine learning techniques (overview)

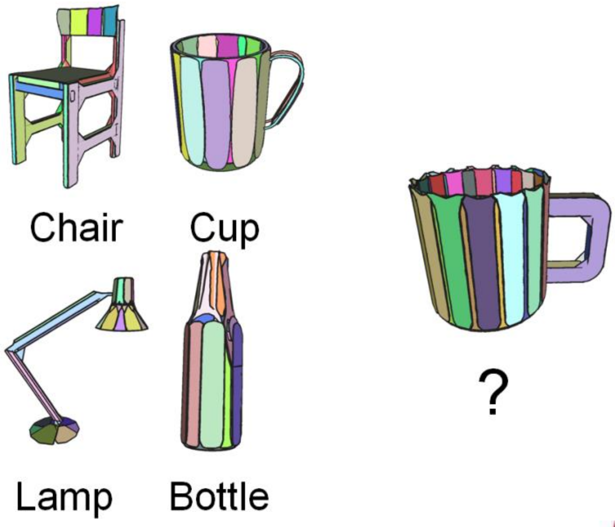
- **K-means clustering:** An unsupervised algorithm that learns which things 'go together'. User has to specify K.
- **Bayes Classifier:** Assumes features are Gaussian distributed and independent of each other. For each class find mean and variance of its attributes. Then, given some attributes compute the probability that it is a member of each class and take the most probable one. Works surprisingly well and can handle large data sets.
- **Decision Trees:** Finds data features and thresholds that best splits the data into separate classes. This is repeated recursively until data has been split into homogeneous (or mostly homogeneous) groups. Can immediately identify the features that are most important.
- **Boosting:** A collection of weak classifiers (typically single level decision trees). During training each classifier learns a weight for its vote from its accuracy on the data. Each classifier is trained one by one, data that is poorly represented by earlier classifiers is given a higher weighting so that subsequent classifiers pay more attention to points where the errors are large.
- **Random Forests:** An ensemble of decision trees. During learning tree nodes are split using a random subset of data features. All trees vote to produce a final answer. Can be one of the most accurate techniques.

Machine learning techniques (overview)

- **Expectation maximization (EM) Maximum Likelihood Estimation (MLE):** Typically we assume the data is a mixture of Gaussians. In this case EM fits N multidimensional Gaussians to the data. User has to specify N.
- **Neural Networks / Multilayer Perceptron:** Slow to train but fast to run, design is a bit of an art but can the the best performer on some problems.
- **Support Vector Machines:** Algorithm finds hyperplanes that maximally separates classes. Projecting the data into higher dimensions makes the data more likely to be linearly separable. Works well when there is limited data.

Example: object recognition from point cloud databases via planar abstraction

- Provides robustness against defective data
  - Rely on robust shape detection methods
- Invariance to rotation and scale
  - Up vector not required
- Meaningful global descriptor
  - Function constrains object shape



