Shape Similarity based on Combinatorial Maps and a Tree Pattern Kernel

Sébastien Bougleux
Université de Caen Basse-Normandie
CNRS UMR 6072 - GREYC
bougleux@unicaen.fr

François-Xavier Dupé
Aix-Marseille Université
CNRS UMR 7279 - LIF
francois-xavier.dupe@lif.univ-mrs.fr

Luc Brun, Benoît Gaüzère, Myriam Mokhtari
ENSICAEN, CNRS UMR 6072 - GREYC
{*luc.brun,benoit.gauzere,myriam.brun*}*@ensicaen.fr*

Abstract

While the skeleton of a 2D shape corresponds to a planar graph, its encoding by usual graph data structures does not allow to capture its planar properties. Graph kernels may be defined on graph's encoding of the skeleton in order to define a similarity measure between shapes. Such graph kernels are usually based on a decomposition of graphs into bags of walks or trails. These linear patterns do not allow to fully encode the structure of a skeleton on branching points, hence losing important informations about the shape. This paper aims to solve these two drawbacks by using an encoding of the skeleton taking explicitly into account the orientation of the plane and by decomposing the resulting graph model into both linear and nonlinear patterns.

1. Introduction

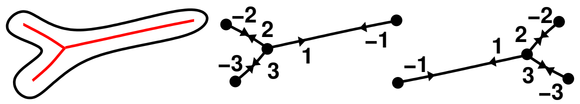
The skeleton is a key feature within the shape recognition framework. It is a thin set, homotopic to the shape, and invariant under Euclidean transformations. Due to the homotopic property, the skeleton of a 2D shape is a planar structure. However, the set of points composing a skeleton does not highlight the structure of a shape. Consequently, the recognition step is usually based on a graph comparison where graphs encode the main properties of skeletons. Several encoding systems have been proposed based on graphs, trees [13] or set of paths [8]. All these encodings use various features attached to nodes and edges, and vary according to the type of highlighted properties of the skeleton. However, usual graph or tree structures do not allow to capture the orientation of the plane on which shapes are embedded.

The recognition of shapes using graph comparisons may be tackled using various methods. A first family of methods is based on graph edit distance and graph matching algorithms [13]. These methods perform in the graph space, which almost contains no mathematical structure, thus forbidding many common mathematical tools. A solution consists to project graphs into a richer space, which can be realized through graph kernels. By using appropriate kernels, graphs can be mapped either explicitly or implicitly into a vector space whose dot product corresponds to the kernel function. Usual graph kernels, such as the random walk kernel [9], are based on a decomposition of a graph into linear patterns such as walks, paths or trails. These patterns do not allow to fully capture the complex structure of the skeleton on branching points. However, several graph kernels based on nonlinear patterns have been proposed in the chemioinformatics framework. These patterns include unlabeled subgraphs [15], tree patterns [11], i.e. trees where a node can appear more than once, and subtrees of limited size [6]. Though, only kernels based on linear patterns have been proposed within the shape recognition framework.

The insights of this paper are twofolds: first we propose to modify the usual graph encoding of the skeleton in order to take explicitly into account the planar embedding of 2D shapes (section 2). Secondly, using this encoding we define a new kernel based on an enumeration of subtrees embedded onto the plane (Section 3). The resulting kernel is evaluated through several experiments in Section 4.

2. Shape representation

Planar properties of the skeleton are not encoded by usual graph data structures. Indeed, these structures do not encode the planar structure of the graph and remain



(a) skeleton (b) M=(D, σ, α) (c) M'=(D, σ^{-1}, α)

Figure 1. Skeleton encoding.

invariant for any permutation of the edges incident to a given node. To overcome this drawback, we propose to encode the skeleton of a 2D shape by a 2D combinatorial map [3]. Such a model may be understood as an encoding of a planar graph taking explicitly into account the orientation of the plane.

A 2D combinatorial map (Fig.1(b)) is defined by the triplet $M = (D, \sigma, \alpha)$, where D corresponds to the set of darts (or half-edges) obtained by decomposing each edge into two darts, $\sigma : D \to D$ is a permutation whose cycles correspond to the sequence of darts encountered when turning counter-clockwise around each node. Note that permutation σ explicitly encodes the orientation of edges around each node. Finally, $\alpha : D \to D$ is a fixed point free involution whose cycles correspond to the two darts associated to a same edge. The encoding of a skeleton by such a map is performed by representing each branch by two darts defining one edge (a cycle of α). The orientation of branches, around branching points, is explictly encoded by the cycles of permutation σ.

