Analysis of 3D data

Florent Lafarge
Inria Sophia Antipolis - Mediterranee

Contents

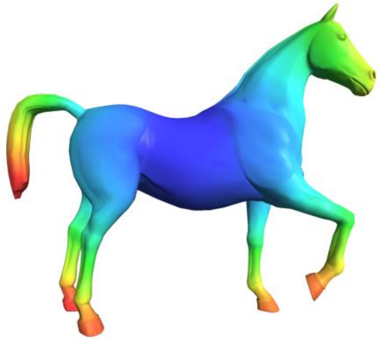
- Local analysis of 3D data
- Segmentation and classification of 3D data

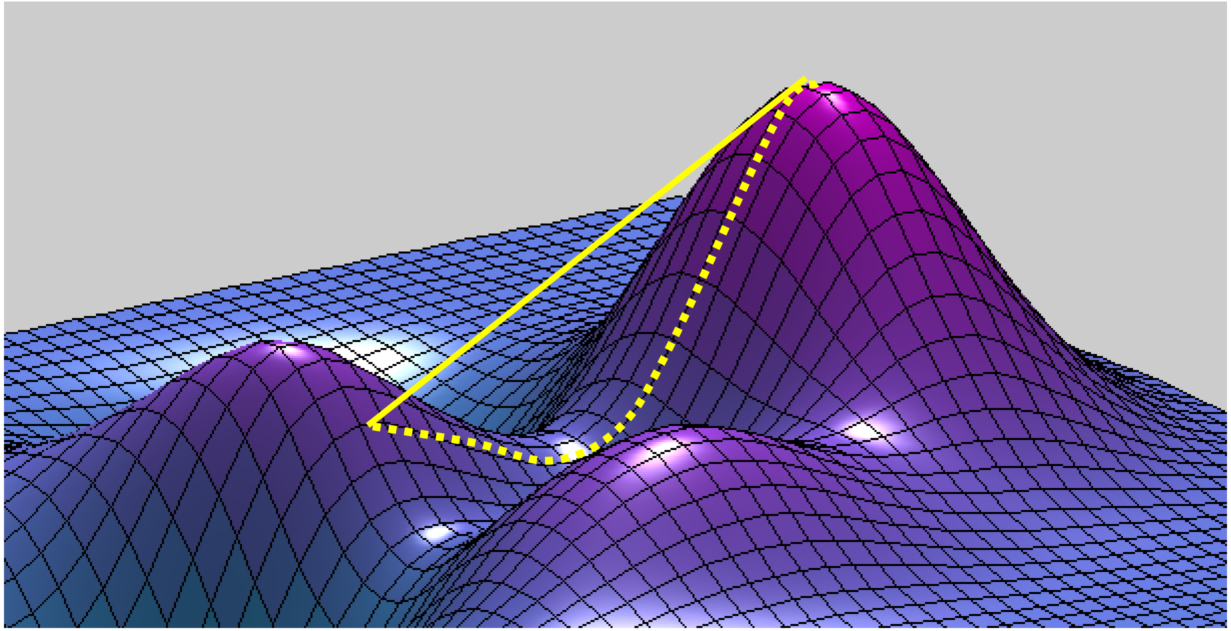
Local analysis of 3D data

- Geometric attributes
- Advanced 3D descriptors

Attributes

- Distance and Geodesic distance



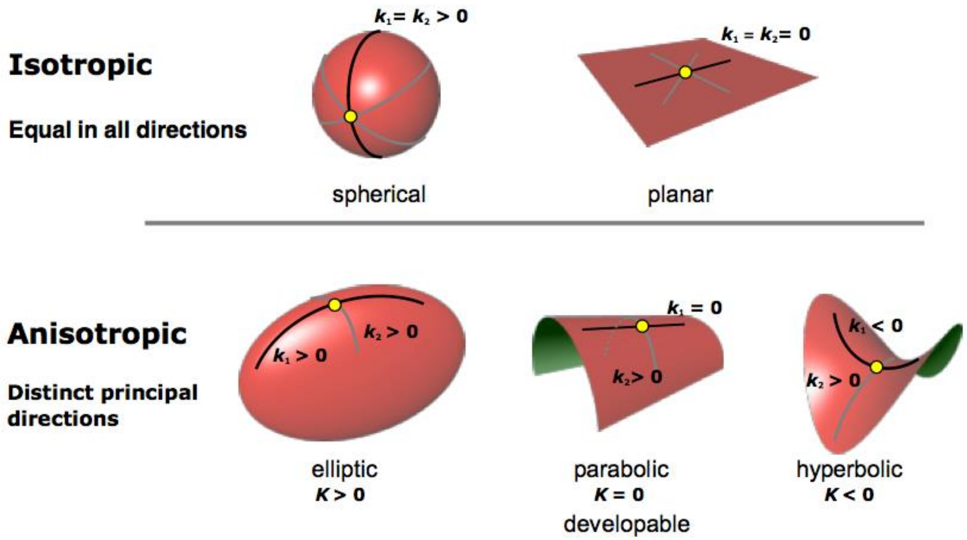


Attributes

- Distance and Geodesic distance
- Planarity, normal direction

Attributes

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



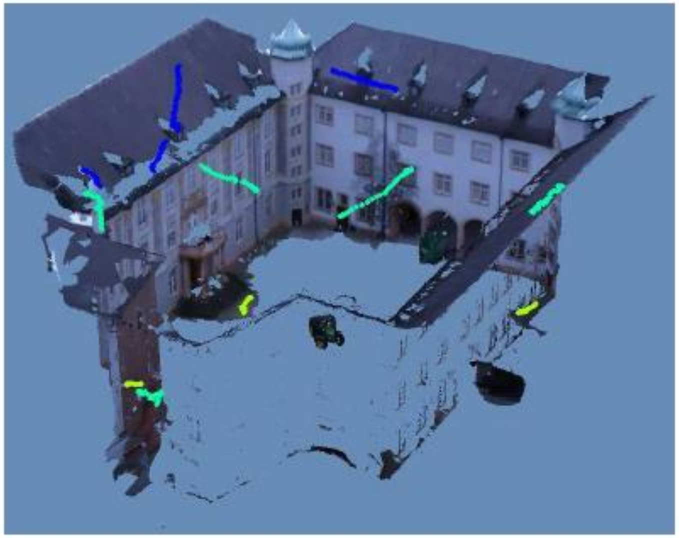
Attributes

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

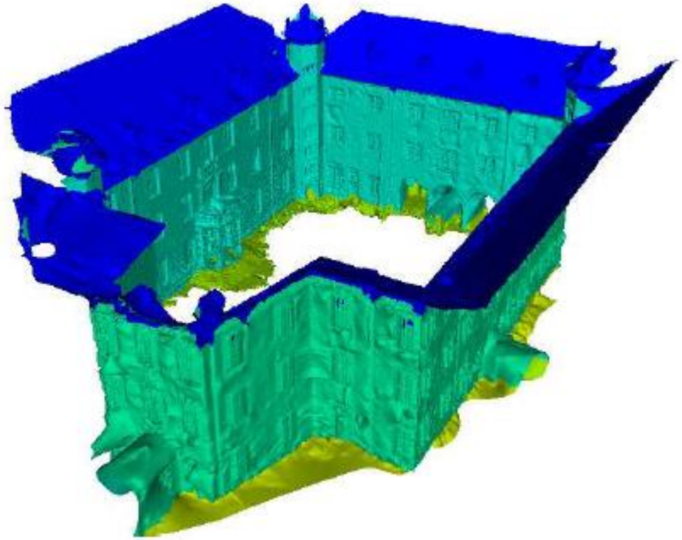
Attributes

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

Attributes

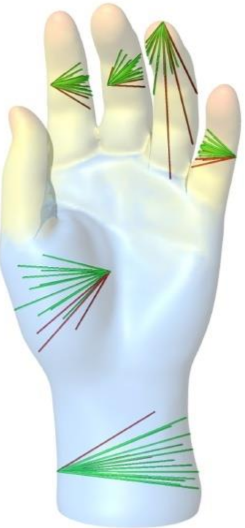


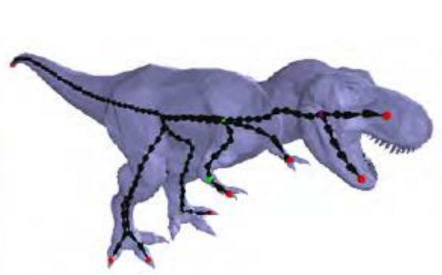
- Distance and Geodesic distance
- Planarity, normal direction
- Smoothness, curvature
- Distance to complex geometric primitives
- Symmetry
- Medial Axis, Shape diameter
- Texture
- …

