Shape Similarity based on a Treelet Kernel with Edition

joint work

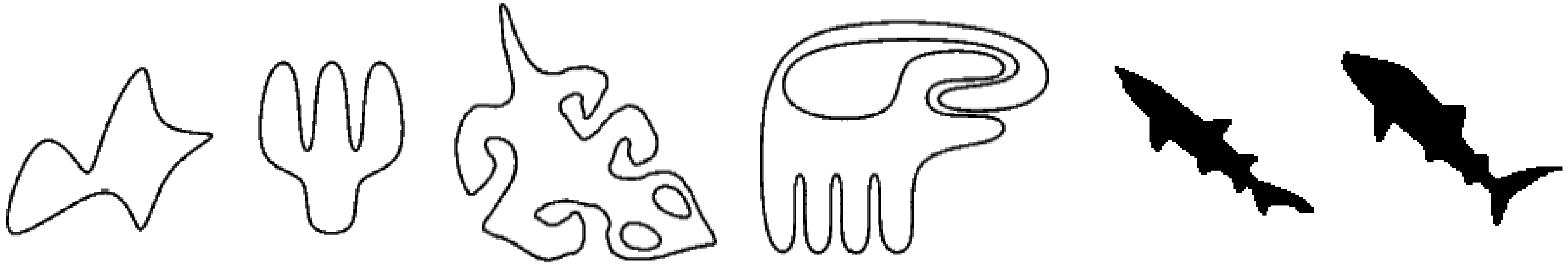
S. Bougleux, L. Brun, M. Mokhtari F.-X. Dupé

UNICAEN université de Caen Basse-Normandie
ENSICAEN ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN & CENTRE DE RECHERCHE
CNRS
Aix*Marseille université
LIF
GREYC

Introduction

Context

- 2D shapes given by
  - continuous functions of their boundaries
  - binary functions defined over a discrete domain



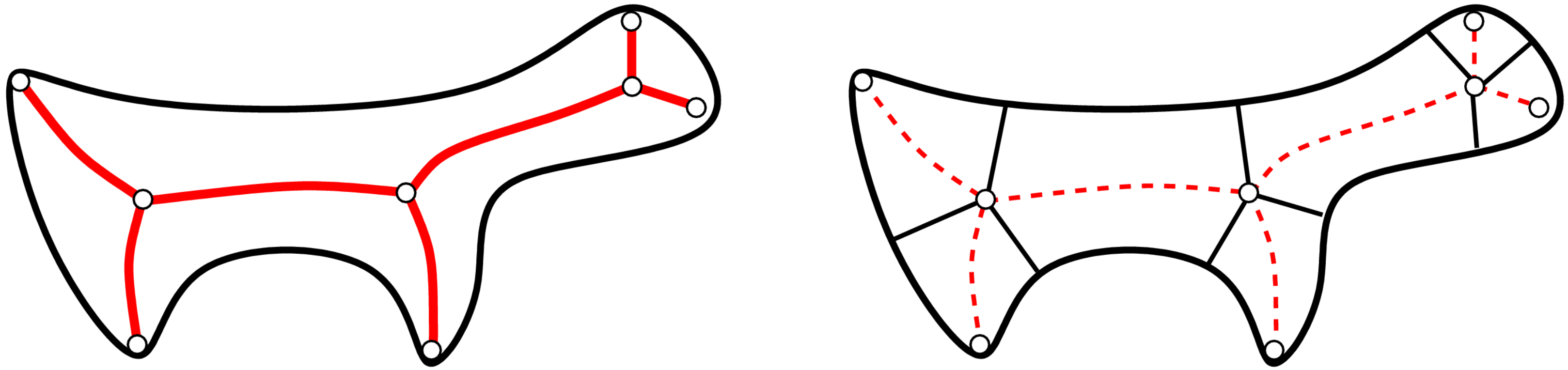
How to compare these shapes ?

- for matching, classification
  - structural and numerical methods
  - boundary-based, skeleton-based

Introduction

From shapes to graphs

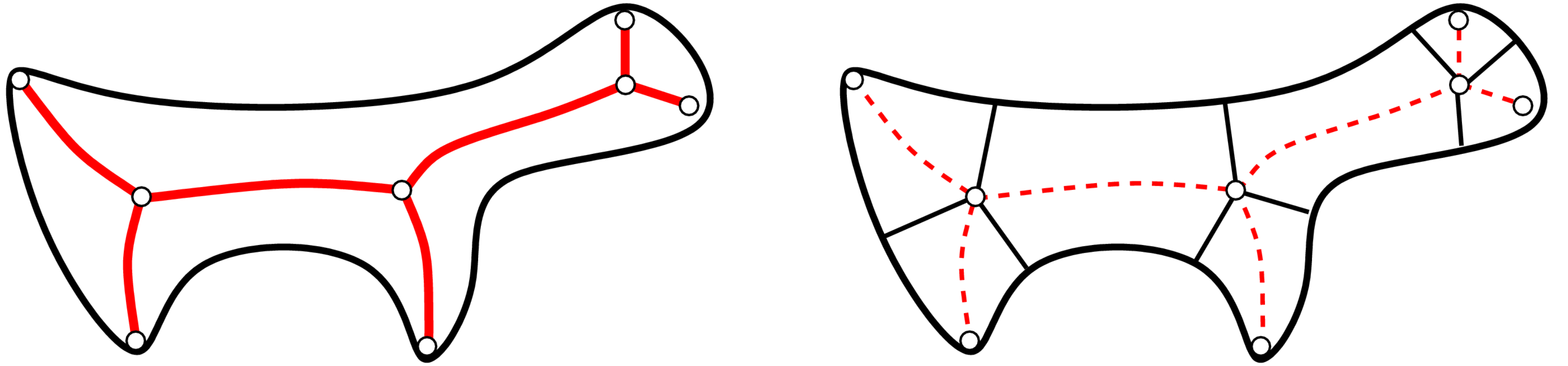
- **medial axis – skeleton** (centers of maximal inscribed disks)
  - graph structure invariant to scaling, rotations and translations
  - homotopic to the shape
  - induces a natural decomposition of the shape
  - weak representation of local properties needed for shape comparison



Introduction

From shapes to graphs

- **skeleton graph + attributes**
- **nodes** (intersection and terminal points)
  - distance to gravity center (normalized by shape area)
- **edges** (branches)
  - length of the boundary induced by the branch (normalized by total length)
  - evolution of the radius of minimal inscribed disks along the branch
  - area of the corresponding sub-shape (normalized by the total area)



Introduction

From shapes to graphs

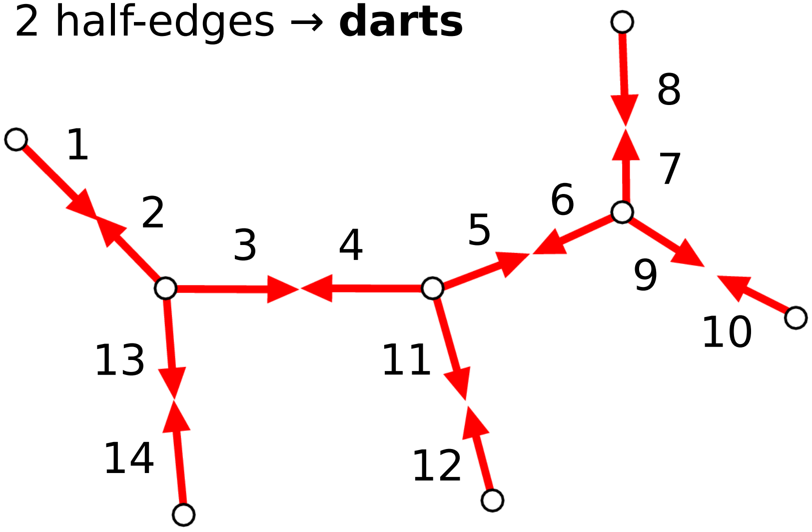
- skeleton graph **embedded in the plane**
  - **combinatorial map** to encode the orientation of edges around nodes

