# Edition within a graph kernel framework for shape recognition

François-Xavier Dupé* and Luc Brun

GREYC UMR CNRS 6072,
ENSICAEN-Université de Caen Basse-Normandie,
14050 Caen France,
{francois-xavier.dupe,luc.brun}@greyc.ensicaen.fr

**Abstract.** A large family of shape comparison methods is based on a medial axis transform combined with an encoding of the skeleton by a graph. Despite many qualities this encoding of shapes suffers from the non continuity of the medial axis transform. In this paper, we propose to integrate robustness against structural noise inside a graph kernel. This robustness is based on a selection of the paths according to their relevance and on path editions. This kernel is positive semi-definite and several experiments prove the efficiency of our approach compared to alternative kernels.

**Key words:** Shape, Skeleton, Support Vector Machine, Graph Kernel

## 1 Introduction

The skeleton is a key feature within the shape recognition framework [1–3]. Indeed, this representation holds many properties: it is a thin set, homotopic to the shape and invariant under Euclidean transformations. Moreover, any shape can be reconstructed from the maximal circles of the skeleton points.

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 the skeletons. Several encoding systems have been proposed: Di Ruberto [4] proposes a direct translation of the skeleton to the graph using many attributes. Siddiqi [5] proposes a graph which characterises both structural properties of a skeleton and the positive, negative or null slopes of the radius of the maximal circles along a branch. Finally this last encoding has been improved and extended to 3D by Leymarie and Kimia [6].

The recognition of shapes using graph comparisons may be tackled using various methods. A first family of methods is based on the graph edit distance which is defined as the minimal number of operations required to transform the graph encoding the first shape into the graph encoding the second one [2, 3].

* This work is performed in close collaboration with the laboratory Cycéron and is supported by the CNRS and the région Basse-Normandie.

Another method, introduced by Pelillo [1], transforms graphs into trees and then models the tree matching problem as a maximal clique problem within a specific association graph. A last method proposed by Bai and Latecki [7] compares paths between end-node (node with only one neighbor) after a matching task on the end-nodes. Contrary to previously mentioned approaches this last method can deal with loops and may thus characterize holed shapes.

All the above methods perform in the graph space which almost contains no mathematical structure. This forbids many common mathematical tools like the mean graph of a set which has to be replaced by its median. A solution consists to project graphs into a richer space. Graph kernels provide such an embedding: by using appropriate kernels, graphs can be mapped either explicitly or implicitly into a vector space whose dot product corresponds to the kernel function.

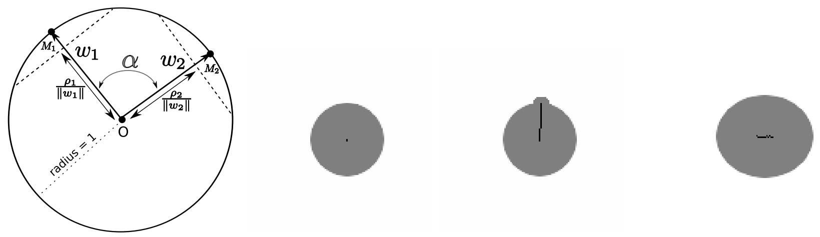
Most famous graph kernels are the random walk kernel, the marginalized graph kernel and the geometric kernel [8]. A last family of kernels is based on the notion of bag of paths [9]. These methods describe each graph by a subset of its paths, the similarity between two graphs being deduced from the similarities between their paths. Path similarity is based on a comparison between the edges and nodes attributes of both paths.

However, skeletonization is not a continuous process and small perturbations of a shape may produce ligatures and spurious branches. Graph kernels may in this case lead to inaccurate comparisons. Neuhaus and Bunke have proposed several kernels (e.g. [10]) based on the graph edit distance in order to reduce the influence of graph perturbations. However the graph edit distance does not usually fulfills all the properties of a metric and the design of a definite positive kernel from such a distance is not straightforward. Our approach is slightly different. Indeed, instead of considering a direct edit distance between graphs, our kernel is based on a rewriting process applied on the bag of paths of two graphs. The path rewriting follows the same basic idea than the string edit distance but provides a definite positive kernel between paths.

This paper follows a first contribution [11] where we introduced the notion of path rewriting within the graph kernel framework. It is structured as follows: first, we recall how to construct a bag of path kernel [9, 11] (Section 2). Then, we propose a graph structure (Section 3) which encodes both the structure of the skeleton and its major characteristics. This graph contains a sufficient amount of information for shape reconstruction. We then extend the edition operations (Section 4) by taking into account all the attributes and by controlling the effect of the edition on them. Finally, we present experiments (Section 5) in order to highlight the benefit of the edition process.