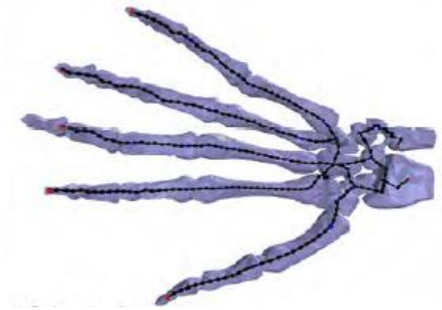


Attributes

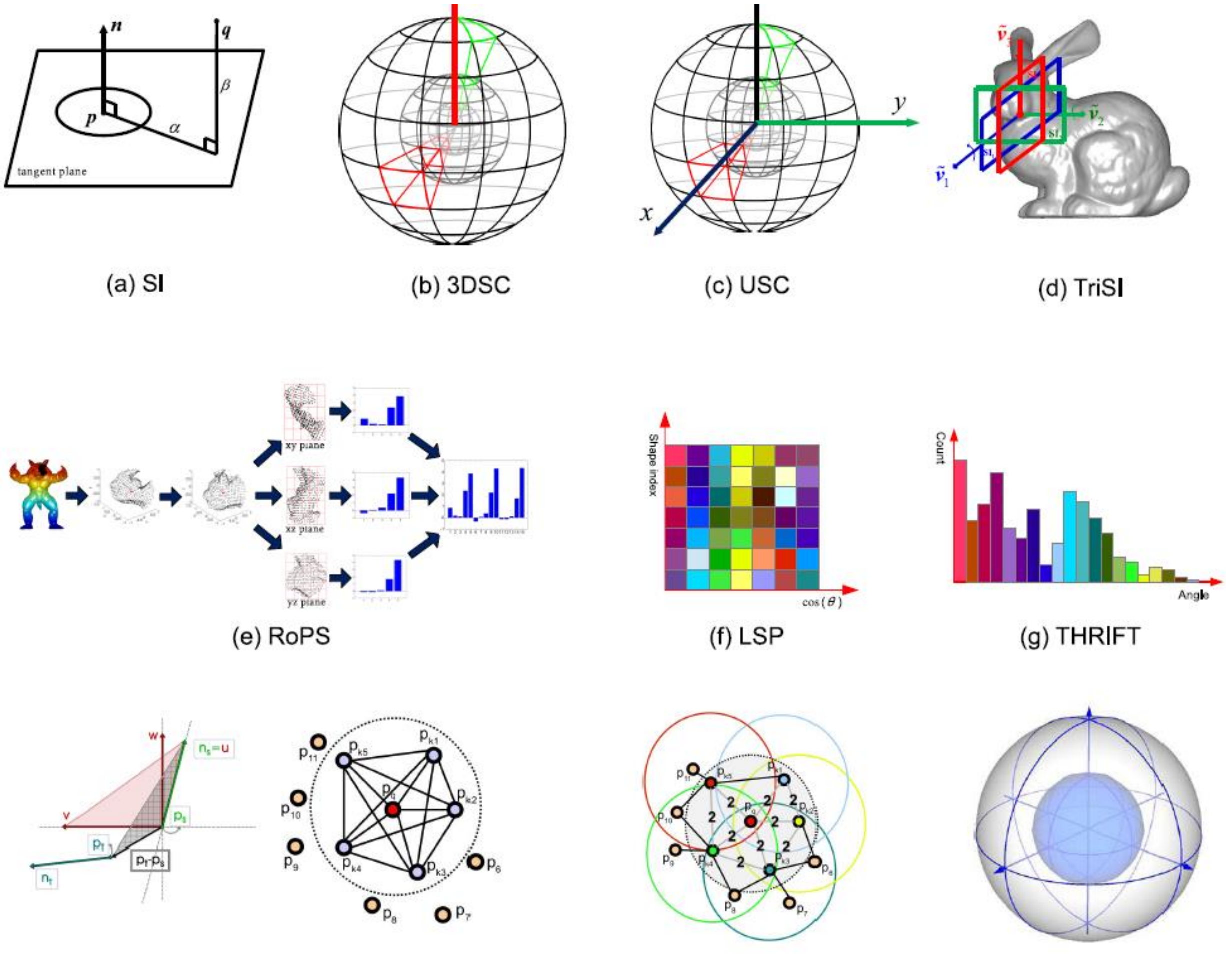
- Distance and Geodesic distance
- Planarity, normal direction
- Smoothness, curvature
- Distance to complex geometric primitives
- Symmetry
- Medial Axis, Shape diameter







3D descriptors



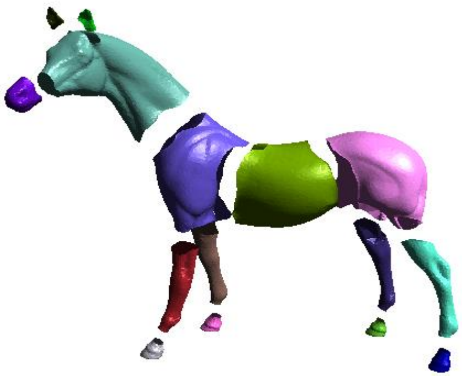
(a) SI (b) 3DSC (c) USC (d) TriSI

(e) RoPS (f) LSP (g) THRIFT

(h) PFH (i) FPFH (j) SHOT

Segmentation and classification

- Unsupervised (MRF)
- Machine learning (Random Forest)
- Deep learning (PointNet)



Markov Random Fields (MRF)

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

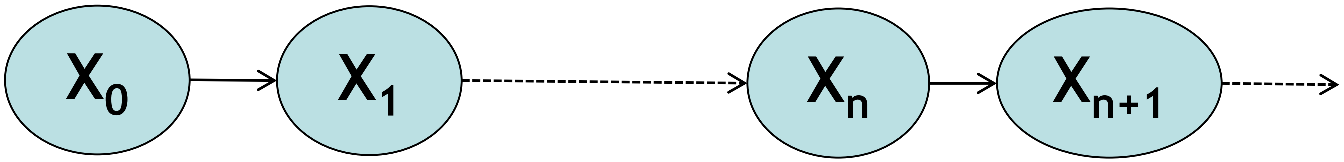
Let V be the set of nodes in the graph

Card(V) = number of random variables in the MRF

Markov property

- in 1D (Markov chain)