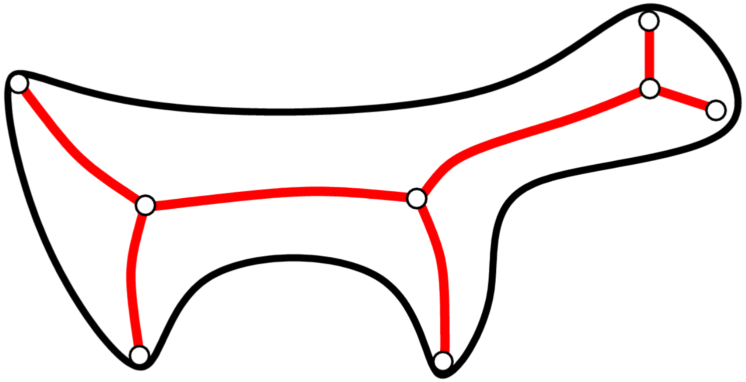
$$M=(D,\sigma,\alpha)$$

involution α encodes edges

permutation σ encodes the order of the darts around each vertex

each edge is decomposed
into 2 half-edges → **darts**





$$D=\{i\}_{i=1,\ldots,14}$$

$$\sigma=(1)(2\ 3\ 13)(4\ 5\ 11)(6\ 7\ 9)(8)(10)(12)(14)$$

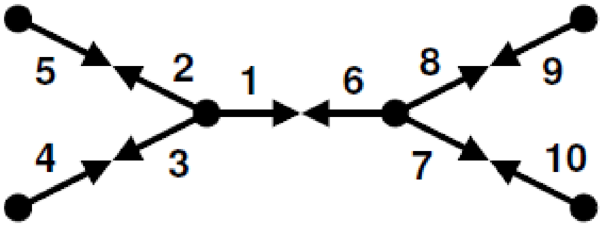
$$\alpha=(1\ 2)(3\ 4)(5\ 6)(7\ 8)(9\ 10)(11\ 12)(13\ 14)$$

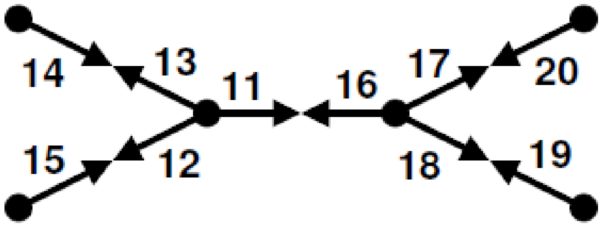
Introduction

Comparison of shapes → combinatorial maps

- equivalence between maps (having the same number of darts)
- **orientation-preserving symmetry** [Cori 85]
  - ψ isomorphism between two maps $M_1=(D_1,\sigma_1,\alpha_1)$ and $M_2=(D_2,\sigma_2,\alpha_2)$
  - edges are preserved: $\psi\circ\alpha_1=\alpha_2\circ\psi$
  - orientation around nodes is preserved: $\psi\circ\sigma_1=\sigma_2\circ\psi$
  - set of automorphisms of M = $Aut(M)$

$$M_1=(D_1,\sigma_1,\alpha_1)\xrightarrow[\psi:D_1\to D_2]{}M_2=(D_2,\sigma_2,\alpha_2)$$





isomorphisms

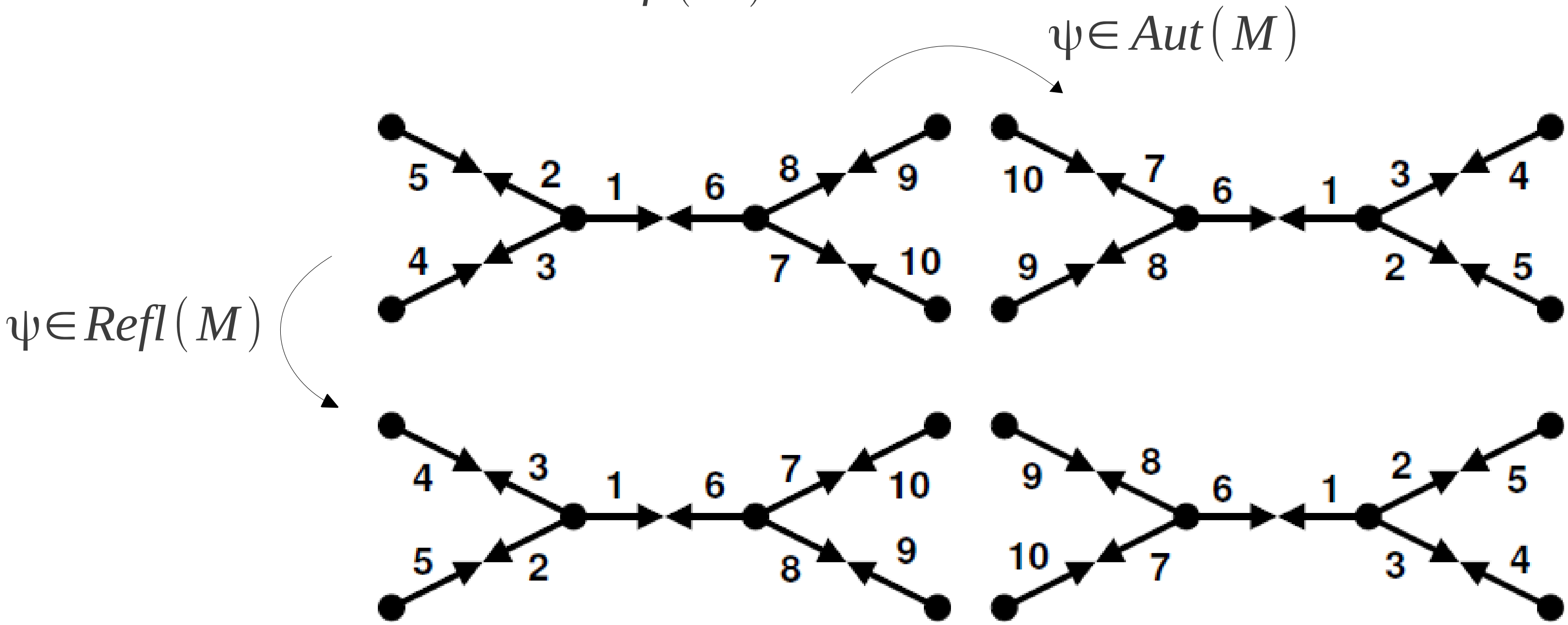
$$\psi_1=\begin{pmatrix}1&2&3&4&5&6&7&8&9&10\\11&13&12&15&14&16&18&17&20&19\end{pmatrix}.$$

$$\psi_2=\begin{pmatrix}1&2&3&4&5&6&7&8&9&10\\16&18&17&20&19&11&13&12&15&14\end{pmatrix}.$$

Introduction

Comparison of shapes → combinatorial maps

- equivalence between maps (having the same number of darts)
- **orientation-reversing symmetry**
  - ψ reflection between two maps $M_1=(D_1,\sigma_1,\alpha_1)$ and $M_2=(D_2,\sigma_2,\alpha_2)$
  - edges are preserved: $\psi\circ\alpha_1=\alpha_2\circ\psi$
  - orientation around nodes is reversed: $\psi\circ\sigma_1=\sigma_2^{-1}\circ\psi$
  - set of reflections of M = $Refl(M)$



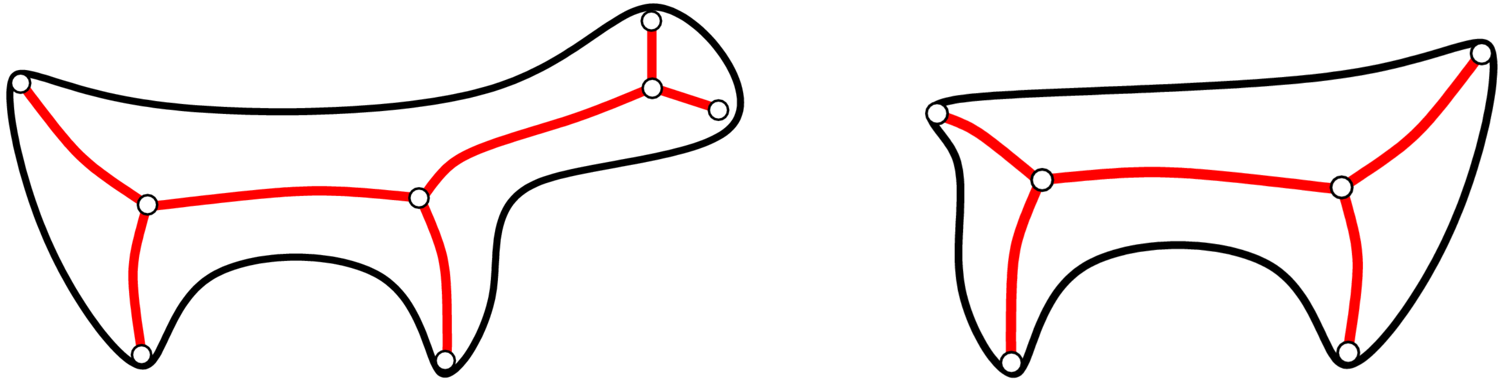
Introduction

Comparison of shapes → combinatorial maps

- equivalence between maps (having the same number of darts)

$$Sym(M) = Aut(M) \cup Refl(M) = Aut(M) \cup Aut(M^{-1})$$

- equivalence between maps with different numbers of darts ?
- comparison of maps with attributes attachted to nodes and edges ?