## 2 Bag of path kernel

Let us consider a graph $G = (V, E)$ where $V$ denotes the set of vertices and $E \subset V \times V$ the set of edges. A bag of paths $P$ associated to $G$ is defined as a set of paths of $G$ whose cardinality is denoted by $|P|$. Let us denote by $K_{path}$ a generic path kernel. Given two graphs $G_1$ and $G_2$ and two paths $h_1 \in P_1$




(a) Sets on the unit sphere (b) Original (c) Spurious branch (d) Ligature

**Fig. 1.** (a) Separating two sets using one-class SVM. The symbols ($w_1$,$\rho_1$) and ($w_2$,$\rho_2$) denote the parameters of the two hyperplanes which are represented by dashed lines. Influence of small perturbation on the skeleton (in black) ((b),(c) and (d)).

and $h_2 \in P_2$ of respectively $G_1$ and $G_2$, $K_{path}(h_1, h_2)$ may be interpreted as a measure of similarity between $h_1$ and $h_2$. The aim of a bag of path kernel consists to aggregate all these local measures between pairs of paths into a global similarity measure between the two graphs. Such a kernel differs from random walk kernels where all the paths of the two graphs are compared.

### 2.1 Change Detection Kernel

Desobry [12] proposed a general approach for the comparison of two sets which has straightforward applications in the design of a bag of path kernel (bags are sets). The two bags are modelled as the observation of two sets of random variables in a feature space.

Desobry proposes to estimate a distance between the two distributions without explicitly building the pdf of the two sets. The considered feature space is based on a normalised kernel ($K(h, h') = K_{path}(h, h')/\sqrt{(K_{path}(h, h)K_{path}(h', h'))}$). Using such a kernel we have $\|h\|_K^2 = K(h, h) = 1$ for any path. The image in the feature space of our set of paths lies thus on an hypersphere of radius 1 centered at the origin (Fig. 1(a)). Using the one-class $\nu$-SVM, we associate a set of paths to a region on this sphere. This region corresponds to the density support estimate of the set of paths' unknown pdf.

Once the two density supports are estimated, the one-class SVM yields $w_1$ (resp. $w_2$), the mean vector, and $\rho_1$ (resp. $\rho_2$), the ordinate at the origin, for the first bag (resp. the second bag). In order to compare the two mean vectors $w_1$ and $w_2$, we define the following distance function:

$$d_{mean}(w_1, w_2) = \arccos\left(\frac{w_1^t K_{1,2} w_2}{\|w_1\| \, \|w_2\|}\right), \tag{1}$$

where $K_{1,2}(i, j) = K(h_i, h_j),\ h_i \in P_1,\ h_j \in P_2$ and $w_1^t K_{1,2} w_2$ is the scalar product between $w_1$ and $w_2$. This distance corresponds to the angle $\alpha$ between the two mean vectors $w_1$ and $w_2$ of each region (Fig. 1(a)).Then we define the kernel between two bags of path $P_1$ and $P_2$ as 1) the product of a Gaussian RBF kernel associated to $d_{mean}(w_1, w_2)$ and 2) a Gaussian RBF kernel associated to the difference between the two coordinates at the origin ($\rho_1$ and $\rho_2$):

$$K_{change}(P_1, P_2) = \exp\left(\frac{-d^2_{mean}(w_1, w_2)}{2\sigma^2_{mean}}\right) \exp\left(\frac{-(\rho_1 - \rho_2)^2}{2\sigma^2_{origin}}\right). \quad (2)$$

Finally, we define the kernel between two graphs $G_1$, $G_2$ as the kernel between their two bags of path: $K_{change}(G_1, G_2) = K_{change}(P_1, P_2)$.

The distance between the mean vectors is a metric based on a normalized scalar product combined with arccos which is bijective on $[0, 1]$. However, the relationship between the couple $(w, \rho)$ and the bag of path being not bijective, the final kernel between bags is only semi positive-definite [13]. Though, in all our experiments run so far the Gram matrices associated to the bags of paths were positive-definite.

### 2.2 Path kernel

The above bag of path kernel is based on a generic path kernel $K_{path}$. A kernel between two paths $h_1 = (v_1, \ldots, v_n)$ and $h' = (v'_1, \ldots, v'_p)$ is classically [14] built by considering each path as a sequence of nodes and a sequence of edges. This kernel denoted $K_{classic}$ is defined as 0 if both paths have not the same size and as follows otherwise:

$$K_{classic}(h, h') = K_v(\varphi(v_1), \varphi(v'_1)) \prod_{i=2}^{|h|} K_e(\psi(e_{v_{i-1}v_i}), \psi(e_{v'_{i-1}v'_i})) K_v(\varphi(v_i), \varphi(v'_i)), \quad (3)$$

where $\varphi(v)$ and $\psi(e)$ denote respectively the vectors of features associated to the node $v$ and the edge $e$. The terms $K_v$ and $K_e$ denote two kernels for respectively node's and edge's features. For the sake of flexibility and simplicity, we use Gaussian RBF kernels based on the distance between the attributes defined in section 3.2.