Our kernel between combinatorial maps (Section 3) is based on their decomposition into trees of limited size, each tree being encoded by a sub combinatorial map. In order to identify similar subtrees, an isomorphism relationship between combinatorial maps must be defined [4]. An isomorphism of a map $M_1 = (D_1, \sigma_1, \alpha_1)$ on a map $M_2 = (D_2, \sigma_2, \alpha_2)$ is defined as a bijection $\psi : D_1 \to D_2$ that satisfies the two following properties: (i) $\psi \circ \alpha_1 = \alpha_2 \circ \psi$, and (ii) $\psi \circ \sigma_1 = \sigma_2 \circ \psi$. In other terms, an isomorphism maps the two darts of an edge onto two darts of a same edge, and preserves the orientation of edges around nodes. The set of such bijections forms the isomorphism group $\mathrm{Isom}(M_1, M_2)$. If $M_1 = M_2$, this set is called the automorphism group of M_1, denoted by $\mathrm{Aut}(M_1)$.

The combinatorial map encoding a skeleton is invariant to translations, rotations, and scaling (as long as the skeleton remains also invariant). However, a reflection transforms a map $M = (D, \sigma, \alpha)$ onto a map $M' = (D, \sigma^{-1}, \alpha)$, hence reversing the orientation of the map (Fig 1(c)). The map M' is called the trivial mirror symmetric of M. More generally, we say that two maps M_1 and M_2 are mirror (or orientation-reversing) symmetric if it exists a bijection $\phi : D_1 \to D_2$ satisfying: (i) $\phi \circ \alpha_1 = \alpha_2 \circ \phi$, and (ii) $\phi \circ \sigma_1 = \sigma_2^{-1} \circ \phi$. The set of such bijections forms the group of mirror symmetries $\mathrm{Mir}(M_1, M_2)$. The set of bijections reversing the orientation of a map M is denoted by $\mathrm{Aut}_R(M)$. Such a set may be deduced from $\mathrm{Aut}(M)$ by composing each automorphism of $\mathrm{Aut}(M)$ with the trivial symmetry operation [2]. Finally, we say that two combinatorial maps are symmetric (or equivalent) if they are orientation-preserving or orientation-reversing symmetric. We denote by $\mathrm{Sym}(M_1, M_2) = \mathrm{Isom}(M_1, M_2) \cup \mathrm{Mir}(M_1, M_2)$ the set of such bijections from M_1 to M_2.

Shape features. In order to attach features to a combinatorial map encoding a skeleton, we define a set of node and edge labels (V and E), each node and edge label being respectively associated to a single cycle σ and α of the map [3].

Following [8], our edge and node attributes combine local and global features. After several experiments we selected 2 features $(f_{E,i}(e))_i$ attached to each edge of E, and 2 features $(f_{V,i}(v))_i$ attached to each node of V.

Following [8, 5], we model the evolution of the radius of the maximal inscribed disk along a branch by a regression polynomial of order 4. A first edge feature corrresponds thus to the 4 polynomial coefficients modeling the evolution of the radius along the branch.

The second feature associates, to each edge, the length of the shape boundary which contributes to the creation of its associated branch, normalized by the total length of the shape boundary [5]. Such a feature encodes the part of the shape boundary implied in the creation of the branch. This measure, defined as a function $w : E \to \mathbb{R}_+$, may thus be understood both as a relevant feature of an edge and as a measure of its relevance according to the shape.

Regarding nodes, our first feature assigns to each node of V the sum of the relevances of its incident edges. Our second feature, associates to each node its minimal geodesic distance, inside the shape, to the set of geodesic centroids. Geodesic centroids correspond to the points of the skeleton having a maximal inscribed disk radius. Such points can thus be efficiently deduced from the skeleton. This distance is normalized by the square root of the shape area.

3. Shape similarity

Our kernel between combinatorial maps is based on a decomposition of each combinatorial map into a bag of submaps. Similarly to [6], the submaps are restricted to unlabeled and unrooted trees having between 3 and 6 nodes. As illustrated by Fig. 2, these trees form a dic-

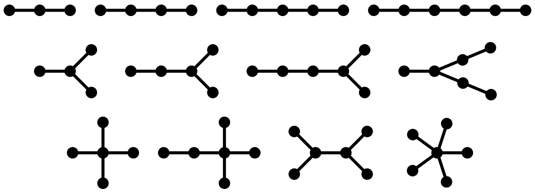


Figure 2. Our set $\mathbb{T}$ of 12 tree patterns.

tionary of tree patterns, denoted by $\mathbb{T}$. This choice corresponds to a compromise between the expressiveness of our kernel and the time required to enumerate subtrees. Tree patterns of $\mathbb{T}$ are extracted [6] using a depth first search strategy from each extremity of paths and from each node having a degree greater than 3. Their enumeration, in a given combinatorial map M, is performed in $O(|V|d^5)$, where $|V|$ is the number of nodes and d is the maximal node's degree in M.