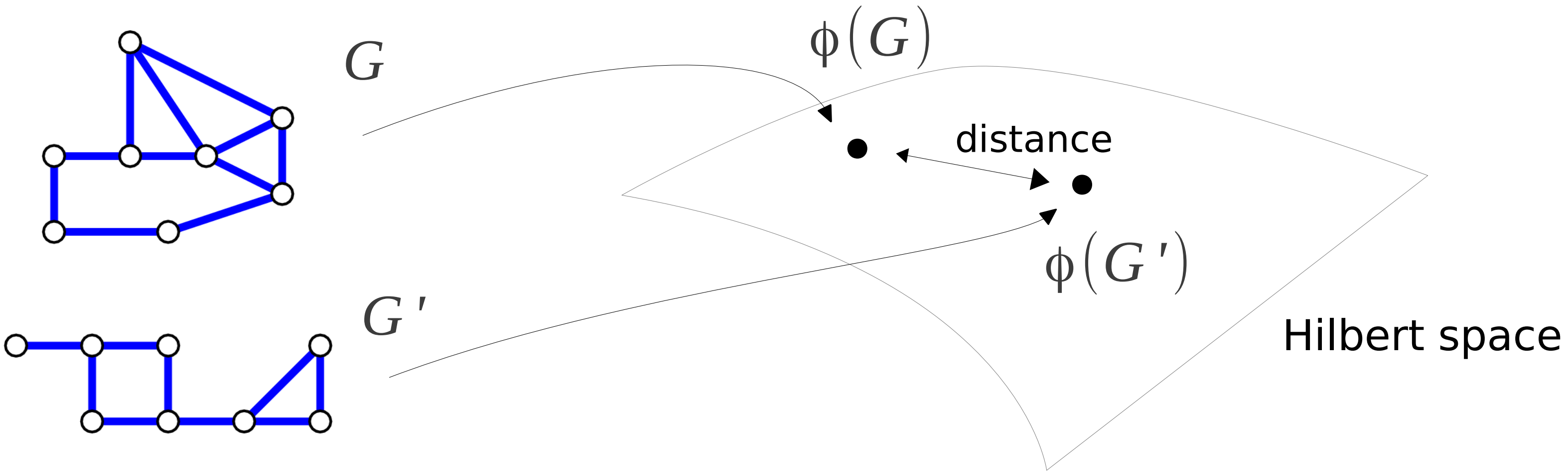
Introduction

Graph kernels

- *idea*: induce a **mapping within a Hilbert space**

$$K(G,G')=\langle\phi(G),\phi(G')\rangle$$

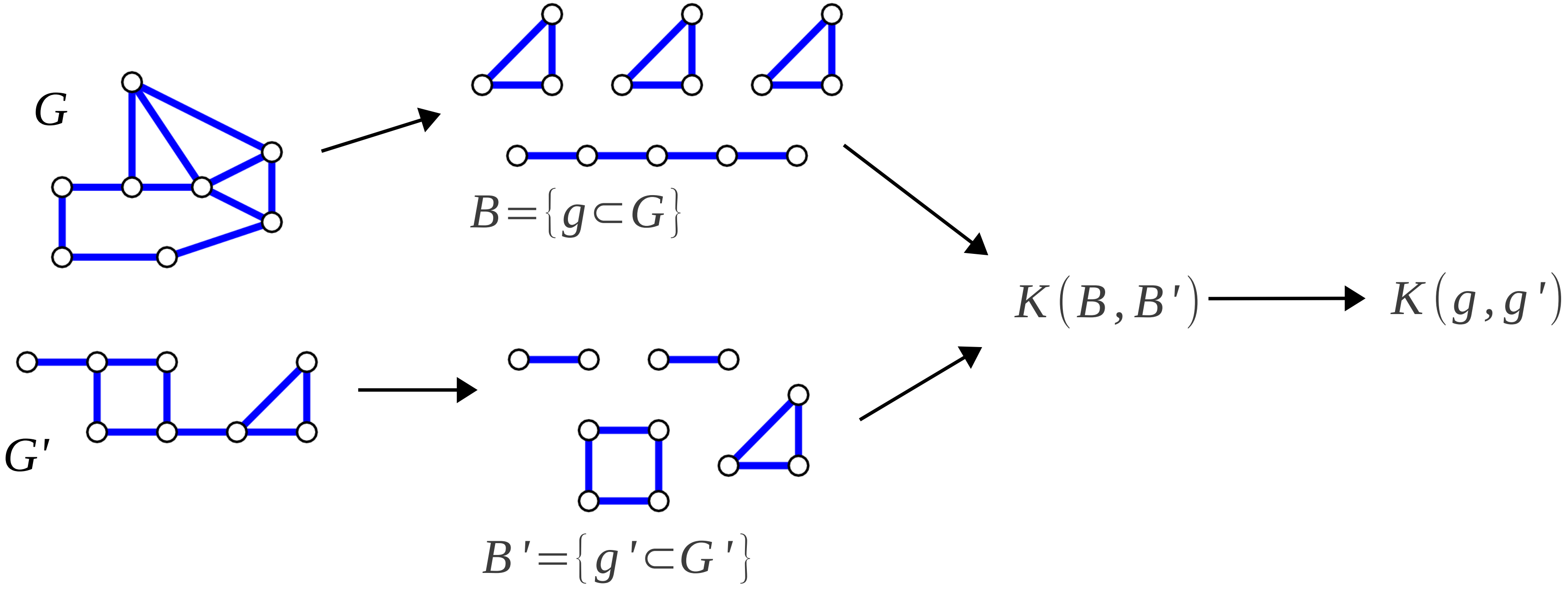
  - ϕ projects graph into a particular vector (feature) space
  - K may be understood as a similarity measure between G and G'
  - K is usually designed to provide an easy separation between classes
  - equility hold only if K is (symmetric) **positive-definite**



Introduction

Design of graph kernels

set of graphs → bags of patterns (subgraphs) → kernel between bags → kernel between patterns



Introduction

Bag of patterns framework

- all walks of a graph (Random walk kernel) [Kashima *et al.*, ICML03]
  - \- implicit enumeration
  - \- linear representation
- all trails up to a given depth [Dupe and Brun, GbR09]
  - \+ explicit enumeration
  - \- linear representation
- all tree patterns up to a given length [Mahé and Vert, Machine Learning 09]
  - \- implicit enumeration
  - \+ nonlinear representation
- all subgraphs up to size 5 [Shervashidze *et al.*, Conf. Art. Intel. and Stat. 09]
  - \+ explicit enumeration
  - \+ nonlinear representation
- all tree patterns of a dictionary [Gauzere *et al.*, GbR11] [Bougleux *et al.*, ICPR12]
  - \+ explicit enumeration
  - \+ nonlinear representation