## 3 Skeleton-based graph

### 3.1 Graph representations

Medial-axis based skeleton are built upon a distance function whose evolution along the skeleton is generally modeled as a continuous function. This function presents important changes of slope mainly located at the transitions between two parts of the shape. Based on this remark Siddiqi and Kimia distinguish three kind of branches within the shock graph construction scheme [2]: branches with positive, null or negative slopes. Nodes corresponding to these slope transitions are inserted within the graph. Such nodes may thus have a degree 2. Finally, edges are directed using the slope sign information.

Compared to the shock graph representation, we do not use oriented edges since small positive or negative values of the slope may change the orientation of an edge and thus alter the graph representation. On the other hand our set of nodes corresponds to junction points and to any point encoding an important change of slope of the radius function. Such a significant change may encode a change from a positive to a negative slope but also an important change of slope with a same sign (Fig. 2(a)). Encoding these changes improves the detection of the different parts of the shape. The main difficulty remains the detection of the slope changes due to the discrete nature of the data. The slopes are obtained using regression methods based on first order splines [15]. These methods are robust to discrete noise and first order splines lead to a continuous representation of the data. Moreover, such methods intrinsically select the most significant slopes using a stochastic criterion. Nodes encoding slope transitions are thus located at the junctions (or knot) between first order splines.

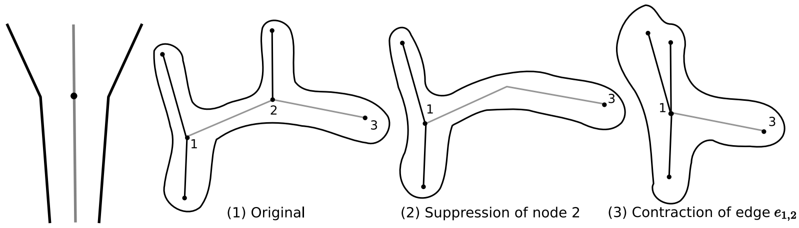
### 3.2 Attributes

The graph associated to a shape only provides information about its structural properties. Additional geometrical properties of the shape may be encoded using node and edge attributes. From a structural point of view, a node represents a particular point inside the shape skeleton and an edge a branch. However, a branch also represents the set of points of the shape which are closer to the branch than any other branch. This set of points is defined as the *influence zone* of the branch and can be computed using SKIZ transforms [16].

Descriptors computed from the influence zone are called *local*, whilst the ones computed from the whole shape are called *global*. In [3] Goh introduces this notion and points out that an equilibrium between local and global descriptors is crucial for the efficiency of a shape matching algorithm. Indeed local descriptors provide a robustness against occlusions, while global ones provide a robustness against noise.

We have thus selected a set of attributes which provides an equilibrium between local and global features. Torsello in [17] proposes as edge attribute an approximation of the perimeter of the boundary which contributes to the formation of the edge, normalized by the approximated perimeter of the whole shape. Suard proposes [9] as node attribute the distance between the node position and the gravity center of the shape divided by the square of the shape area. These two attributes correspond to our global descriptors.

Goh proposes several local descriptors [3] for edges based on the evolution of the radius of the maximal circle along a branch. For each point $(x(t), y(t))$ of a branch, $t \in [0, 1]$, we consider the radius $R(t)$ of its maximal circle. In order to normalize the data, the radius is divided by the square root of the area of the influence zone of the branch. We also introduce $\alpha(t)$, the angle formed by the tangent vector at $(x(t), y(t))$ and the $x$-axis. Then we consider $(a_k)_{k\in\mathbb{N}}$ and $(b_k)_{k\in\mathbb{N}}$ the coefficients of two regression polynomials that fit respectively $R(t)$ and $\alpha(t)$ in the least square sense. If both polynomials are of sufficient orders, the skeleton can be reconstructed from the graph and so the shape (Section 1). Following Goh [3], our two local descriptors are defined by: $\sum_k a_k/k$ and $\sum_k b_k/k$.




(a) A change of slope

(b) Edition effect on the shape (path in gray)

**Fig. 2.** Slope detection (a) and edition of paths (b).

The distance associated to each attribute is defined as the absolute value of the difference between the values $a$ and $b$ of the attribute: $d(a, b) = |a - b|$. As the attributes are normalized , the distances are invariant to change of scale and rotation. Such distances are used to define the Gaussian RBF kernels ($\exp\left(\frac{-d^2(.,.)}{2\sigma^2}\right)$) used to design $K_{path}$ (Section 2.2).