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

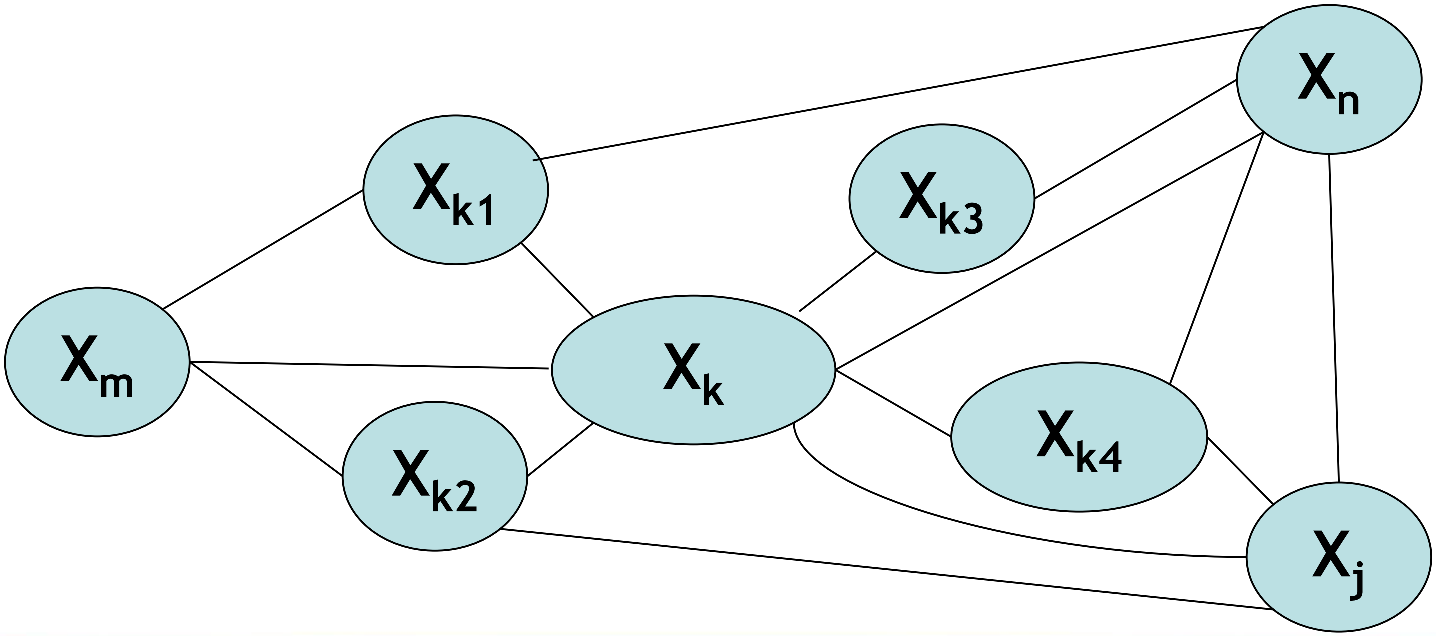


n usually corresponds to time

Markov property

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

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

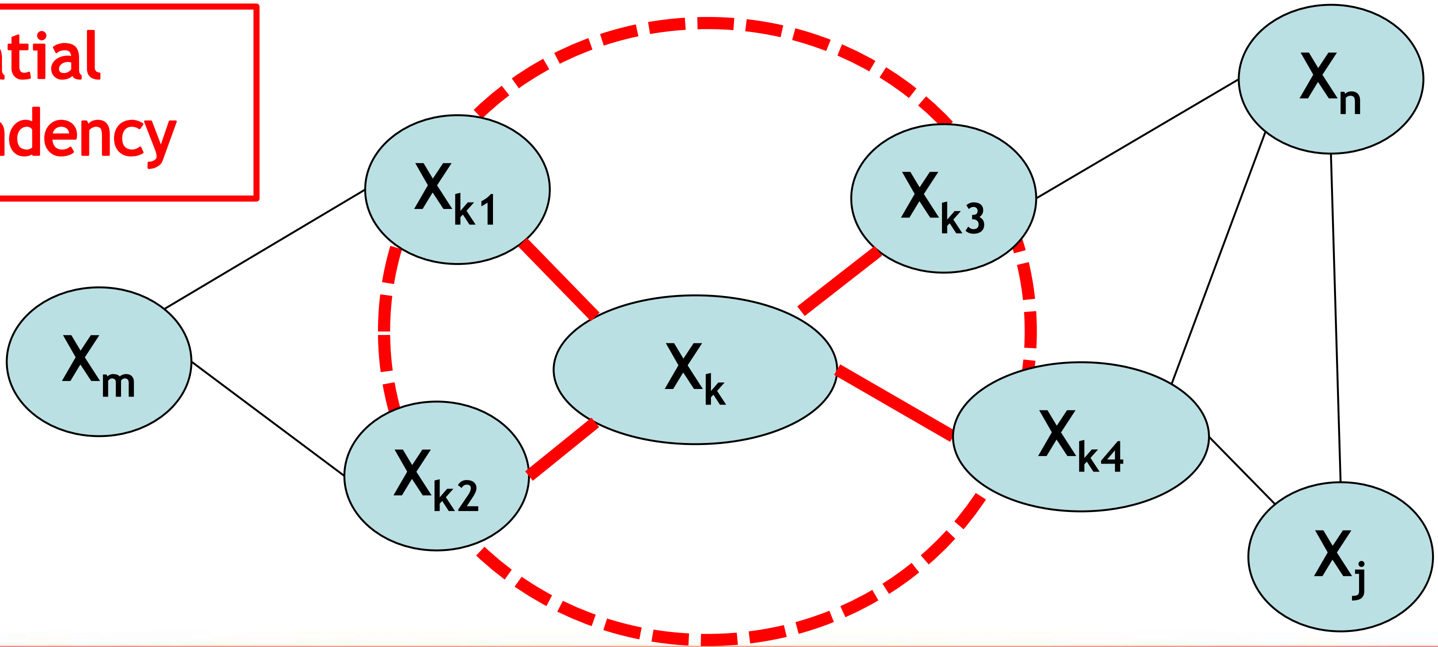


Markov property

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

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

Spatial dependency



Notion of neighborhood

$\mathcal{N} = \{n(i) \,/ i \in V\}$ is a neighborhood system if

- (a) $i \notin n(i)$
- (b) $i \in n(j) \Leftrightarrow j \in n(i)$

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

MRF as an energy

- Gibbs energy (Hammersley-Clifford theorem)

- Let X be a MRF so that for all $x \in \Omega$, $P(X=x)>0$,
Then P(X) is a Gibbs distribution of the form

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

- U is called a Gibbs energy

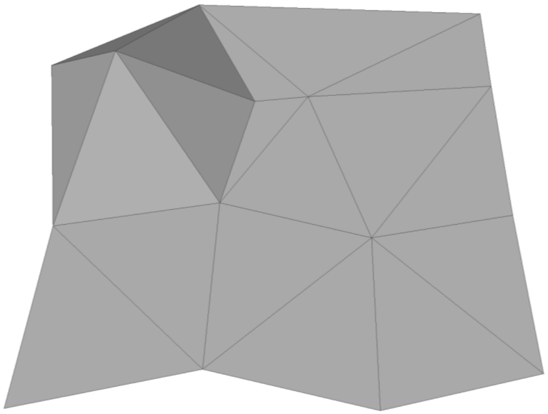
- $Z = \sum_{X \in \Omega} \exp -U(X)$

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 for meshes

- Graph nodes = vertices & graph edges = edges
- Graph nodes = facets & graph edges = edges



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

$$\Pr(X = x \ / \ Y = y) \propto \Pr(Y = y \ / \ X = x) \ . \ \Pr(X = x)$$

$\Pr(X = x \ / \ Y = y)$	$\Pr(Y = y \ / \ X = x)$	$\Pr(X = x)$
$\downarrow$	$\downarrow$	$\downarrow$
Posterior probability	**Likelihood**	**Prior probability**

Standard assumptions

- conditional independence of the observation

$$P(Y=y|X=x) = \prod_{i \in V} P(y_i|x_i)$$

- X is an MRF

From probability to energy

- data term : local dependency hypothesis (l=x)
- regularization : soft constraints

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

Data term

= -log (likelihhood) when Bayesian

Regularisation term

= - log (pairwise interaction prior) when Bayesian

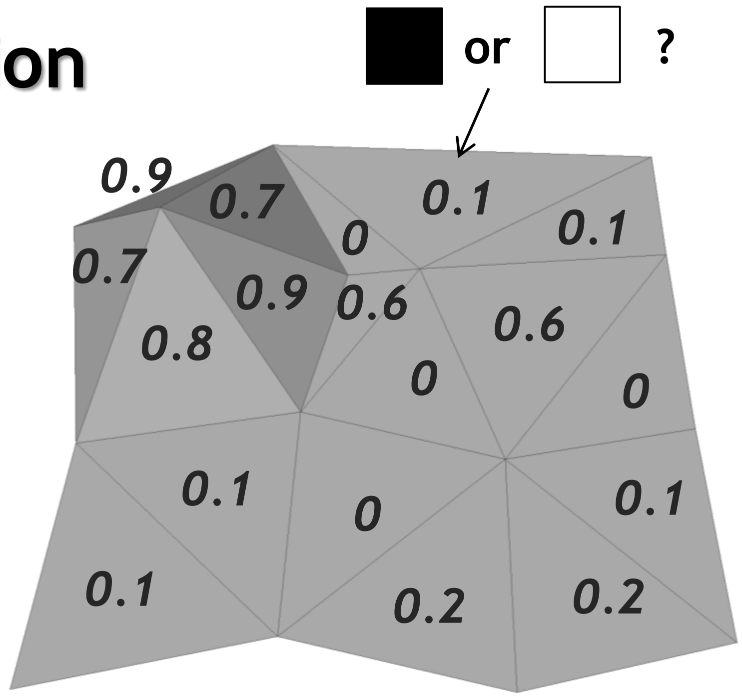
Optimal configuration

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)

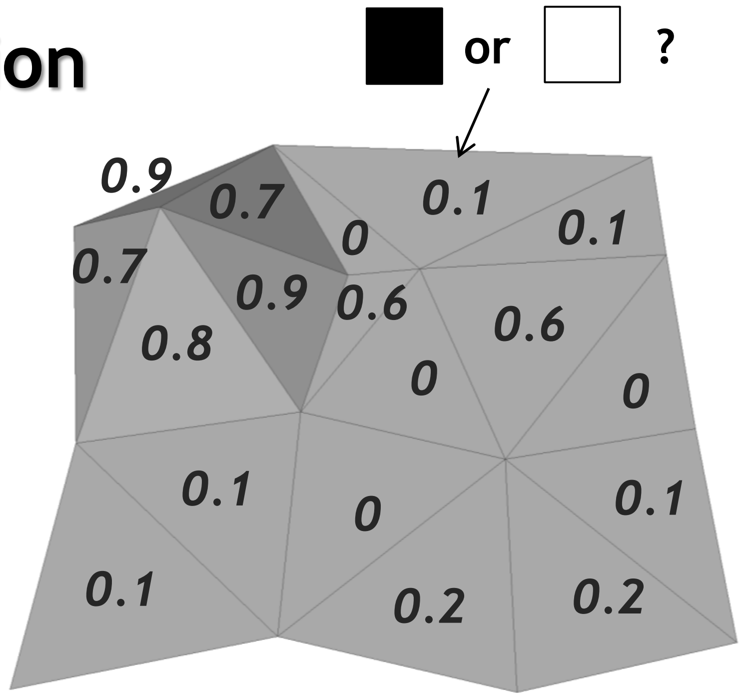
labels: {*white*, *black*} (l)