Shape Similarity based on a Treelet Kernel with Edition

1) Introduction

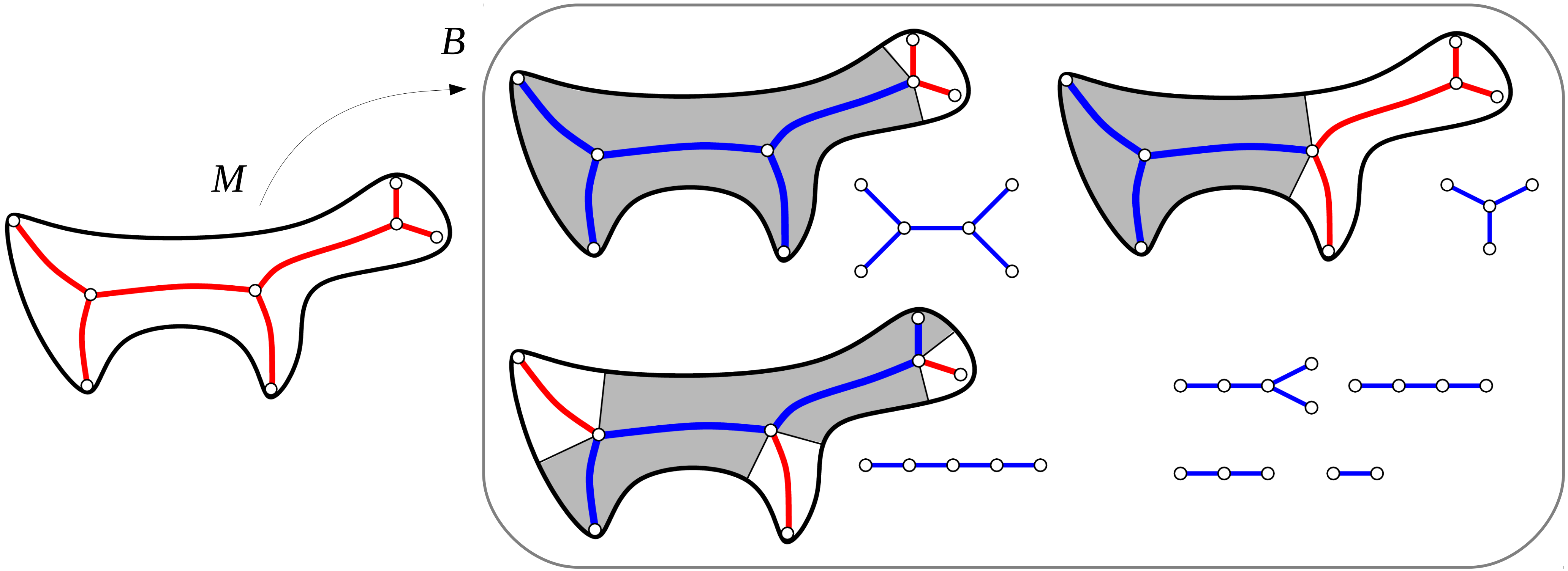
2) Shape similarity

3) Experiments

Shape similarity

Combinatorial map encoding → Bag of sub-structures

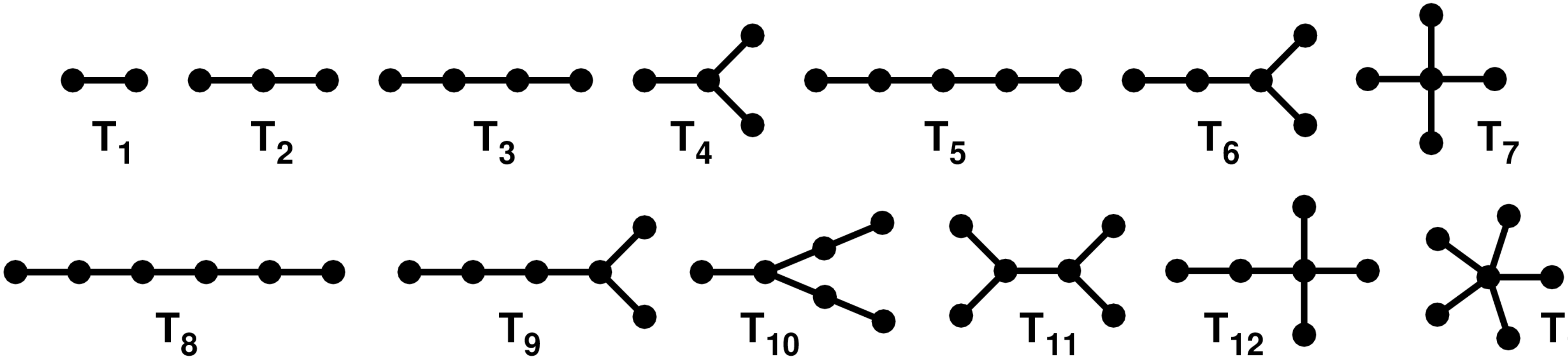
- shape represented by a bag B of sub-shapes easier to compare
- each sub-shape is encoded by a submap of the skeleton map M



Shape similarity

Combinatorial map encoding → Bag of treelets

- a treelet is an instance of a tree pattern in a map
- bag of treelets
  - enumeration of all the treelets of a map
  - according to a dictionary of tree patterns



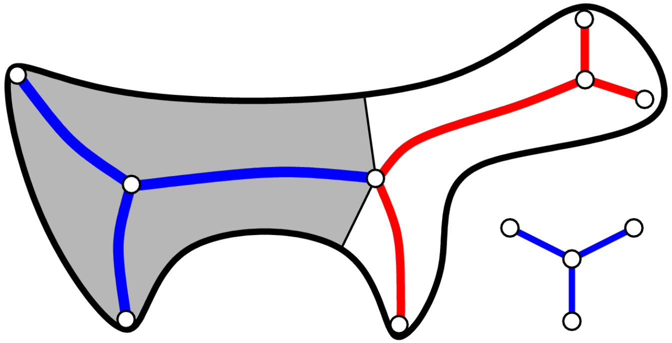
Shape similarity

Kernel between two bags of treelets

$$K_T(M,M')=\frac{1}{|B||B'|}\sum_{t\in B}\sum_{t'\in B'}\underbrace{\lambda_B(t)}\underbrace{\lambda_{B'}(t')}\underbrace{K(t,t')}$$

minor kernel bewteen two treelets

relevance of a treelet according to its bag



$$\lambda_B(t)=\frac{w(t)}{\max\limits_{t'\in B} w(t')}$$

- with $w(t)$ the imortance of a treelet according to the map

$$w(t)=\sum_{e\in t} w(e)$$

normalized induced boudary length

- reduce the influence of treelets encoding non-relevant parts of the shape

Shape similarity

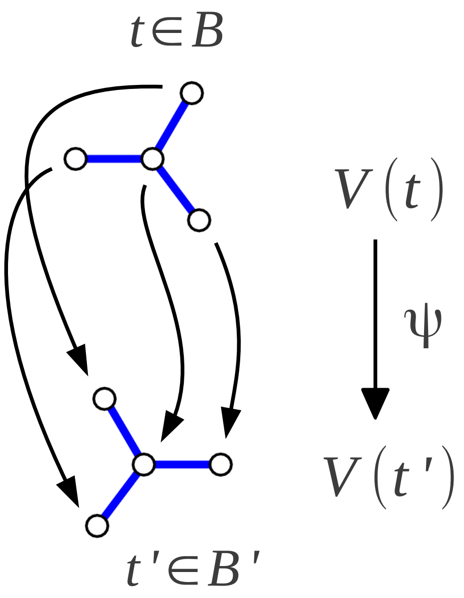
Treelets corresponding to a same tree pattern T_p

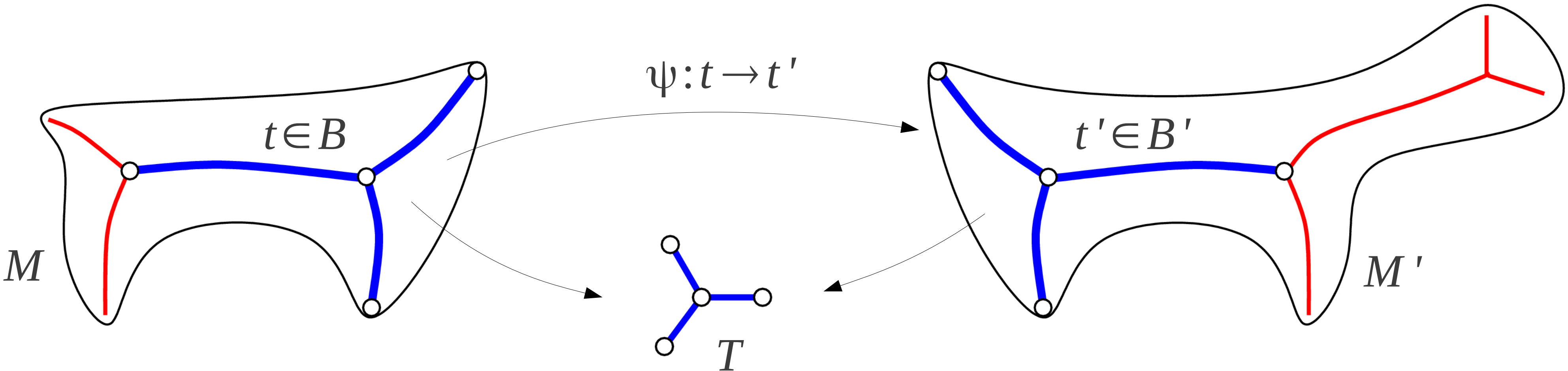
- given a mapping $\psi : t \rightarrow t'$
- **kernel between treelets aligned by ψ**
  = product of node and edge similarities

$$K_\psi(t,t') = \prod_{v\in V(t)} K_V(v,\psi(v)) \prod_{e\in E(t)} K_E(e,\psi(e))$$