Example: object recognition from point cloud databases via planar abstraction

- Object descriptor
  - Feature vector

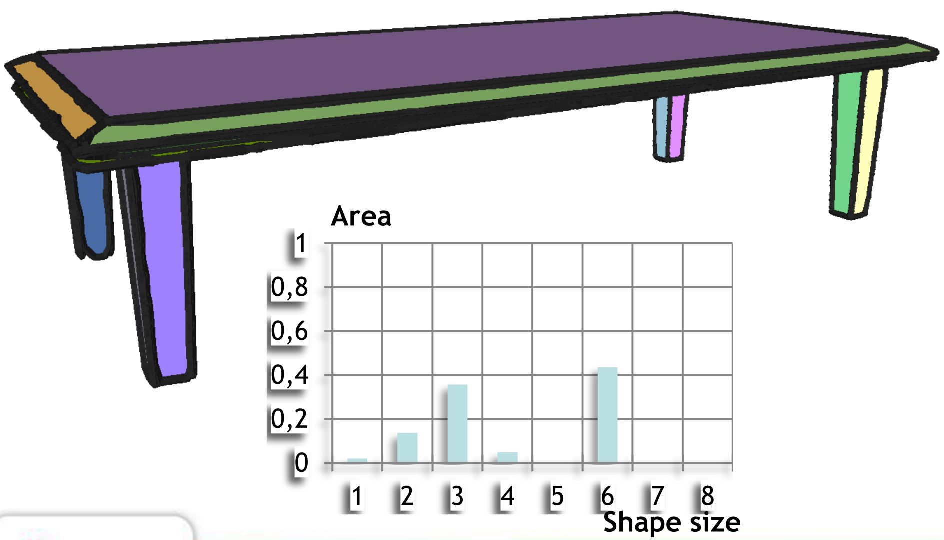
$$x = (x_1, x_2, \ldots, x_n) \in R^n$$

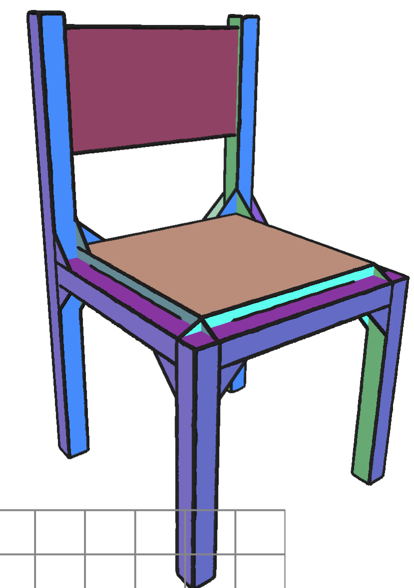
  - Geometric features
    - Histograms
  - Training and classification via Random Forest

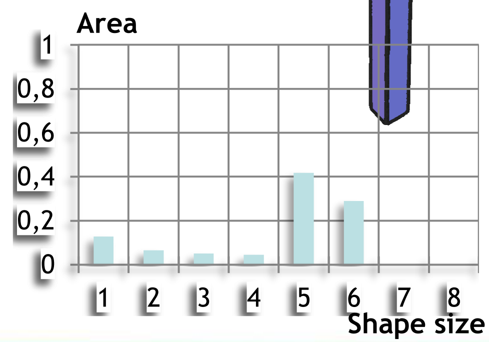
Example: object recognition from point cloud databases via planar abstraction

