Shape Similarity based on a Treelet Kernel with Edition[*]

Sébastien Bougleux [1], François-Xavier Dupé [2], Luc Brun [3], and Myriam Mokhtari [3]

[1] Université de Caen Basse-Normandie, GREYC CNRS UMR 6072
`bougleux@unicaen.fr`
[2] Aix-Marseille Université, CNRS UMR 7279 - LIF
`francois-xavier.dupe@lif.univ-mrs.fr`
[3] ENSICAEN, GREYC CNRS UMR 6072
`{luc.brun,myriam.brun}@ensicaen.fr`

Abstract. Several shape similarity measures, based on shape skeletons, are designed in the context of graph kernels. State-of-the-art kernels act on bags of walks, paths or trails which decompose the skeleton graph, and take into account structural noise through edition mechanisms. However, these approaches fail to capture the complexity of junctions inside skeleton graphs due to the linearity of the patterns. To overcome this drawback, tree patterns embedded in the plane have been proposed to decompose the skeleton graphs. In this paper, we reinforce the behaviour of kernel based on tree patterns by explictly incorporating an edition mechanism adapted to tree patterns.

Keywords: shape similarity, kernel methods, tree patterns, edition

1 Introduction

Several 2D shape representations and signatures have been proposed as a basis of shape recognition and classification, in particular the medial axis (or skeleton) and the associated medial axis transform. Indeed, the medial axis is a geometric graph homotopic to the shape and the medial axis transform allows to reconstruct the shape. However, the medial axis does not highlight enough the local shape properties needed for shape comparison, especially for the design of similarity measures. To overcome this drawback, suitable local shape properties are attached to the elements of the graph encoding the skeleton, leading to graph-based similarity measures.

Graph comparison can be performed by various methods, for example, graph edit distance and graph matching algorithms [1] form a first family. However, they are defined in graph space which almost contains no mathematical structure, thus prohibiting the use of many common tools. One solution is to project graphs into a richer (or more flexible) space. Such a projection can be done

---

through graph kernels. With appropriately defined kernels, graphs can be implicitly (sometimes explicitly) mapped into a vector space whose dot product corresponds to the kernel. Most of graph kernels rely on graph decomposition into walks, paths or trails [2–5]. However, these patterns fail to capture the complexity of junctions inside graphs, and so the branching points of the skeletons. One solution has been proposed in the chemioinformatics framework, where several graph kernels based on nonlinear patterns have been proposed. These patterns include unlabeled subgraphs [6], tree patterns [7], i.e. trees where a node can appear more than once, and subtrees of limited size [8]. Following [8], we have recently proposed a kernel based on a decomposition of skeletons into bags of subtrees embedded in the plane [9]. While this kernel provides good classification results compared to more sophisticated ones, it does not include any mechanism that would allow to be robust to spurious branches inside skeletons.

As the skeleton is very sensitive to small variations of the shape boundary (noise or small elongations), spurious nodes and edges (structural noise) are present inside its graph structure. In order to tackle such problems, an edition mechanism has been proposed for kernels based on bags of paths [4, 5]. Given a pertinence measure of each egde and node, the idea is to compute for each path, a sequence of reduced paths by successively removing their less pertinent part. Then, the resulting graph kernels are based on hierarchical comparisons between features attached to the elements of the rewritten paths.

This paper presents an extension of the treelet kernel proposed in [9] by incorporating an edition mechanism inspired by [4, 5]. First we recall our shape representation, which is based on a combinatorial map encoding of the skeleton, allowing to explicitly take into account its embedding in the plane (Section 2). Based on this encoding, we describe our extension of the treelet kernel which improves its robustness against structural noise (Section 3). Finally, several experiments are proposed in order to evaluate the performance of the resulting kernel and to measure the performances of our edition mechanism (Section 4).

2   Shape representation

Usual graph-based encoding of the skeleton of a 2D shape do not take into account its planar properties, and thus remain invariant for any permutation of adjacent branches. To overcome this drawback, the skeleton can be encoded by a 2D combinatorial map [9]. Such a model may be understood as an encoding of a planar graph taking explicitly into account the orientation of the plane.