- **kernel between nodes or edges**

$$K_A(a,a') = \prod_{k=1}^{k=n_A} \exp\left(-\frac{\|f_{A,k}(a)-f_{A,k}(a')\|^2}{2\sigma^2}\right)$$

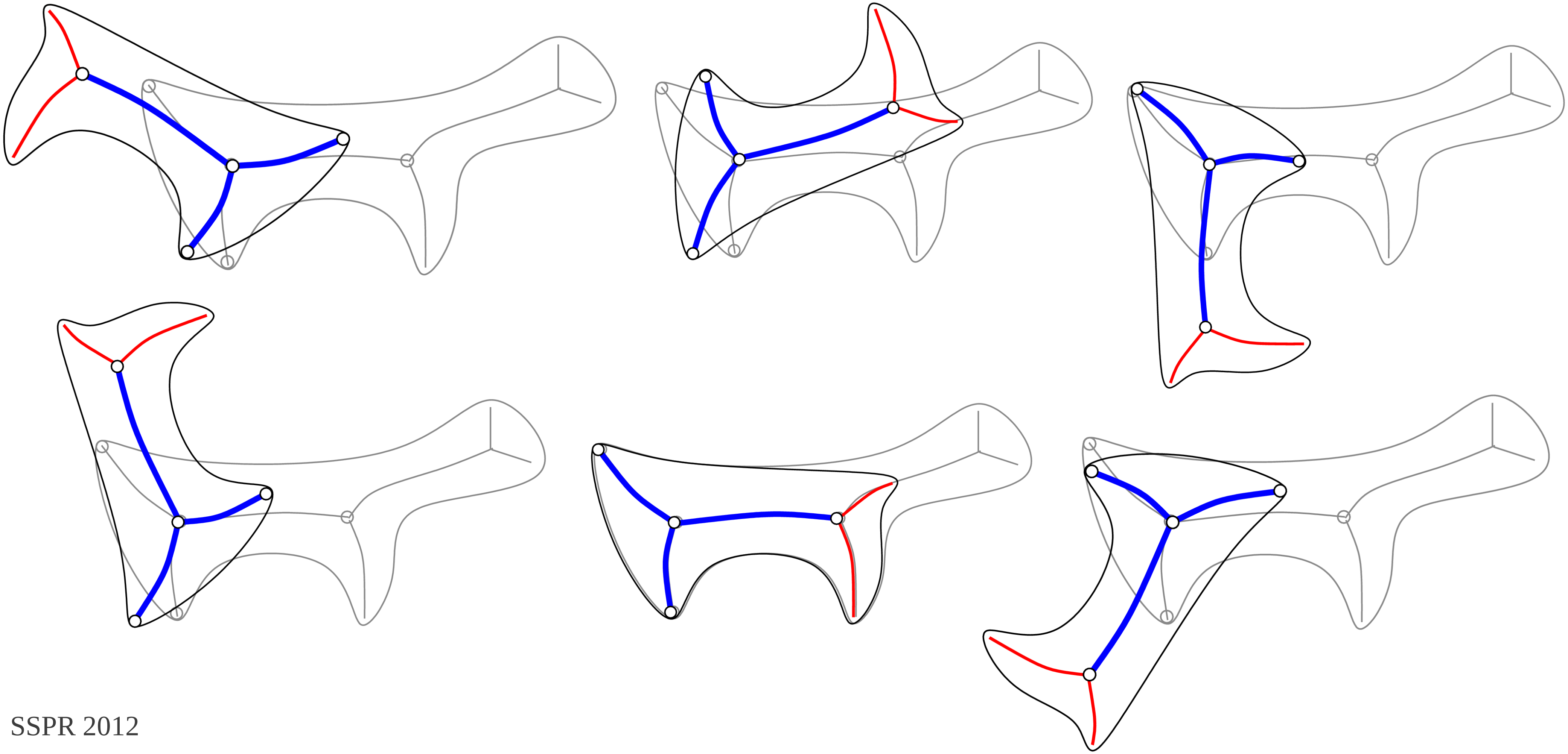




Shape similarity

Treelets corresponding to a same tree pattern T_p

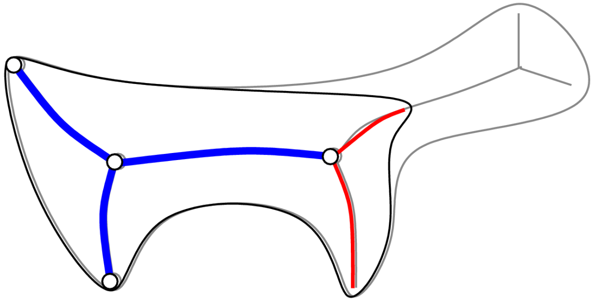
- several mappings $\psi \in Sym(T_p)$
- correspond to rotational and mirror shape symmetries



Shape similarity

Treelets corresponding to a same tree pattern T_p

- among the mappings $\psi \in Sym(t,t')$
- choosing the one realizing the **best alignment**: $\underset{\psi}{argmax}\ K_{\psi}(t,t')$
  - ➔ may lead to kernels not (symmetric) positive-definite
  - ➔ inadapted to discriminate dissimilar treelets



- **kernel between two treelets**
  = average of the similarities between their different matches

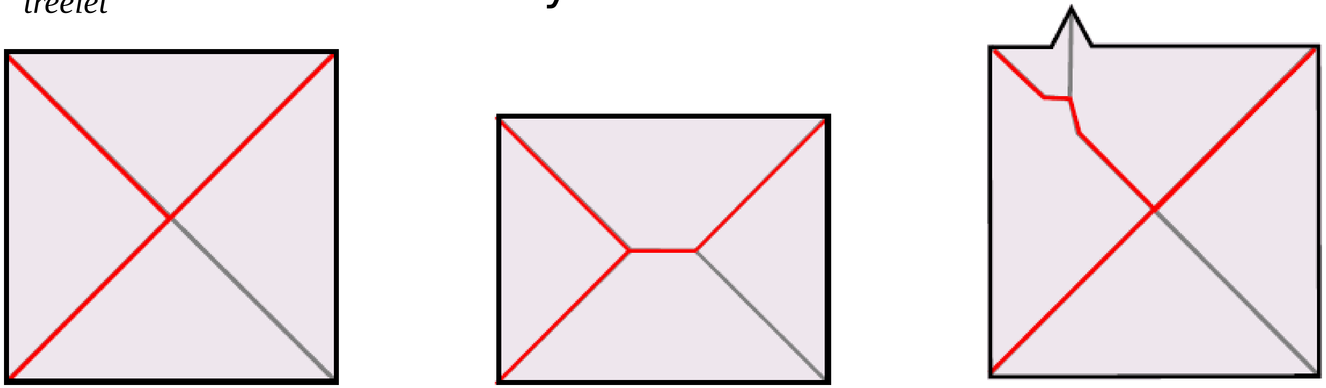
$$K_{treelet}(t,t') = \frac{1}{|Sym(t,t')|} \sum_{\psi \in Sym(t,t')} K_{\psi}(t,t'), \quad if\ Sym(t,t') \neq \emptyset,$$

and 0 otherwise (t and t' are not isomorphic to a same tree pattern)