Features

- Area fragmentation
  - Many small shapes
  - Few large shapes



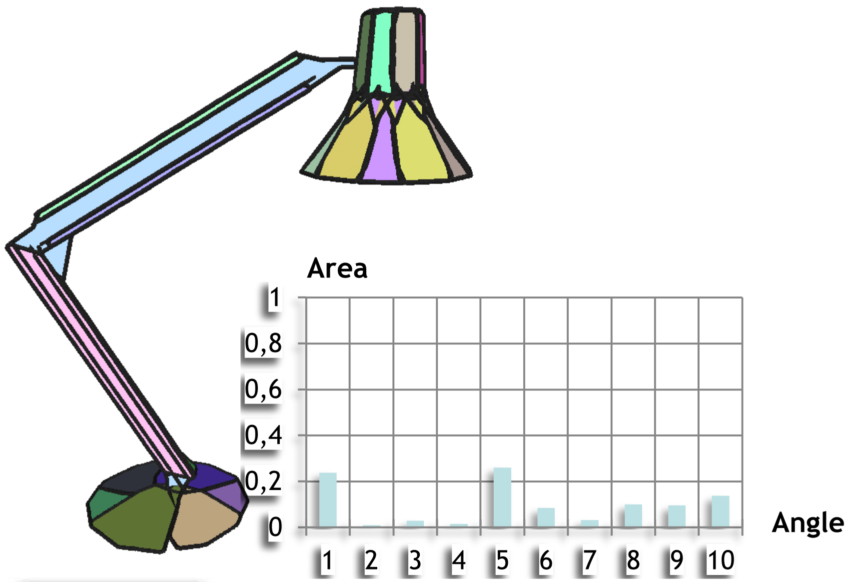


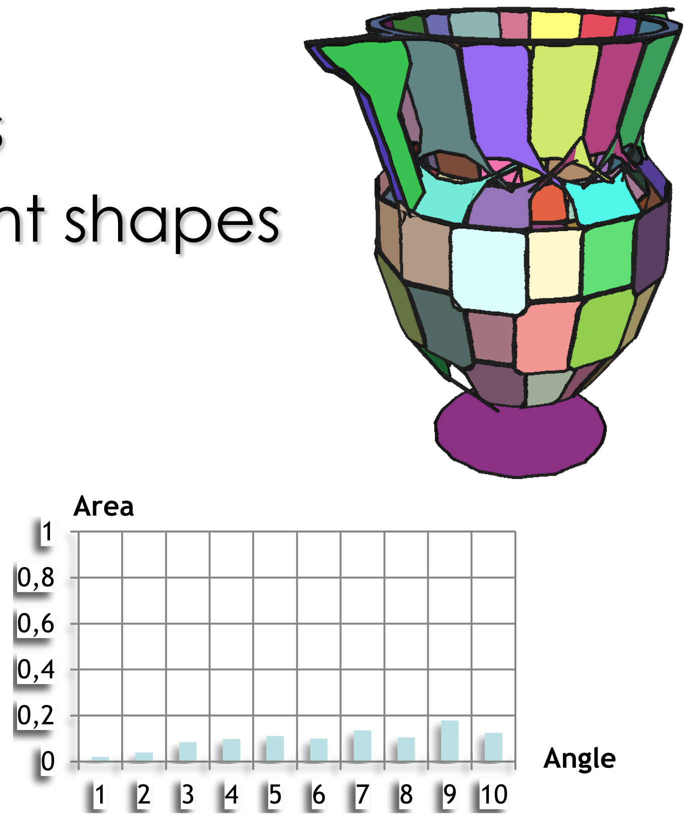


Example: object recognition from point cloud databases via planar abstraction

Features

- Pairwise orientation of shapes
  - Relative orientation
  - Histogram 0-90 degrees
- Pairwise angles of adjacent shapes

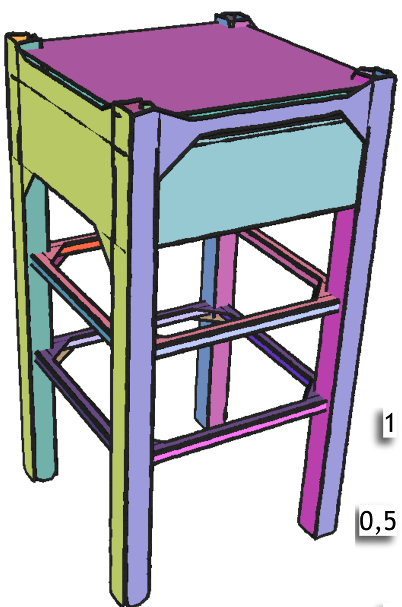


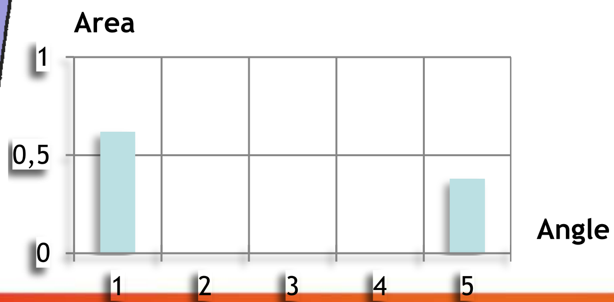


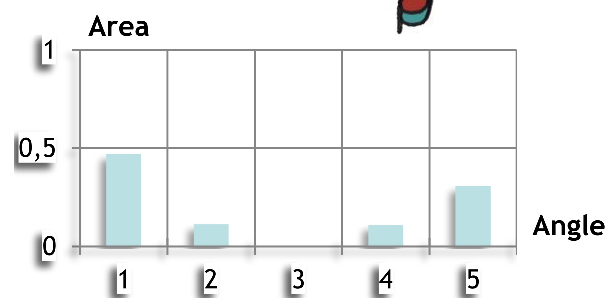
Example: object recognition from point cloud databases via planar abstraction

Features

- Histogram of slope
  - Angle between shape and reference vector
  - Major axis of oriented bounding box