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

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

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

exercise: binary segmentation



Graph structure

Graph nodes = facets
Graph edges = common edges

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

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

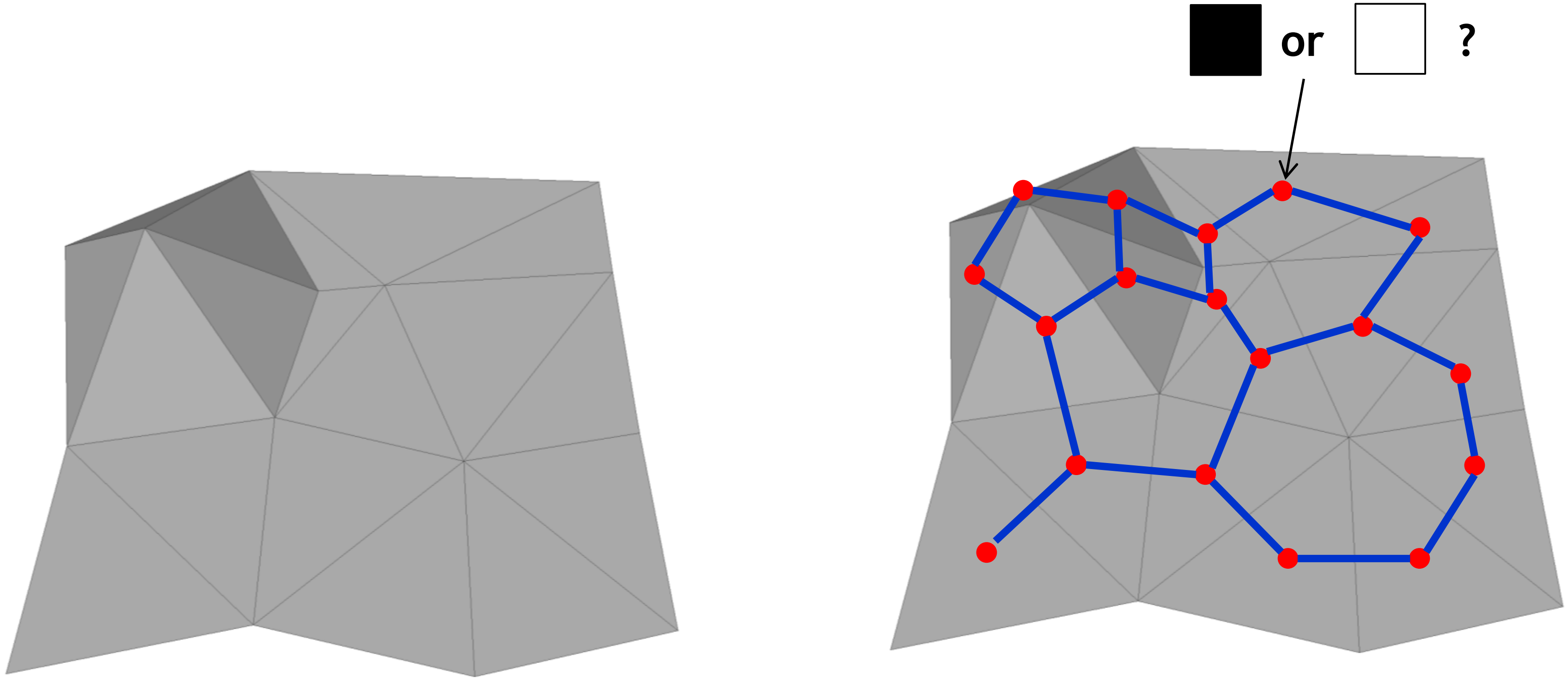
Energy: $U(l) = \sum_{i \in V} D_i(l_i) + \beta \sum_{\{i,j\} \in E} V_{ij}(l_i, l_j)$

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 $\beta = 0$? 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 restrictions on energy formulation

Monte Carlo sampling

slow but no restriction

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)

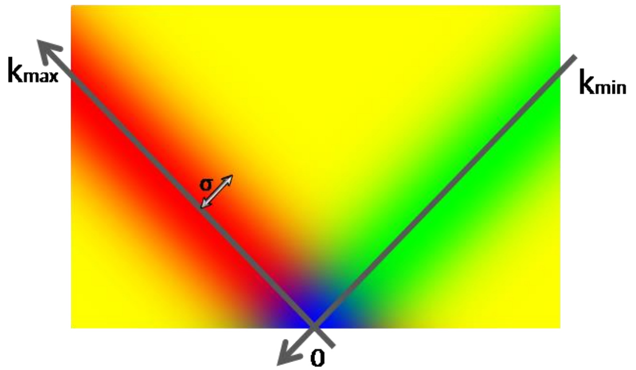
Example: mesh segmentation with principal curvature attributes & soft geometric constraints

Data term

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

with

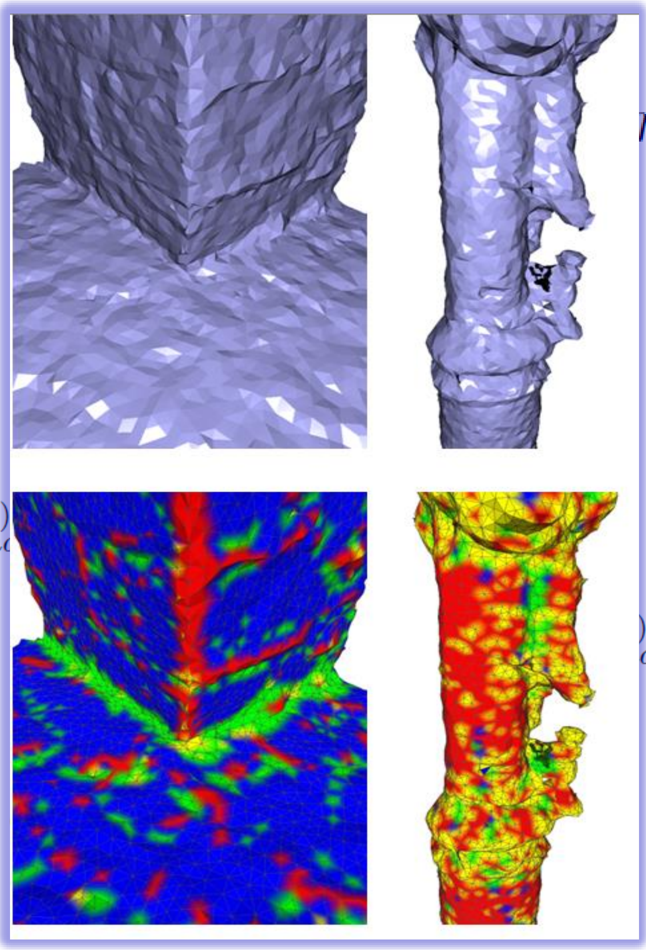
$$Pr(l_i | k_{min}^{(i)}, k_{max}^{(i)}) = \begin{cases} G_\sigma(k_{min}^{(i)}) G_\sigma(k_{max}^{(i)}) & \text{if } l_i = 1 \\ G_\sigma(k_{min}^{(i)})(1 - G_\sigma(k_{max}^{(i)})) & \text{if } l_i = 2 \\ (1 - G_\sigma(k_{min}^{(i)})) G_\sigma(k_{max}^{(i)}) & \text{if } l_i = 3 \\ (1 - G_\sigma(k_{min}^{(i)}))(1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