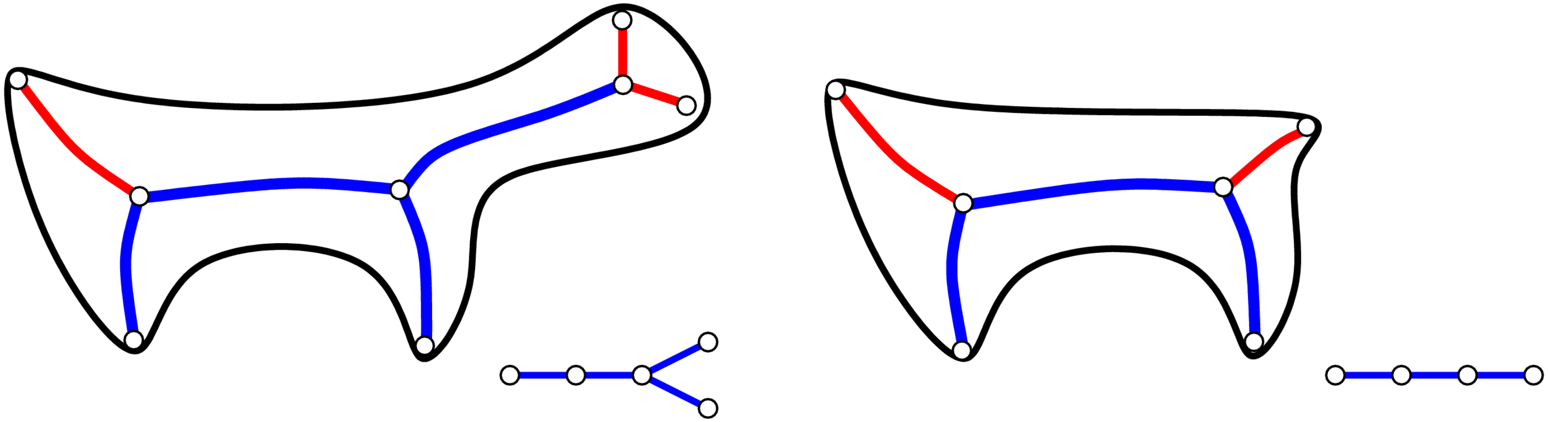
Shape similarity

Edition process

- skeletons are **sensitive to small shape deformations**
  - kernel $K_{treelet}$ can be affected by structural noise



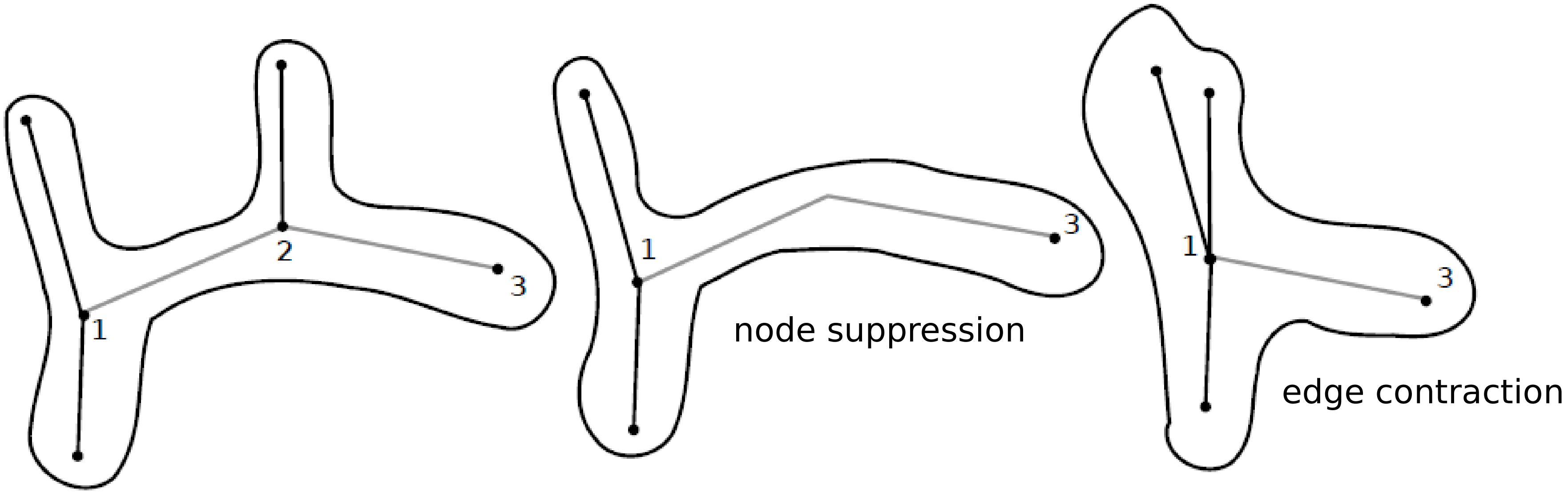
- 2 **treelets not corresponding to a same tree pattern**
  - may be similar up to node suppressions and edge contractions



Shape similarity

Edition process

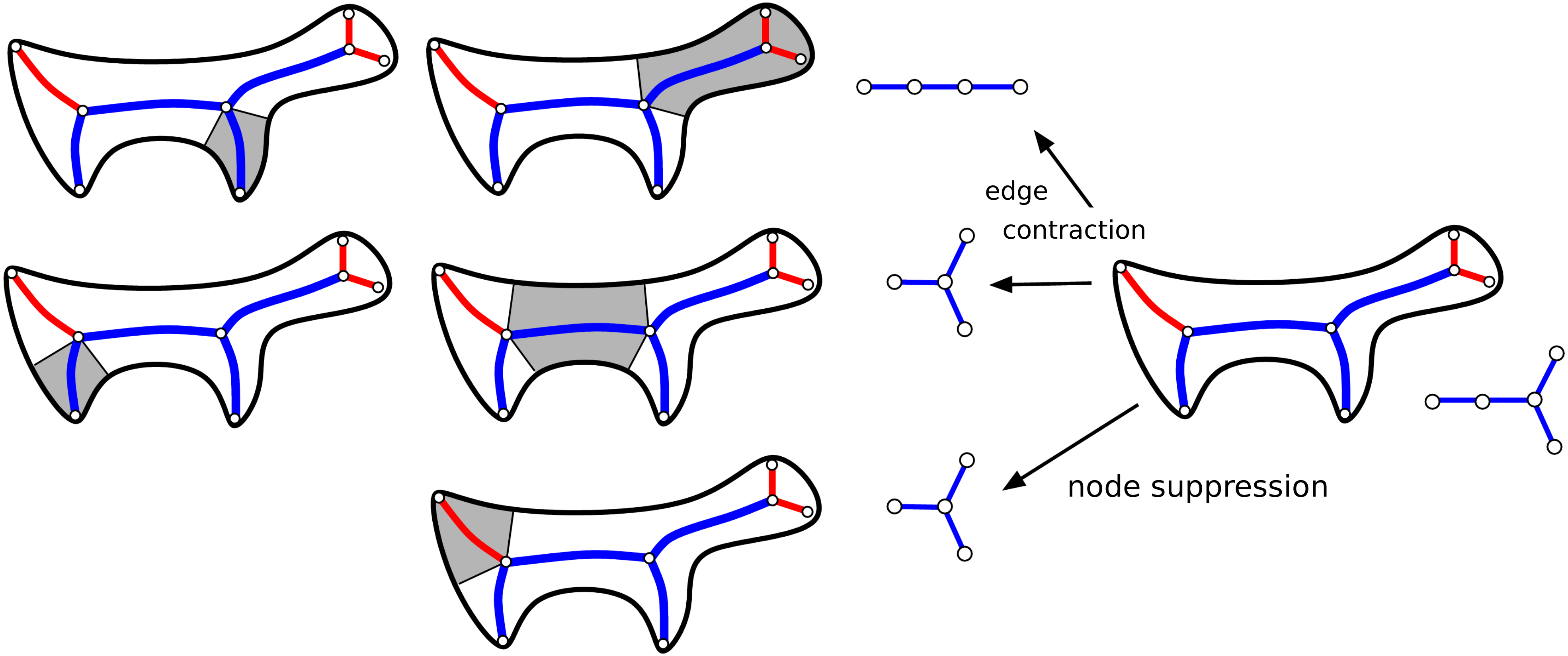
- each treelet is transformed into a sequence of smaller ones
- **node suppression**
  - cut parts connected to the node and outside of the treelet
  - merge parts connected to the node and inside the treelet
  - restricted to nodes of degree 2
- **edge contraction**
  - contraction of the shape
  - applied to each edge of the treelet