Each instance of a tree pattern, obtained from an enumeration, is called a treelet. Note that unlike [6], the orientation of edges around each node of a treelet is encoded by a combinatorial map.

Kernel between maps. Let $\mathcal{B}$ and $\mathcal{B}'$ denote two bags of treelets extracted from combinatorial maps M and M' respectively. Inspired by marginalized kernels [9], our kernel is defined as a weighted sum of minor kernels between all pairs of treelets of $\mathcal{B} \times \mathcal{B}'$:

$$K_{\mathbb{T}}(M, M') = \tfrac{1}{|\mathcal{B}||\mathcal{B}'|} \sum_{t \in \mathcal{B}} \sum_{t' \in \mathcal{B}'} \lambda_{\mathcal{B}}(t) \lambda_{\mathcal{B}'}(t') K(t, t').$$

Kernel K corresponds to a minor kernel between treelets, while the function $\lambda_{\mathcal{B}} : \mathcal{B} \to \mathbb{R}_+$ represents the relevance of each treelet. Following [5] in the case of trail patterns, the relevance of a treelet, relatively to its bag, is defined by $\lambda_{\mathcal{B}}(t) = w(t) / \max_{t' \in \mathcal{B}} w(t')$, where $w(t) = \sum_{e \in t} w(e)$. This weight allows to reduce the influence of treelets encoding non relevant parts of a shape.

Treelet kernel. Our minor kernel between two treelets t and t' is set to 0 if $\mathrm{Sym}(t, t') = \emptyset$. In such a case, both treelets correspond to different tree patterns.

If $\mathrm{Sym}(t, t') \neq \emptyset$, both treelets are considered as structurally equivalent and their similarity must be defined from the similarities of the features attached to their respective nodes and edges. The set of mappings between nodes and edges of both treelets, which preserve (or inverse) the orientation, corresponds to $\mathrm{Sym}(t, t')$. The proposed kernel is given by:

$$K(t, t') = \tfrac{1}{|\,\mathrm{Sym}(t, t')|} \sum_{\psi \in \mathrm{Sym}(t, t')} K_\psi(t, t'). \qquad (1)$$

Kernel K_ψ is defined as the product of the similarities between each pair of nodes and each pair of edges provided by the mapping $\psi : t \to t'$:

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

where kernel K_V (resp. K_E) encodes the similarity between node's features (resp. edge's feature). It is defined as a tensor product of Gaussian kernels between each feature:

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

where A corresponds to V or E and a corresponds alternatively to a node v or an edge e. This last kernel being definite positive, kernel $K_{\mathbb{T}}$ is also definite positive.

Evaluation of (1) supposes to traverse $\mathrm{Sym}(t, t')$ when this set is non empty. In such a case, t and t' are equivalent and one can easily show [2] that $\mathrm{Sym}(t, t')$ corresponds to $\mathrm{Aut}(t_p) \cup \mathrm{Aut}_R(t_p)$, where t_p is the tree pattern of $\mathbb{T}$ isomorphic to both t and t'. Such a set can be pre-computed [4, 2] for any tree pattern of $\mathbb{T}$. The maximal size of this set for any pattern of $\mathbb{T}$ is equal to 5. Note that $\mathrm{Sym}(t, t')$ contains much less elements than the set of usual graph isomorphisms. Indeed, $\mathrm{Sym}(t, t')$ does not contain bijections of edges and nodes which violate the orientation constraint. The evaluation of (1), using combinatorial maps, is thus both more precise and more efficient than its counterpart based on graphs.

4. Experiments

The behaviour of the treelet-based kernel is analyzed through two datasets: Kimia25 and Kimia99 [14], which contain respectively 25 and 99 discrete shapes, which are organized into 6 and 11 classes respectively. Two experiments are performed, one involving indexation and one classification.

***k*-NN matching.** The first experiment computes, for each shape of Kimia25 dataset, its $k = 1, 2, 3$ closest shapes according to a given similarity measure, ours being defined by kernel $K_{\mathbb{T}}$. Values displayed in Table 1 represent, for each value of k, the number of closest shapes belonging to the same class than the input one [10].

The parameters of our kernel $K_{\mathbb{T}}$ (the σ_k associated to each feature) have been optimized through experiments in order to obtain the best global match. As shown by lines 2 and 5 of Table 1, the use of nonlinear patterns, over linear ones, improves the efficiency of

Table 1. Matching on Kimia25 dataset.