Combinatorial map encoding. As illustrated by Fig.1(b), a 2D combinatorial map (e.g. [10]) 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 pairs
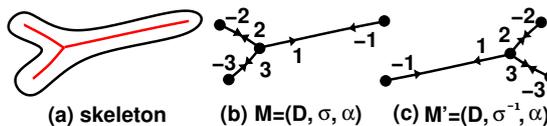
Fig. 1. Skeleton encoding: $\sigma = (-1)(1\,2\,3)(-2)(-3)$, $\alpha = (1\,-1)(2\,-2)(3\,-3)$.

of darts, each pair corresponding to an 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.

The kernel between two shapes, described in Section 3, is based on the decomposition of their associated combinatorial maps into submaps having a tree structure. The identification of similar submaps relies on the computation of the symmetry group between submaps. The symmetry group $\mathrm{Sym}(M_1, M_2)$ from a map $M_1 = (D_1, \sigma_1, \alpha_1)$ to a map $M_2 = (D_2, \sigma_2, \alpha_2)$ defines the set of bijections $\psi : M_1 \to M_2$ that align the edges of M_1 onto the edges of M_2, while preserving or reversing their orientation around the nodes. Such bijections describe both rotational and mirror symmetries needed to align the two maps. They satisfy the following relations: (i) $\psi \circ \alpha_1 = \alpha_2 \circ \psi$, and (ii) $\psi \circ \sigma_1 = \sigma_2 \circ \psi$ or (iii) $\psi \circ \sigma_1 = \sigma_2^{-1} \circ \psi$. Relations (i) and (ii) correspond to a rotational symmetry, in which case ψ is a map isomorphism [11]. Relations (i) and (iii) correspond to a mirror symmetry and ψ is considered as a reflection [9, 12]. If $M_1 = M_2 = M$, then the symmetry group $\mathrm{Sym}(M_1, M_2)$ is equal to the set of permutations $\psi : M \to M$ which satisfy (i), and (ii) or (iii). This set is respectively composed of the automorphism group of M, noted $\mathrm{Aut}(M)$, and the automorphism group of the trivial mirror symmetric of M [4], noted $\mathrm{Aut}_R(M)$ [9]. These two groups can be computed by Cori's algorithm (see [11, 12] for more details).

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 [10]. We use mainly the same shape features as [9]. Let $f_E = (f_{E,i}(e))_i$ be the features attached to each edge of E, and $f_V = (f_{V,i}(v))_i$ the ones attached to each node.

Following [3, 5], a first edge feature corrresponds to the 4 polynomial coefficients of a regression polynomial of order 4 that modelize the evolution of the radius (of the inscribed disk) along the branch. The second edge feature associates the length of the shape boundary which contributes to the creation of the branch, normalized by the total length of the shape boundary in order to be invariant to scaling (see [5] for more details). 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.

---

[4] The trivial mirror symmetric of M is the map $M' = (D, \sigma^{-1}, \alpha)$ constructed by reversing the orientation of the darts around nodes (see Fig.1(c)).
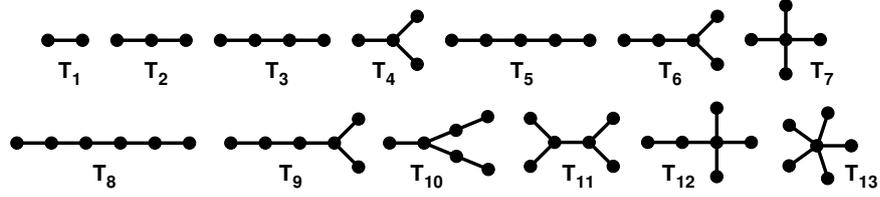
Fig. 2. The set $\mathbb{T}$ of tree patterns.

Regarding nodes of V, the first feature corresponds to the sum of the relevances of its incident edges. The second feature associates to each node its Euclidean distance to the gravity center of the shape, normalized by the square root of the shape area in order to be invariant to scaling.

3   Shape similarity

Based on the previous combinatorial map representation, similarity between two shapes relies on a decomposition of each map into a bag of subtrees [9], and on a hierarchical kernel between these subtrees in order to be robust to structural noise. This kernel replaces the edition mechanisms proposed in [5] in the context of subtrees.

3.1   Bag of treelets

Given a shape represented by a combinatorial map M, and features attached to its nodes and edges, M is transformed into a bag of submaps having a tree structure. Each submap, together with its corresponding features, represent a part of the shape. Following [8, 9], the enumeration of the submaps is restricted to unlabeled and unrooted trees having between 2 and 6 nodes. As illustrated by Fig. 2, these trees of limited size form a dictionary of 13 *tree patterns*, denoted by $\mathbb{T} = \{T_p\}_{p=2,\dots,13}$. The choice of the bounds on the number of nodes corresponds to a compromise between the expressiveness of the resulting bag and the time required to enumerate predefined subtrees.