Shape similarity

Edition process

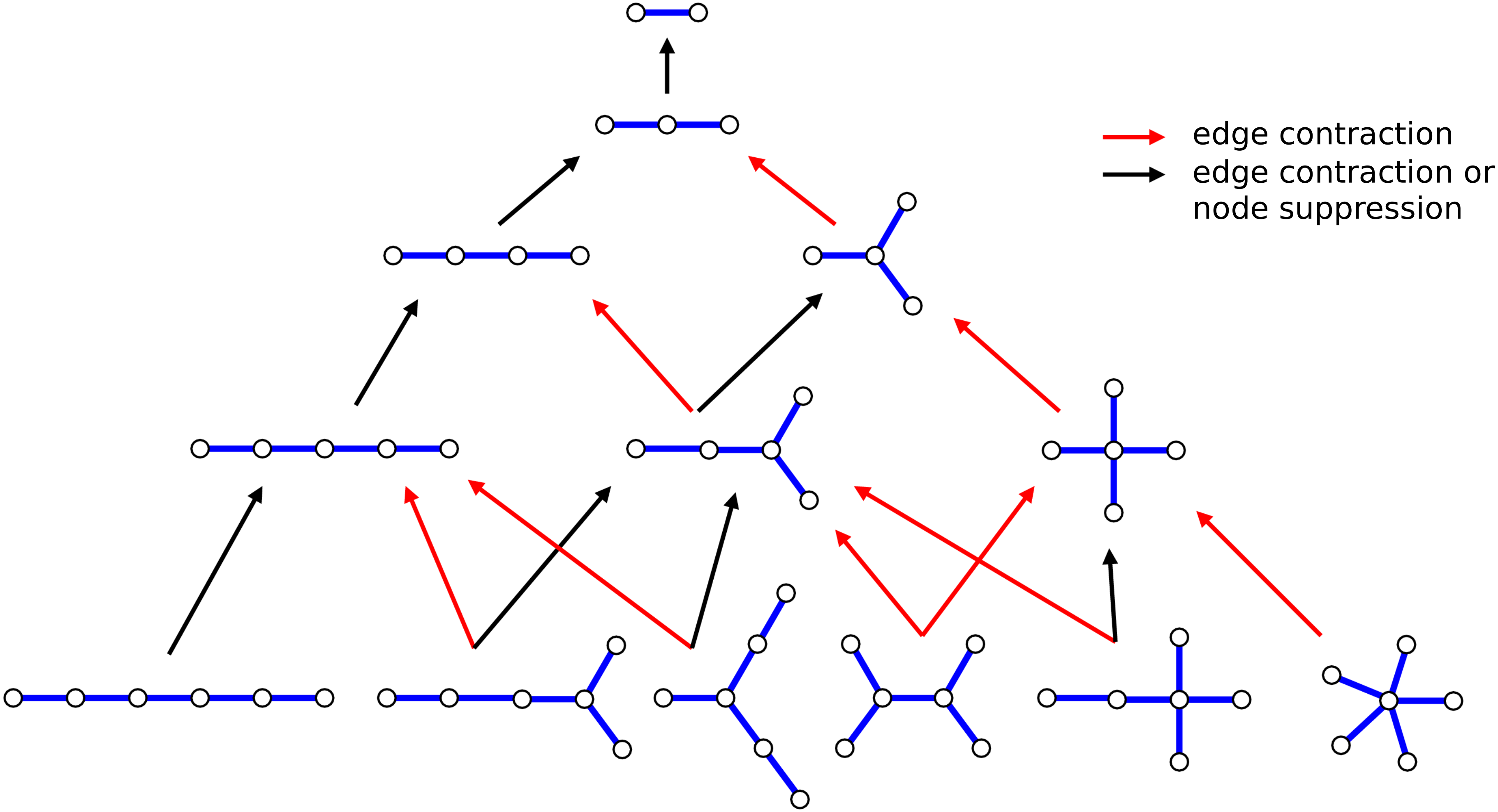
- given a treelet t with k nodes
  - transform t into a treelet with k-1 nodes with an edit operation
- set of possible rewritings $R(t)$



Shape similarity

Edition process

- set of possible rewritings of treelets
  - acyclic graph on the set of tree patterns



Shape similarity

Edition process

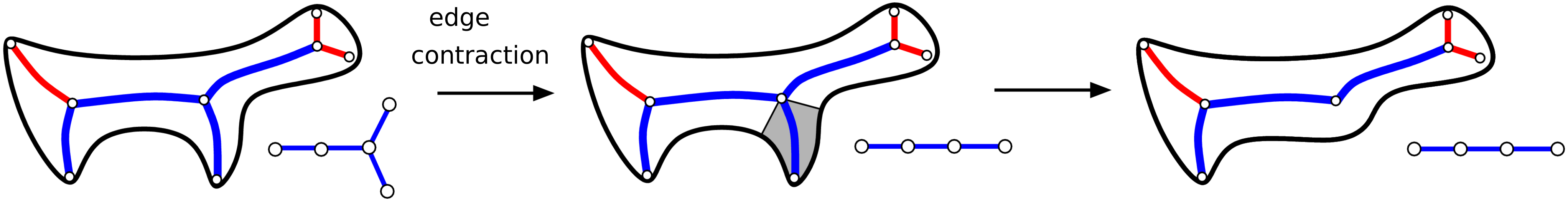
- given a treelet t with k nodes
  - transform t into a treelet with k-1 nodes with an edit operation
- several rewritings $R(t)$
  - retain the one inducing a minimal shape distortion
- **cost assigned to an edit operation $r(t) \in R(t)$**

$$cost(r(t)) = \frac{length(\partial P_{r(t)})}{length(\partial S)}$$

$P_{r(t)}$: part of the shape which is deleted

S: total shape

- **minimal operation**: $\kappa(t) = \underset{r \in R(t)}{argmin}\ cost(r)$



Shape similarity

Edition process

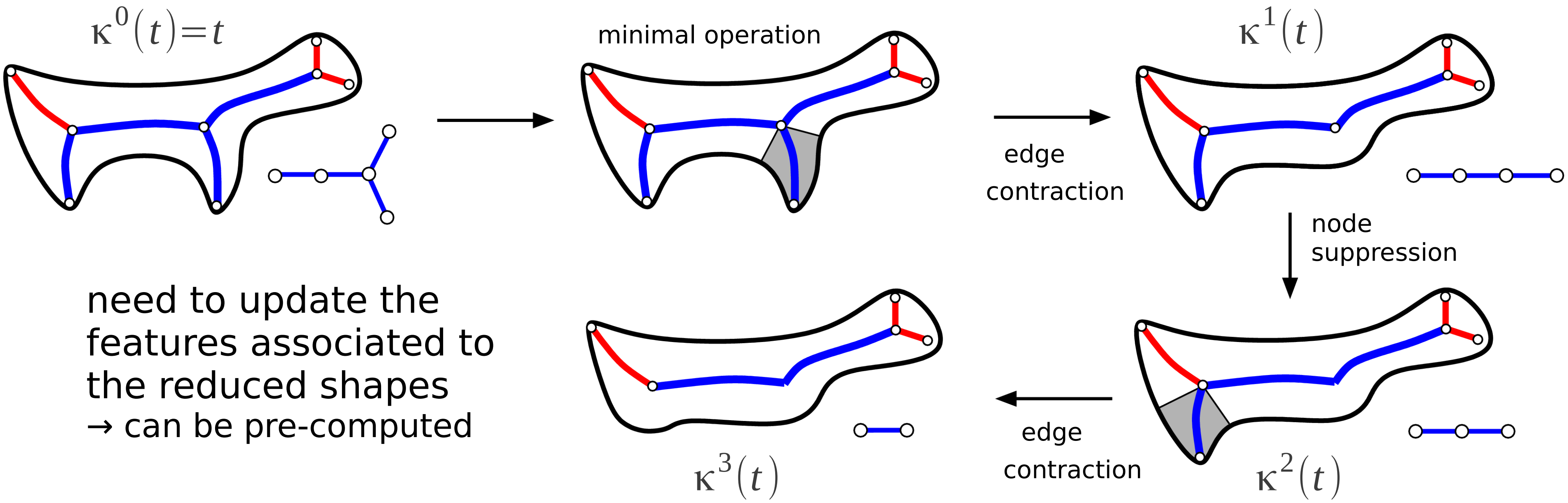
- application of k **successive minimal operations**

$$t=\kappa^0(t),\ \ \kappa^k(t)=\kappa(\kappa(\ldots\kappa(t)))$$

- **cost of k successive minimal operations**

$cost_k(t)$ = sum of the costs of each operation needed to reduce t to $\kappa^k(t)$

- consider the m_t operations needed to reduce t to an edge



need to update the features associated to the reduced shapes
→ can be pre-computed