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

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

Data term



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

with

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

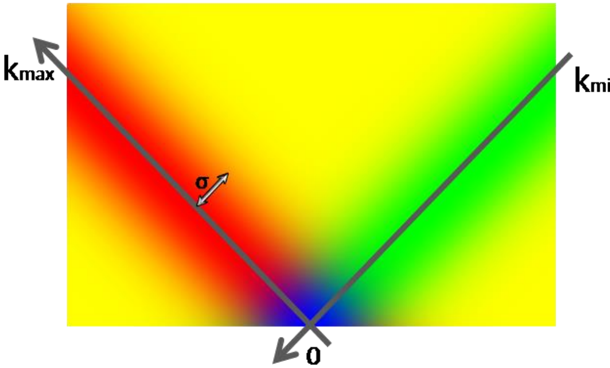
$_{ax}))$

if $l_i = 1$

if $l_i = 2$

if $l_i = 3$

if $l_i = 4$



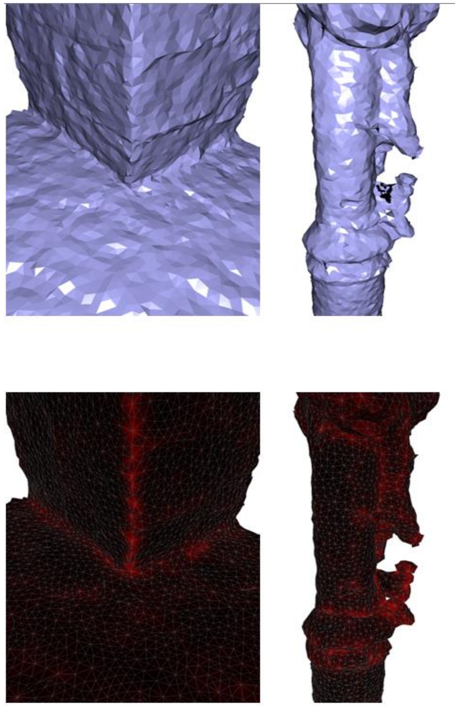
$G_\sigma(k) =$

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

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

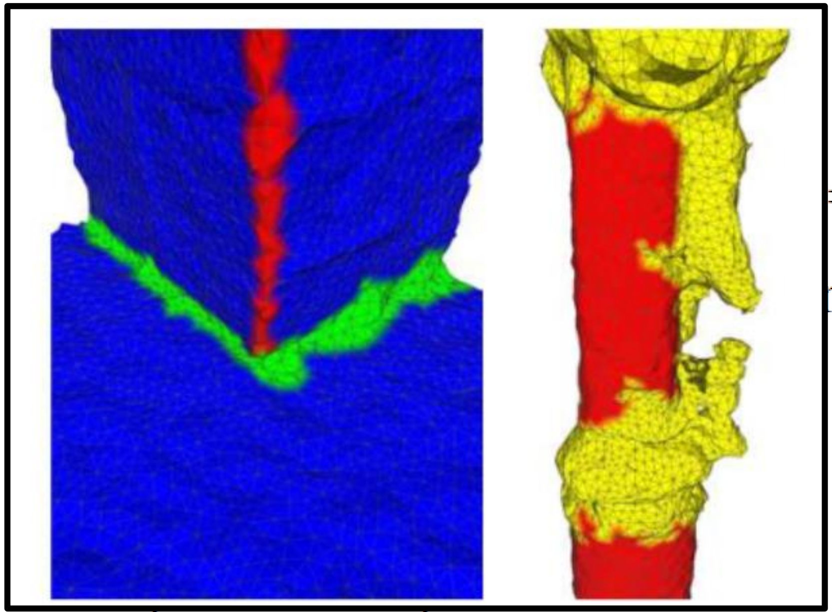
Example: mesh segmentation with principal curvature attributes & soft geometric constraints

Soft constraints

Label smoothness

Edge preservation

$V_{ij}(l_i, l_j)$

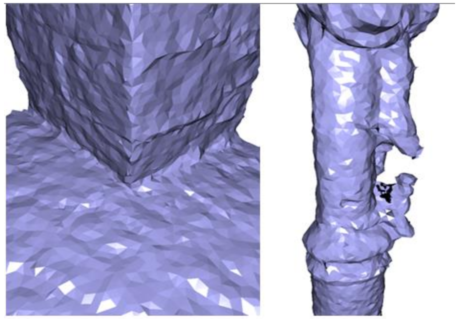


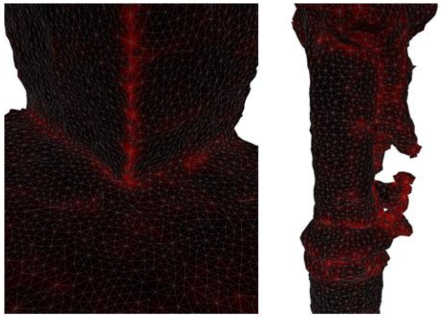
$\neq l_j$

rwise

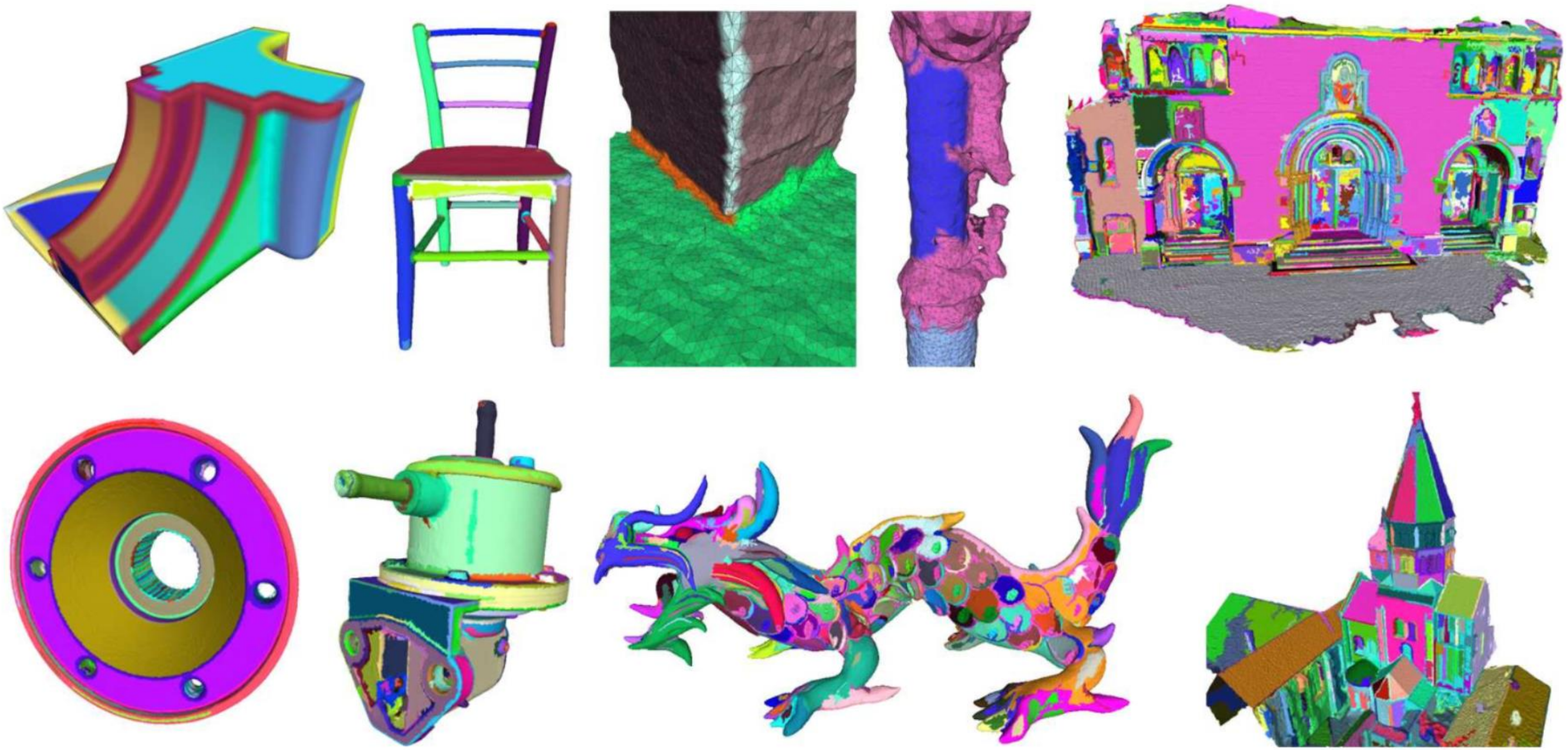
with

$k_{max} . \mathbf{w}_{\mathbf{max}}$





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

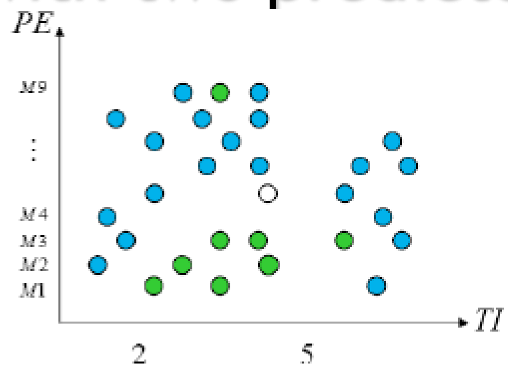


Classification by Machine learning (Random Forest)