An instance t of a tree pattern of $\mathbb{T}$ in M is called a *treelet*. It is represented as a 5-uplet (V, E, f_V, f_E, w), where f_V and f_E denote the features associated to the part of the shape described by t, and $w(t) = \sum_{e \in E(t)} w(e)$ represents its relevance according to the shape (the normalized boundary length induced by the edges of t). In practice, a treelet can be encoded by the index p of the corresponding tree pattern $T_p \in \mathbb{T}$, and an injection from edges of T_p to edges of M. The extraction of all the treelets from M can be performed by an enumeration process similar to the one proposed by [8]. The only difference is the preservation of the orientation of edges around each node.

3.2 Kernel between bags of treelets

Let $\mathcal{B}$ and $\mathcal{B}'$ denote two bags of treelets extracted from combinatorial maps M and M' respectively. Inspired by marginalized kernels [2], we have proposed in [9] a kernel defined as a weighted sum of minor kernels between all pairs of treelets of $(\mathcal{B} \times \mathcal{B}') \setminus \{(t, t') \in \mathcal{B} \times \mathcal{B}' : |V(t)| = |V(t')| = 2\}$ (we do not consider treelets isomorphic to tree pattern T_1):

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

Kernel K corresponds to a minor kernel between treelets (see the following sections). The function $\lambda_{\mathcal{B}} : \mathcal{B} \to \mathbb{R}_+$ represents the relevance of each treelet relatively to its bag, which is defined by $\lambda_{\mathcal{B}}(t) = w(t) / \max_{t' \in \mathcal{B}} w(t')$. This weight allows to reduce the influence of treelets encoding non relevant parts of a shape.

3.3 Treelet kernel

Let t and t' be two treelets representing parts of shapes. When they correspond to the same tree pattern $T_p \in \mathbb{T}$ (t and t' are structurally isomorphic to T_p), they can differ according to the features attached to their nodes and edges. Also, depending on the tree pattern, several matches between the two treelets are possible. In order to take into account both rotational and mirror symmetries of the shapes, the set of mappings between t and t' must preserve their orientations, but also reverse their orientations. This set corresponds to the symmetry group $\mathrm{Sym}(t, t')$, which is equivalent to $\mathrm{Aut}(T_p) \cup \mathrm{Aut}_R(T_p)$, and which can thus be easily pre-computed for each tree pattern [12].

In order to measure the similarity between the treelets, we have proposed in [9] a positive-definite kernel defined as the average of similarities between their different matches derived from $\mathrm{Sym}(t, t')$:

$$K_{\mathrm{treelet}}(t, t') = \begin{cases} \dfrac{1}{|\,\mathrm{Sym}(t,t')|} \displaystyle\sum_{\psi \in \mathrm{Sym}(t,t')} K_{\psi}(t, t') & \text{if } \mathrm{Sym}(t, t') \neq \emptyset, \\ 0 & \text{else.} \end{cases} \tag{2}$$

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:

$$K_A(a, a') = \prod_{k=1}^{n_A} \exp \left(-\frac{\|f_{A,k}(a) - f_{A,k}(a')\|^2}{2\sigma_k^2} \right),$$
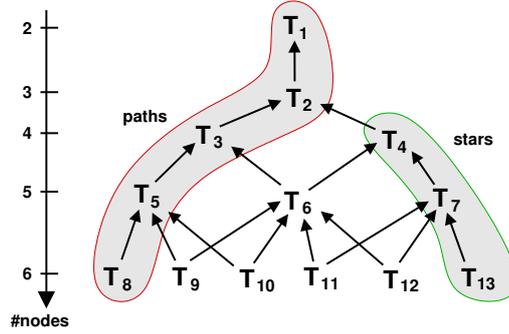
Fig. 3. Edition rules for the tree patterns of $\mathbb{T}$. Arcs represent transitions between treelets by either node suppression or edge contraction.

where A corresponds to V or E and a corresponds to a node v or an edge e. Kernel K_{Sym} can be seen as an extension of kernels on paths, trails or walks [2, 5] to trees embedded in the plane.

Experiments in [9] show the efficiency of treelet kernel K_{treelet}. In particular, the results are close to the one obtained by [5], which includes an edition process to reduce structural noise and to enhance similarity between closely related treelets. In the sequel, we extend this process to treelets in order to improve the robustness of kernel $K_{\mathbb{T}}$.