Shape similarity

Hierarchical treelet kernel based on edition

- similarity between two treelets (equivalent or not)

  = sum of the similarities between the equivalent reduced treelets

$$K_{edit}(t,t') = \sum_{k=0}^{k=m_t} \sum_{l=0}^{l=m_{t'}} \exp\left(-\frac{cost_k(t)+cost_l(t')}{2\sigma_{edit}^2}\right) K_{treelet}(\kappa^k(t),\kappa^l(t'))$$

where equivalent treelets are compared according to all their different correspondances

$$K_{treelet}(t,t') = \begin{cases} \frac{1}{|Sym(t,t')|} \sum_{\psi \in Sym(t,t')} K_\psi(t,t') & if\ Sym(t,t') \neq \emptyset \\ 0 & otherwise \end{cases}$$

and global kernel between two maps (shapes) becomes

$$K_T(M,M') = \frac{1}{|B||B'|} \sum_{t \in B} \sum_{t' \in B'} \lambda_B(t) \lambda_{B'}(t') K_{edit}(t,t')$$

Shape similarity

Hierarchical treelet kernel based on edition

- similarity between of shapes

$$K_T(M,M')=\frac{1}{|B||B'|}\sum_{t\in B}\sum_{t'\in B'}\lambda_B(t)\lambda_{B'}(t')K_{edit}(t,t')$$

  - kernel between bags of combinatorial maps
  - (symmetric) positive-definite
  - weighted mean kernel [Kashima *et al.*, ICML03] [Suard *et al.*, ESANN07] but with edition [Dupe and Brun, ICIAP09]
  - minimal edition operations can be pre-computed for each treelet
  - relies on reacher structures than its counterpart based on paths [Dupe and Brun, ICIAP09 and GbR09]

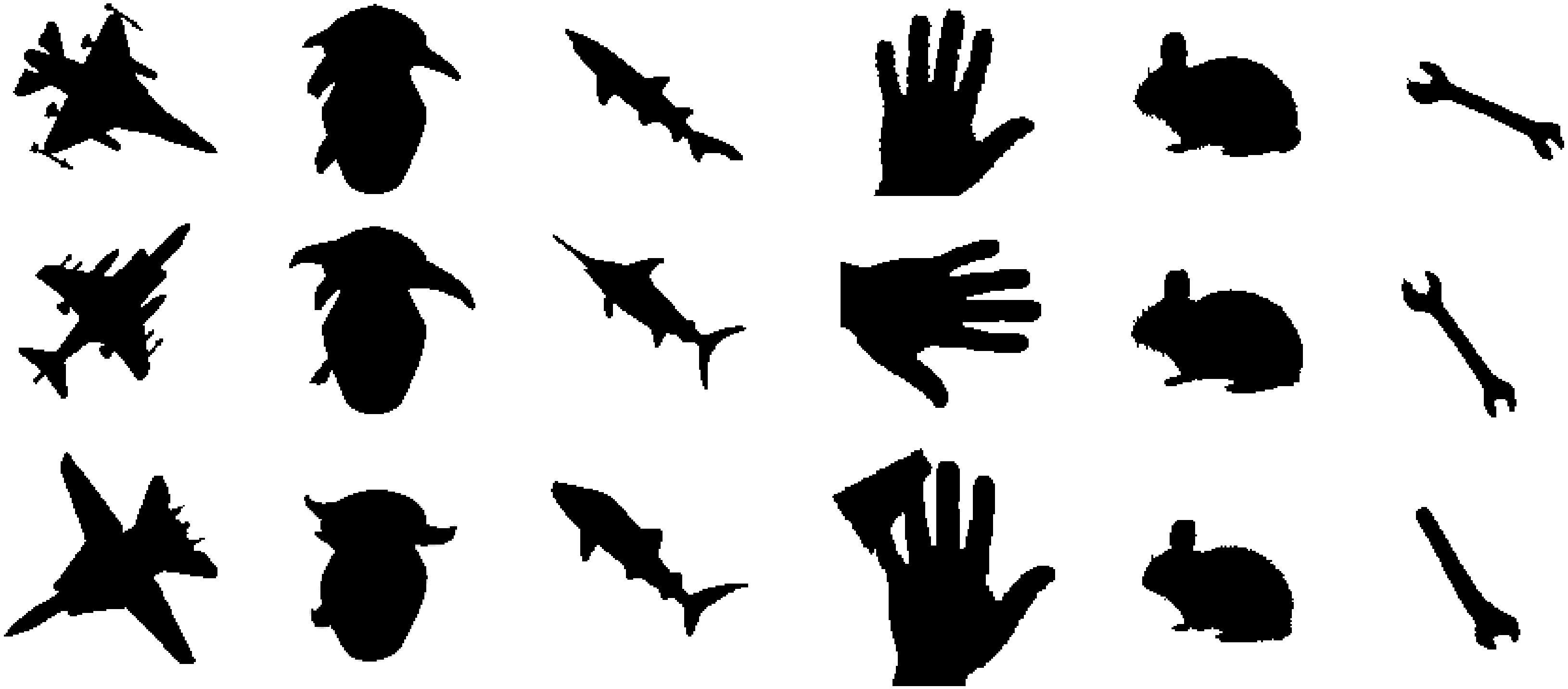
Shape Similarity based on a Treelet Kernel with Edition

1) Introduction
2) Shape similarity
3) Experiments

Experiments

k-NN matching on Kimia25 dataset [Shavit *et al.*, JVCIP98]

- 6 classes, 25 shapes
  - part of the dataset



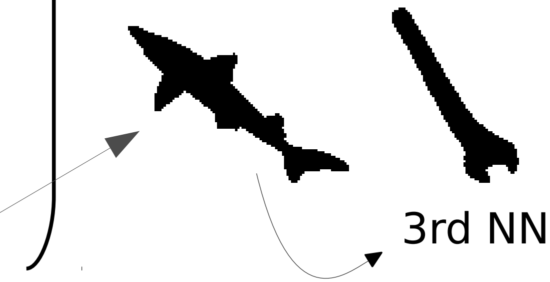
Experiments

k-NN matching on Kimia25 dataset [Shavit *et al.*, JVCIP98]

- 6 classes, 25 shapes
- consider the 3 NN of each shape

Method	k=1	k=2	k=3
edit distance [Neuhaus and Bunke, Patt. Recog. 06]	23	19	18
SID [Sharvit *et al.*, JVCIP 98]	23	21	20
K_T restricted to paths without edition [Dupe and Brun, GbR09]	**24**	**22**	**21**
syntactic matching [Gdalyahu and Weinshall, PAMI 99]	25	21	19
shape context [Belongie *et al.*, PAMI 02]	25	24	22
K_T without edition [Bougleux *et al.*, ICPR12]	**25**	**24**	**22**
ID-shape context [Ling and Jacobs, PAMI 07]	25	24	25
K_T with edition	**25**	**25**	**24**

number of closest shapes belonging to the same class



Experiments

Classification on Kimia25 and 99 [Shavit *et al.*, JVCIP98]

- 6 classes, 25 shapes – 11 classes, 99 shapes
- k-fold cross-validation based on Mahalanobis distance or k-NN

Method	4-NN	Maha.	5-NN	Maha.
edit distance [Neuhaus and Bunke, Patt. Recog. 06]	0.89	0.84	0.927	0.907
bag of trails with edition and covering [Dupe and Brun, GbR09]	0.96	0.952	0.921	0.92
K_T without edition [Bougleux *et al.*, ICPR12]	**0.953**	**0.946**	**0.936**	**0.933**
K_T with edition	**0.981**	**0.975**	**0.962**	**0.958**

Kimia25 Kimia99

$$accuracy = \frac{\#\, true\ positive}{\#\, shapes}$$

- estimation of kernel parameters by a cross-validation on a reduced dataset

Conclusion

Shape similarity

- decomposition of skeletons into treelets embedded in the plane
- weighted mean kernel between bags of treelets
- hierarchical comparison through an edition mechanism
- take into accout rotational and mirror shape symmetries
- improve the behavior of similar kernels
  - without edition
  - with edition and covering, but restricted to paths (trails)

Futur work

- behavior of the kernel on more complex datasets ?
  - need to take into account shapes with holes
- other strategies for computing a minimal set of successive reductions
- extension to 3D shapes and other type of data (images)

Shape Similarity based on a Treelet Kernel with Edition

Thanks for your attention.

Any question !