	Method	$k=1$	$k=2$	$k=3$
1	SID [14]	23	21	20
2	$K_{\mathbb{T}}$ (paths only)	**24**	**22**	**21**
3	Syntactic matching [7]	25	21	19
4	Shape Context [1]	25	24	22
5	$K_{\mathbb{T}}$	**25**	**24**	**22**
6	ID-Shape Context [10]	25	24	25

Table 2. Classification accuracy.

Method	*Accuracy*			
	Kimia25		*Kimia99*	
	k-NN	*Maha.*	*k-NN*	*Maha.*
Edit distance [12]	0.89	0.84	0.927	0.907
Trails [5]	0.96	0.952	0.921	0.92
$K_{\mathbb{T}}$	**0.953**	**0.946**	**0.936**	**0.933**

our kernel $K_{\mathbb{T}}$. Results obtained using $K_{\mathbb{T}}$ (line 5) are only outperfomed by [10], which provides a result very close to the optimum. A similar behaviour has been observed on Kimia99 dataset. Note that [10] proposed a matching method which does not induce a definite positive similarity measure. Such a drawback prevents [10] to readily combine its similarity measure with complex numerical tools such as PCA or SVM.

Classification. In this experiment, we compare the kernel $K_{\mathbb{T}}$ with two state-of-the-art kernels. These two other methods use edition mechanisms to deal with the structural noise inside graphs. For each method, the best kernel parameters have been estimated with a cross-validation on a reduced training set of Kimia25 or Kimia99 datasets. Then, a k-fold cross-validation, based on a Mahalanobis distance to each class and a k-NN, is computed to evaluate the efficiency of the kernels ($k=4$ for Kimia25, and $k=5$ for Kimia99). The resulting accuracies (number of true positive divided by the total number of shapes) are reported in Table 2. Our kernel outperforms the one based on a Gaussian edit distance [12], and obtains a result close to the one provided by trail kernels, which uses rewriting and covering mechanisms [5]. Note that our kernel seems to be robust against structural noise despite the fact that it does not integrate any edition mechanism.

5. Conclusion

We have defined a new kernel based on a decomposition of combinatorial maps into tree patterns for shape recognition. Such a kernel is more expressive than kernels based on linear patterns and takes explicitly into account the orientation of the branches of a skeleton around each branching point. Experiments have shown the competitiveness of our kernel relatively to methods incorparating edition mechanisms. Such mechanisms will be studied in a future work in order to improve the robustness of our kernel against structural noise.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on PAMI*, 24(4):509–522, 2002.

[2] S. Bougleux and L. Brun. Symmetry group of 2d combinatorial maps. Technical report, GREYC, France, 2012.

[3] L. Brun, M. Mokhtari, and J. P. Domenger. Incremental modifications on segmented image defined by discrete maps. *Journal of Visual Communication and Image Representation*, 14:251–290, 2003.

[4] R. Cori. Computation of the automorphism group of a topological graph embedding. Technical Report I-8612, Université Bordeaux 1, France, 1985.

[5] F.-X. Dupé and L. Brun. Tree covering within a graph kernel framework for shape classification. In *ICIAP*, volume 5716 of *LNCS*, 2009.

[6] B. Gauzere, L. Brun, and D. Villemin. Two new graph kernels and applications to chemoinformatics. In *In 8th GbR*, volume 6658 of *LNCS*, pages 112–122, 2011.

[7] Y. Gdalyahu and D.Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Trans. on PAMI*, 21(2):1312–1328, 1999.

[8] W.-B. Goh. Strategies for shape matching using skeletons. *Computer Vision and Image Understanding*, 110(3):326–345, 2008.

[9] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *Proc. of the 20st Int. Conf. on Machine Learning*, pages 321–328, 2003.

[10] H. Ling and D. W. Jacobs. Shape classification using the inner-distance. *IEEE Trans. on PAMI*, 29(2):286–299, 2007.

[11] P. Mahé and J.-P. Vert. Graph kernels based on tree patterns for molecules. *Machine Learning*, 75:3–35, 2009.

[12] M. Neuhaus and H. Bunke. Edit-distance based kernel for structural pattern classification. *Pattern Recognition*, 39:1852–1863, 2006.

[13] M. Pelillo, K. Siddiqi, and S. Zucker. Matching hierarchical structures using association graphs. *IEEE Trans. on PAMI*, 21(11):1105–1120, Nov 1999.

[14] D. Sharvit, J. Chan, H. Tek, and B. B. Kimia. Symmetry-based indexing of image databases. *Journal of Visual Communication and Image Representation*, 9(4):366–380, 1998.

[15] N. Shervashidze, S. V. N. Vishwanathan, T. H. Petri, K. Mehlhorn, and K. M. Borgwardt. Efficient graphlet kernels for large graph comparison. In *Proc. of the 12th Int. Conf. on Artificial Intelligence and Statistics*, pages 488–495, 2009.