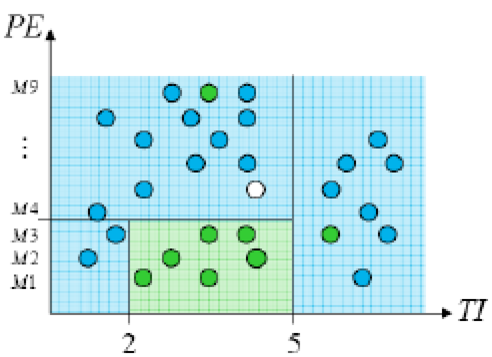
Decision tress involve greedy, recursive partitioning

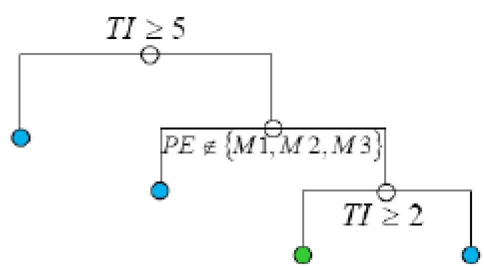
- Simple dataset with two predictors

TI	PE	Response
1.0	M2	good
2.0	M1	bad
...	...	...
4.5	M5	?

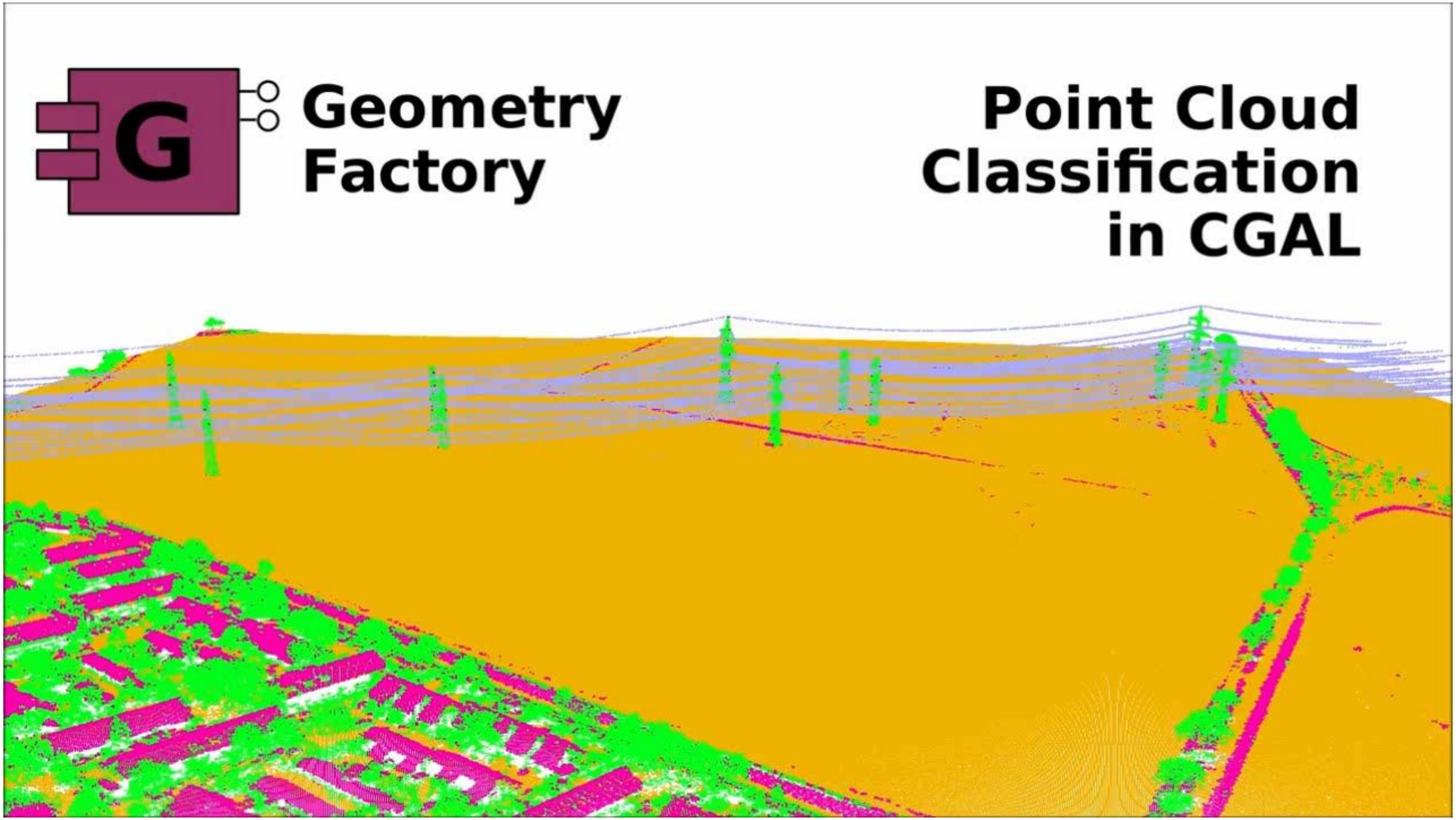


- Greedy, recursive partitioning along TI and PE



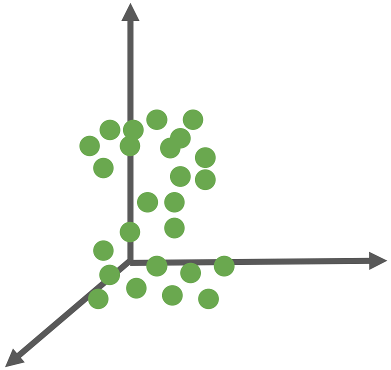


Example with cgal



Classification by Deep learning (PointNet)

End-to-end learning for irregular point data

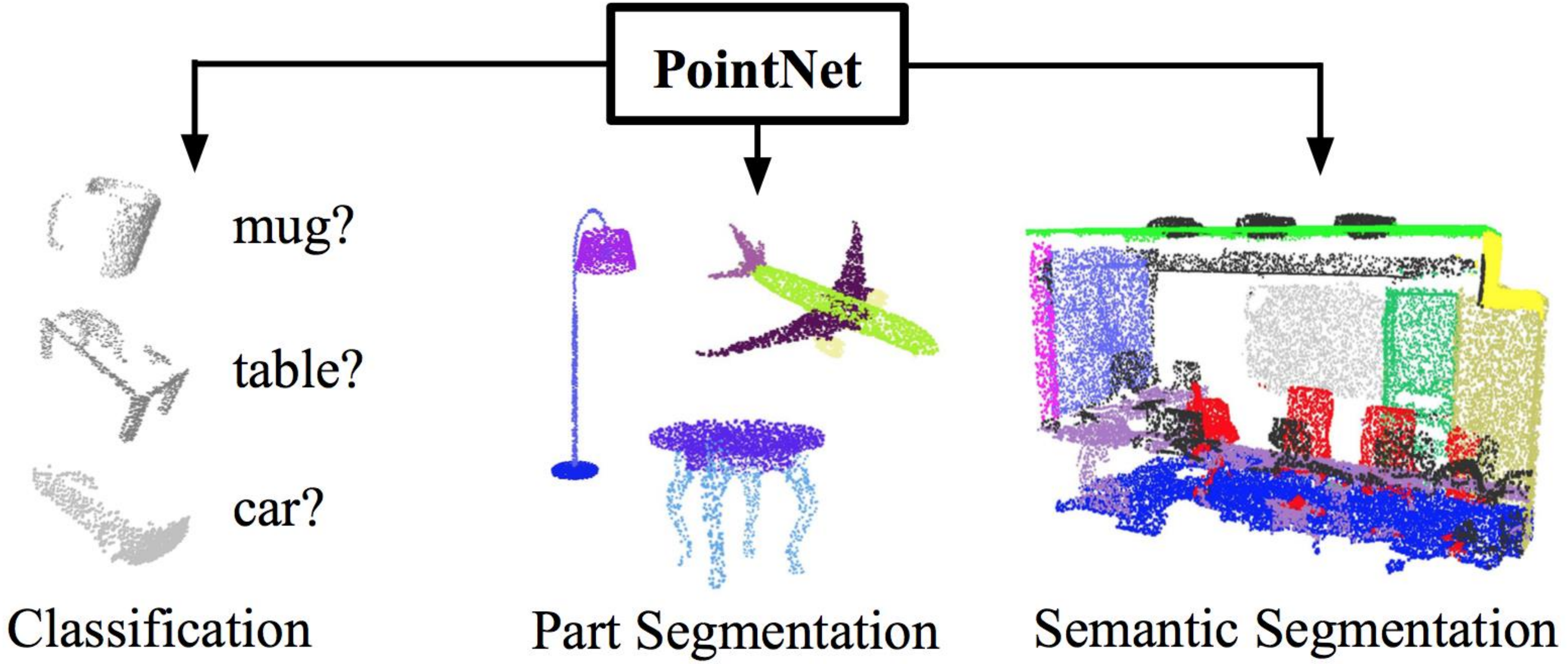


Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. (CVPR'17)