Example: object recognition from point cloud databases via planar abstraction

- Multi-scale
  - Different kind of detail for objects
  - Non planar shapes depend on detection parameters
- Histogram features performed on 3 scales
  - 2x and 4x of basis scale

Example: object recognition from point cloud databases via planar abstraction

Confution matrix

Increasing defects

Planar abstraction

(a)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	2	0	0	0	1	0	0	2
Chair	0	4	1	0	0	0	0	0
Couch	0	0	5	0	0	0	0	0
Lamp	0	1	0	3	1	0	0	0
Mug	0	0	0	0	4	0	0	1
Shelf	0	0	0	0	0	5	0	0
Table	0	0	0	0	0	0	5	0
Vase	0	0	0	0	0	0	0	5

(d)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	2	0	0	0	1	0	0	
Chair	1	4	0	0	0	0	0	0
Couch	0	1	3	0	0	0	1	0
Lamp	0	0	0	4	1	0	0	0
Mug	0	0	0	0	4	0	0	1
Shelf	0	0	0	0	0	5	0	0
Table	0	0	0	0	0	0	5	0
Vase	1	0	0	0	0	0	0	4

(g)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	4	0	0	0	0	0	0	1
Chair	0	2	1	1	0	1	0	0
Couch	0	1	4	0	0	0	1	0
Lamp	0	0	1	3	0	0	0	0
Mug	2	0	0	0	1	0	0	2
Shelf	0	0	0	0	0	5	0	0
Table	0	0	0	0	0	0	5	0
Vase	0	0	0	1	0	0	0	4

Point-based (Osada)

(b)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	4	0	0	0	0	0	0	1
Chair	0	4	0	0	0	0	1	0
Couch	0	0	3	0	0	0	2	0
Lamp	0	0	0	5	0	0	0	0
Mug	0	0	1	0	4	0	0	0
Shelf	0	0	1	0	0	4	0	0
Table	0	0	0	0	1	2	2	0
Vase	0	1	0	0	0	0	0	4

(e)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	5	0	0	0	0	0	0	0
Chair	0	4	0	0	0	0	1	0
Couch	1	0	4	0	0	0	0	0
Lamp	0	0	0	5	0	0	0	0
Mug	1	0	0	0	3	0	0	1
Shelf	1	0	1	0	0	2	1	0
Table	2	1	0	0	0	0	2	0
Vase	0	1	0	0	2	0	0	2

(h)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	5	0	0	0	0	0	0	0
Chair	0	3	0	0	0	0	2	0
Couch	0	0	5	0	0	0	0	0
Lamp	0	0	0	4	0	0	0	1
Mug	0	0	0	0	1	0	0	4
Shelf	0	0	0	0	1	4	0	0
Table	0	0	2	0	0	2	0	1
Vase	1	0	0	0	1	0	0	3

PCL ESF-feature

(c)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	5	0	0	0	0	0	0	0
Chair	0	3	1	0	0	0	1	0
Couch	0	0	4	0	0	1	0	0
Lamp	0	0	0	4	1	0	0	0
Mug	0	0	0	0	5	0	0	0
Shelf	0	0	1	0	0	4	0	0
Table	0	1	1	1	0	0	2	0
Vase	1	1	0	0	1	0	0	2

(f)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	4	0	1	0	0	0	0	0
Chair	0	2	0	1	0	1	1	0
Couch	0	0	4	0	0	0	1	0
Lamp	2	1	0	1	0	0	0	1
Mug	0	2	0	1	1	0	0	1
Shelf	0	0	0	1	0	4	0	0
Table	0	0	3	0	0	0	2	0
Vase	0	0	0	0	1	0	0	4

(i)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	3	0	0	0	1	0	1	0
Chair	0	3	0	0	1	0	0	1
Couch	2	0	3	0	0	0	0	0
Lamp	1	0	0	2	2	0	0	0
Mug	0	1	0	2	1	0	0	1
Shelf	2	1	0	0	0	2	0	0
Table	1	1	1	0	0	0	2	0
Vase	1	0	0	0	2	0	0	2