3.4   Hierarchical treelet kernel

Since shape skeletons are sensitive to small deformations of the shape, the similarity measure K_{treelet} between two treelets can be affected by structural noise. Also, two treelets not corresponding to the same tree pattern $(\text{Sym}(t, t') = \emptyset)$ may be similar up to some node suppressions or edge contractions. Following [5] in the case of paths, each treelet of a bag is transformed into a sequence of smaller ones through an edition process. Since deformations of the shape can be formalized by additions of nodes and edges, the two operations used to construct the sequence of treelets are *node suppression* and *edge contraction*. Node suppression corresponds to cut the parts of the shape connected to the treelet by the node. Edge contraction corresponds to a contraction of the shape. Each edge of the treelet is candidate to this operation. Node suppression is restricted to nodes of degree 2. This operation is topologically equivalent to the contraction of one of the two edges incident to the node. The set of possible rewritings of treelets defines an acyclic graph on the set $\mathbb{T}$ of tree patterns (see Fig. 3).

Let t be a treelet with k nodes, structurally equivalent to a tree pattern $T_p \in \mathbb{T}$. Depending on T_p, several nodes or edges can be suppressed or contracted in order to obtain a treelet with $k-1$ nodes. The operation which induces a minimal distortion of the shape is retained. In order to encode this notion of distortion, a cost is assigned to each operation. This cost corresponds to the

Table 1. Matching on Kimia25 dataset.

	Method	$k=1$	$k=2$	$k=3$
1	Edit distance [13]	23	19	18
2	SID [14]	23	21	20
3	$K_{\mathbb{T}}$ with K_{treelet} only restricted to paths [9]	**24**	**22**	**21**
4	Syntactic matching [15]	25	21	19
5	Shape Context [16]	25	24	22
6	$K_{\mathbb{T}}$ with K_{treelet} only [9]	**25**	**24**	**22**
7	ID-Shape Context [17]	25	24	25
8	$K_{\mathbb{T}}$ with K_{edit}	**25**	**25**	**24**

boundary of the part of the shape which is deleted: the relevance of the edge in the case of contraction, and the sum of relevances of the deleted edges in the case of suppression (see [5] for more details). The retained operation is the one having a minimal cost. Let κ be the application of this cheapest treelet edition, and let κ^k be the application of k successive editions. Then, the similarity between two treelets is measured by the kernel:

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

where m_t is the number of editions needed to transform t into a treelet equivalent to the tree pattern T_1 (an edge), and $w_k(t)$ is the cost associated to each reduced treelet operation $\kappa^k(t)$, defined as the sum of the costs of the k editions. Each feature associated to a reduced treelet is defined as a modification of the initial features associated to t according to the deformation of the shape (see [5] for more details).

Contrary to K_{treelet}, K_{edit} allows to compare two treelets which are not equivalent to a same tree pattern. Also, one can note that K_{edit} relies upon reacher structures than its counterpart based on paths [5], and thus more candidate operations need to be tested during the construction of the sequence of reduced treelets. But the sequences can be easily pre-computed for each treelet during the construction of the bags, as well as the associated features which have been modified by the edition process. So the proposed extension does not affect the computation of the kernel, and since the maximal number of editions is always 4, it is less time consuming than [5] (as long as the number of editions used in [5] is more than 4).

4 Experiments

In order to illustrate the behaviour of the proposed kernel $K_{\mathbb{T}}$ with the treelet kernel K_{edit}, we have considered the same experiments as in [9], that is k-NN matching and classification of the shapes of Kimia25 and Kimia99 datasets [14]. They contain respectively 25 and 99 discrete shapes, which are organized into 6

Table 2. Classification accuracy.