PointNet

End-to-end learning for irregular point data

Unified framework for various tasks



PointNet: challenges

The model has to respect key properties of point clouds:

Point Permutation Invariance

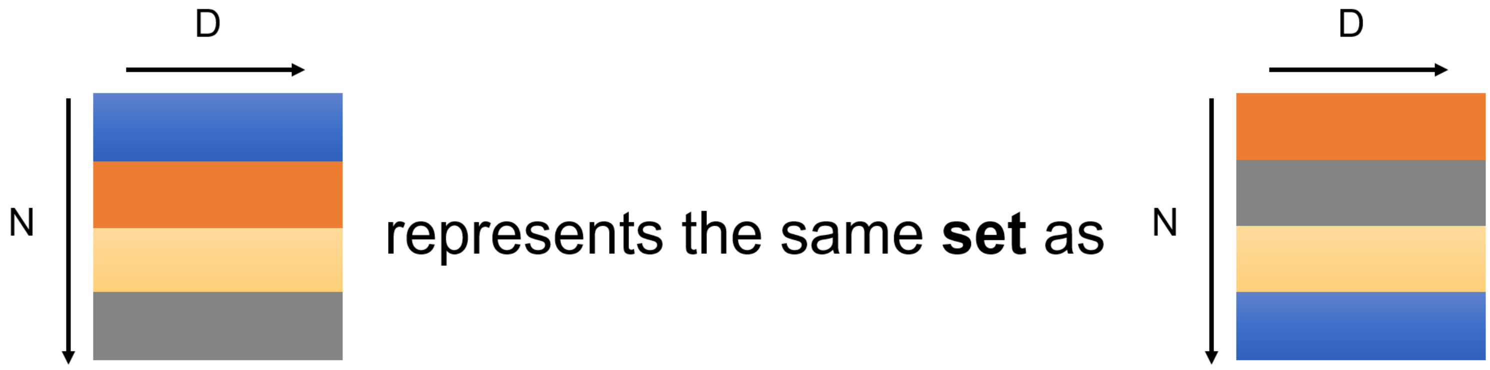
Point cloud is a set of unordered points

Spatial Transformation Invariance

Point cloud rigid motions should not alter classification results

First property: point permutation invariance

Point cloud: set of N **unordered** points, each represented by a D dim vector



Model needs to be invariant to N! permutations

First property: point permutation invariance

$$f(x_1, x_2, \ldots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

Examples:

$$f(x_1, x_2, \ldots, x_n) = \max\{x_1, x_2, \ldots, x_n\}$$

$$f(x_1, x_2, \ldots, x_n) = x_1 + x_2 + \ldots + x_n$$

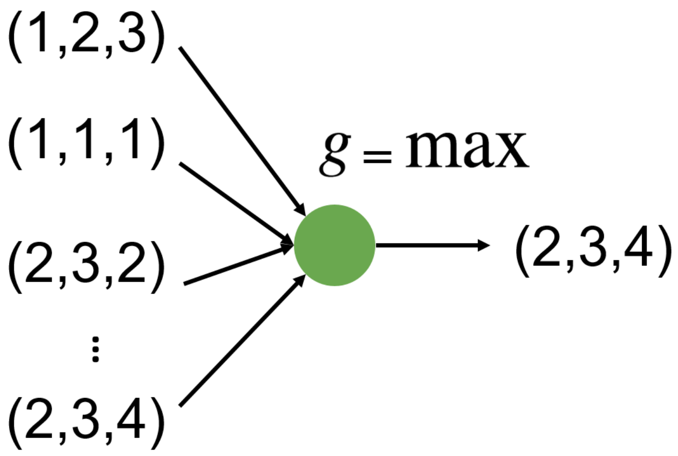
...

How can we construct a universal family of symmetric functions by neural networks?

First property: point permutation invariance

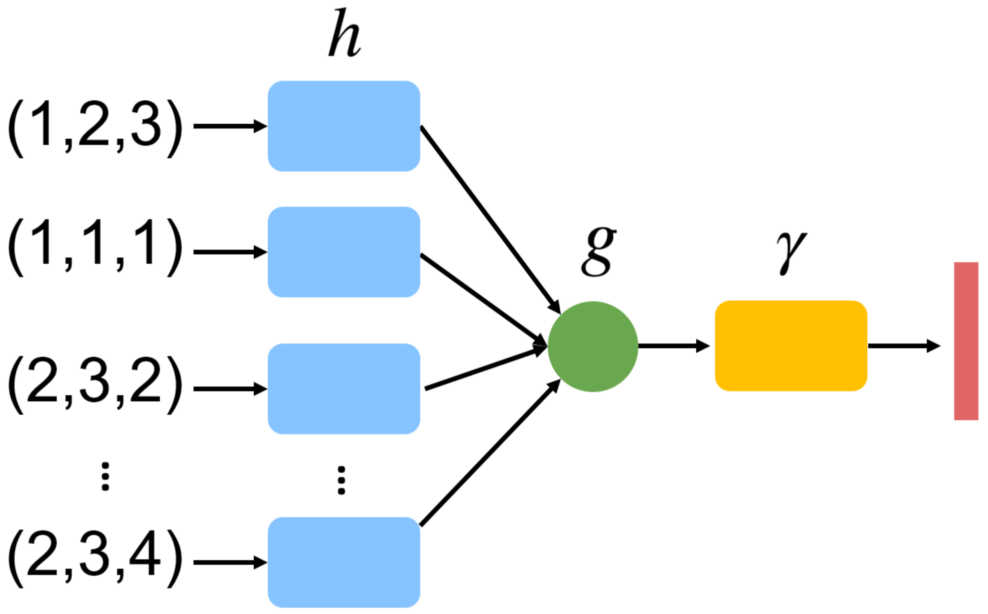
Simplest form: directly aggregate all points with a symmetric operator g

Just discovers simple extreme/aggregate properties of the geometry



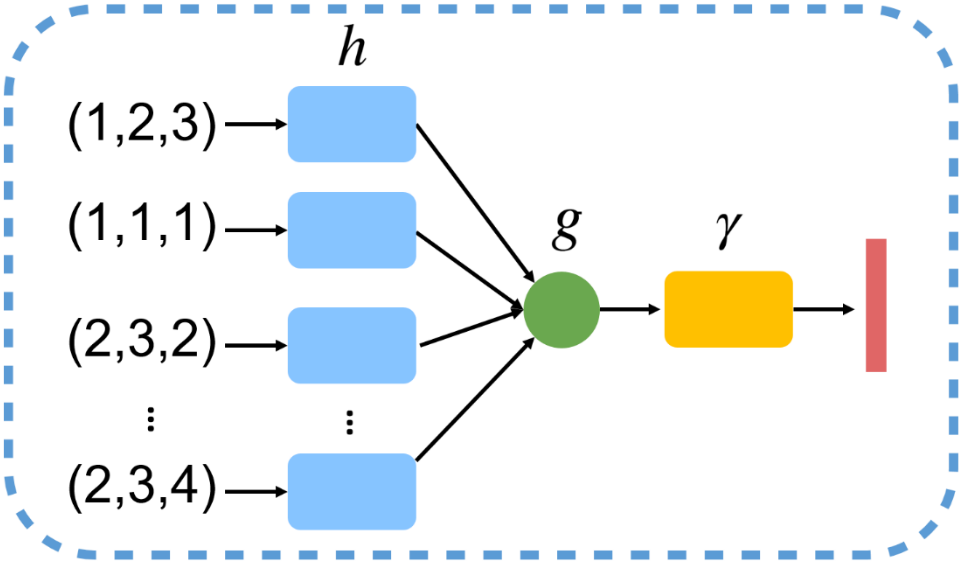
First property: point permutation invariance

Embed points to a high-dim space before aggregation.
Aggregation in the (redundant) high-dim space encodes more interesting properties of the geometry.