Method	Accuracy			
	Kimia25		Kimia99	
	k-NN	Maha.	k-NN	Maha.
Edit distance [13]	0.89	0.84	0.927	0.907
Trails [5]	0.96	0.952	0.921	0.92
$K_{\mathbb{T}}$ with K_{treelet} only [9]	**0.953**	**0.946**	**0.936**	**0.933**
$K_{\mathbb{T}}$ with K_{edit}	**0.981**	**0.975**	**0.962**	**0.958**

and 11 classes.

k-NN matching. For each shape of Kimia25 dataset, its $k=1,2,3$ closest shapes are computed according to a 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. The parameters of $K_{\mathbb{T}}$ (the σ_k associated to each feature as well as σ_{edit}) have been optimized through experiments in order to obtain the best global match. As shown by lines 3 and 6 of Table 1, the use of nonlinear patterns over linear ones improves the efficiency of kernel $K_{\mathbb{T}}$. Line 8 shows the improvement obtained by incorporating K_{edit} into $K_{\mathbb{T}}$. One can also remark that the proposed kernel provides a result very close to the optimum, and slightly improves the one obtained by [17]. A similar behaviour has been observed on Kimia99 dataset. Note that [17] proposed a matching method which does not induce a definite positive similarity measure. Such a drawback prevents [17] to readily combine its similarity measure with complex numerical tools such as PCA or SVM.

Classification. The second experiment compares the proposed kernel $K_{\mathbb{T}}$ with two state-of-the-art kernels. 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. Again, our kernel with edition outperforms our previous kernel based on treelets, as well as the the one based on a Gaussian edit distance [13] and the one provided by trail kernels [5]. Note that this last kernel also use convering mechanisms to reduce the size of the bags.

5   Conclusion

To measure the similarity between 2D shapes, we have presented an extension of the kernel based on a decomposition of skeleton graphs into treelets embedded in the plane [9]. The extension, designed to take explicitly into account structural noise, relies on a hierarchical comparison of the treelets through an edition

mechanism. Experiments show that the proposed kernel improves the results obtained with our previous kernel without edition mechanisms, as well as the ones obtained by several state-of-the-art methods.

References

1. Pelillo, M., Siddiqi, K., Zucker, S.: Matching hierarchical structures using association graphs. IEEE Trans. on PAMI **21**(11) (Nov 1999) 1105–1120
2. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernels between labeled graphs. In: Proc. of the 20st Int. Conf. on Machine Learning. (2003) 321–328
3. Goh, W.B.: Strategies for shape matching using skeletons. Computer Vision and Image Understanding **110**(3) (2008) 326–345
4. Dupé, F.X., Brun, L.: Edition within a graph kernel framework for shape recognition. In: In 7th Workshop on Graph-based Representations in Patt. Rec. (GbR). Volume 5534 of LNCS. (2009) 11–20
5. Dupé, F.X., Brun, L.: Tree covering within a graph kernel framework for shape classification. In: ICIAP. Volume 5716 of LNCS. (2009)
6. Shervashidze, N., Vishwanathan, S.V.N., Petri, T.H., Mehlhorn, K., Borgwardt, K.M.: Efficient graphlet kernels for large graph comparison. In: Proc. of the 12th Int. Conf. on Artificial Intelligence and Statistics. (2009) 488–495
7. Mahé, P., Vert, J.P.: Graph kernels based on tree patterns for molecules. Machine Learning **75** (2009) 3–35
8. Gaüzère, B., Brun, L., Villemin, D.: Two new graph kernels and applications to chemoinformatics. In: 8th Workshop on Graph-Based Representations in Pattern Recognition. Volume 6658 of LNCS. (2011) 112–122
9. Bougleux, S., Dupé, F.X., Brun, L., Gaüzère, B., Mokhtari, M.: Shape similarity based on combinatorial maps and a tree pattern kernel. In: 21st Int. Conf. on Pattern Recognition. (2012)
10. Brun, L., Mokhtari, M., Domenger, J.P.: Incremental modifications on segmented image defined by discrete maps. Journal of Visual Communication and Image Representation **14** (2003) 251–290
11. Cori, R.: Computation of the automorphism group of a topological graph embedding. Technical Report I-8612, Université Bordeaux 1, France (1985)
12. Bougleux, S., Brun, L.: Symmetry group of 2d combinatorial maps. Technical report, GREYC, France (2012)
13. Neuhaus, M., Bunke, H.: Edit-distance based kernel for structural pattern classification. Pattern Recognition **39** (2006) 1852–1863
14. Sharvit, D., Chan, J., Tek, H., Kimia, B.B.: Symmetry-based indexing of image databases. Journal of Visual Communication and Image Representation **9**(4) (1998) 366–380
15. Gdalyahu, Y., D.Weinshall: Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. IEEE Trans. on PAMI **21**(2) (1999) 1312–1328
16. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. on PAMI **24**(4) (2002) 509–522
17. Ling, H., Jacobs, D.W.: Shape classification using the inner-distance. IEEE Trans. on PAMI **29**(2) (2007) 286–299