First property: point permutation invariance

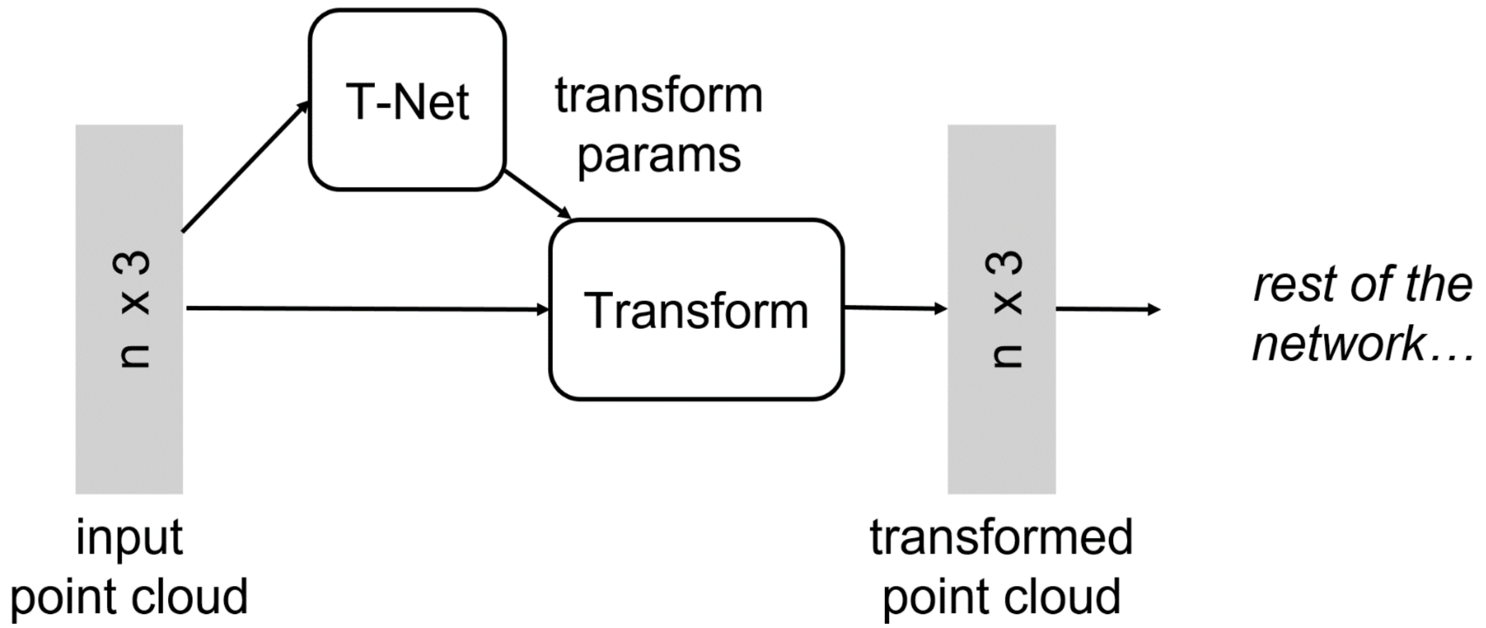
$$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$$ is symmetric if g is symmetric



PointNet (vanilla)

Second property: spatial transformation invariance

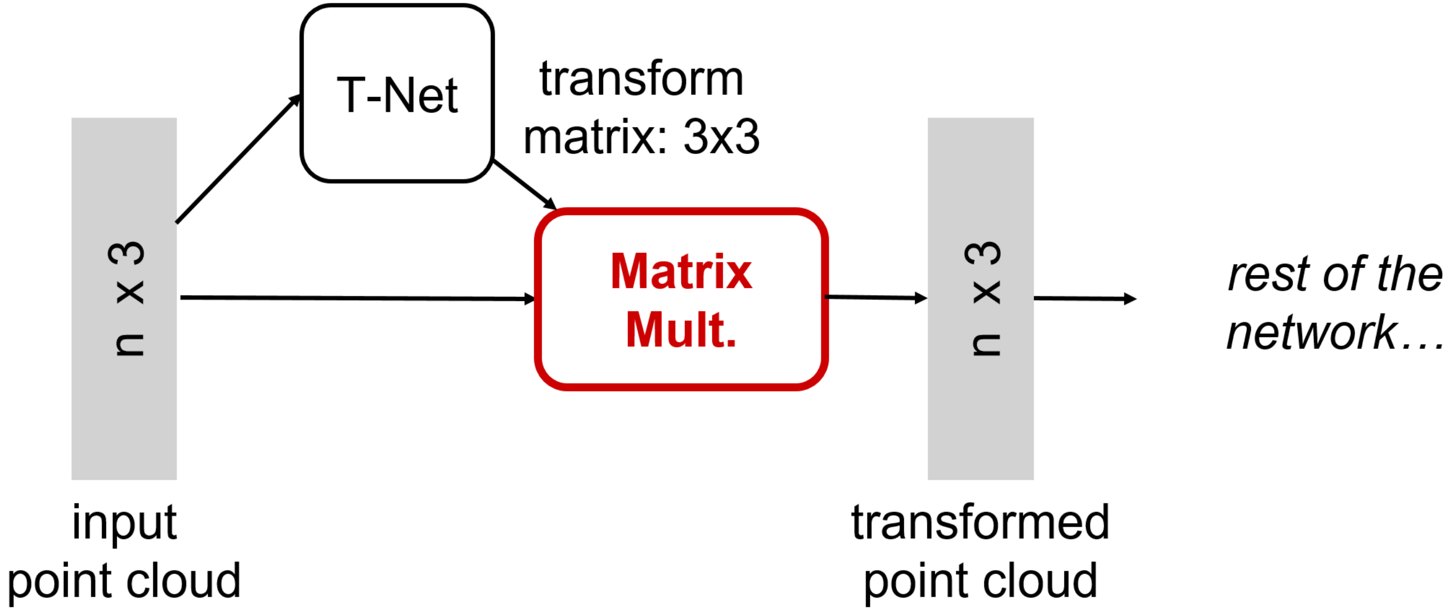
Idea: Data dependent transformation for automatic alignment



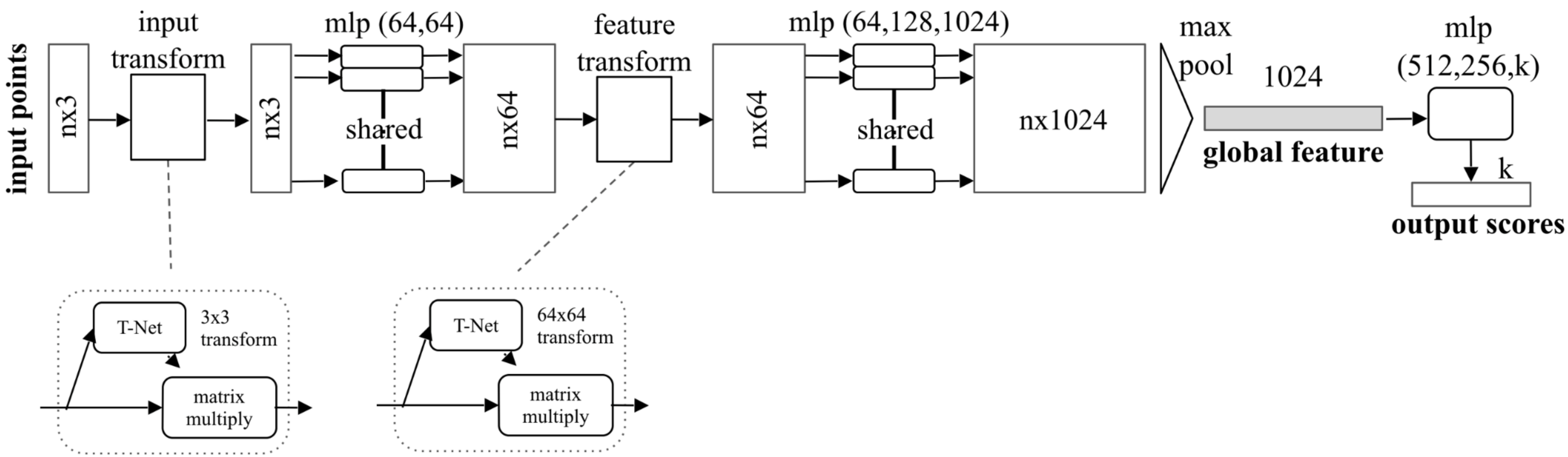
Second property: spatial transformation invariance

Idea: Data dependent transformation for automatic alignment

The transformation is just matrix multiplication!



PointNet architecture for classification tasks



Results on indoor scene classification

Other deep learning architectures for point cloud classification

- 3D CNN
- PointNet++
- DG-CNN
- PointSIFT
- …