## 4 Hierarchical Kernels

The biggest issue with skeleton-based graph representation is the non-negligible effect of small perturbations on the shape [2]: Fig. 1 shows two deformations of the skeleton of a circle (Fig. 1(b)) one induced by a small bump (Fig. 1(c)) and one by an elongation (Fig. 1(d)). On complex shapes, severe modifications of the graphs may occur and lead to inaccurate comparisons.

From a structural point of view, perturbations like bumps (Fig. 1(c)) create new nodes and edges. In contrast, the principal effect of an elongation (Fig. 1(d)) is either the addition of an edge inside the graph or the extension of an existing edge. So shape noise mainly induces two effects on paths: addition of nodes (Fig. 1(c)) and addition of edges (Fig. 1(d)). This leads to the two editions operations: *node suppression* and *edge contraction*. Note that, as the compared structure are paths, the relevance of these operations should be evaluated according to the path under study.

### 4.1 Elementary operations on path

The *node suppression* operation removes a node from the path and all the graph structures that are connected to this path by this node. Within the path, the two edges incident to the nodes are then merged. This operation corresponds to the removal of a part of the shape: for example, if we remove the node 2 in Fig. 2(b1), a new shape similar to Fig. 2(b2) is obtained.

The *edge contraction* operation contracts an edge and merges its two extremity nodes. This results in a contraction of the shape: for example, if we contract the edge $e_{1,2}$ of the shape in Fig. 2(b1) then the new shape will be similar to Fig. 2(b3).

Since each operation is interpreted as a shape transformation, the global descriptors must be updated. From this point of view our method may be considered as a combination of the methods of Sebastian [2] and Goh [3] who respectively use local descriptors with edit operations and both local and global descriptors but without edit operations.

### 4.2 Edition cost

In order to select the appropriate operation, an edition cost is associated to each operation. Let us consider an attribute *weight* associated to each edge of the graph which encodes the relevance of its associated branch. We suppose that this attribute is additive: the weight of two consecutive edges along a path is the sum of both weights.

Note that, we consider the maximal spanning tree $T$ of the graph $G$. As skeletonization is an homotopic transform, a shape with no hole yields $T = G$. Let us consider a path $h = (v_1, \dots, v_n)$ within $T$. Now, an edition cost is assigned to both operations within $h$:

- Let us consider a node $v_i$, $i \in \{2, \dots, n-1\}$ of the path $h$ (extremity nodes are not considered). The cost of *the node suppression* operation on $v_i$ must reflect two of its properties: 1) the importance of the sub-trees of $T$ connected to the path by $v_i$ and 2) the importance of the slope changes (Section 3.1) between the two branches respectively encoded by the edges $e_{v_{i-1}v_i}$ and $e_{v_i v_{i+1}}$.
  The relevance of a sub-tree is represented by its total weight: for each neighbor $v$ of $v_i$, $v \notin h$, we compute the weight $W(v)$ defined as the addition of the weight of the tree rooted on $v$ in $T \setminus \{e_{v_i v}\}$ and the weight of $e_{v_i v}$. This tree is unique since $T$ is a tree. The weight of the node $v_i$ is then defined as the sum of weights $W(v)$ for all neighbors $v$ of $v_i$ ($v \notin h$) and is denoted by $\omega(v_i)$.
  We encode the relevance of a slope change by the angle $\beta(v_i)$ formed by the slope vectors associated to $e_{v_{i-1}v_i}$ and $e_{v_i v_{i+1}}$. An high value of $\beta(v_i)$ encodes a severe change of slopes and conversely. Since slopes are approximated using first-order polynomials (section 3.1), the angle $\beta(v_i)$ is given by $\beta(v_i) = \arccos\left(\frac{1+a_1 * a'_1}{\sqrt{1+a_1^2}\sqrt{1+a'^2_1}}\right)$ where $a_1$ and $a'_1$ are the first order coefficients of the regression polynomials.
  Finally the edition cost of the suppression of a node is defined by $(1-\gamma)\omega(v_i) + \gamma\beta(v_i)/\pi$, where $\gamma$ is a tuning variable.
- The cost of *the edge contraction* operation encodes the importance of the edge inside the graph $T$, this is the purpose of the weight. So, the edition cost of contracting an edge is defined as its weight.

Concerning *weight* any additive measure encoding the relevance of a skeleton's branch may be used. We choose to use the normalized perimeter as computed by Torsello [17], because of its resistance to noise on the shape boundary.

### 4.3 Edition path kernel

Let us denote by $\kappa$ the function which applies the cheapest operation on a path and $D$ the maximal number of reductions. The successive applications of the function $\kappa$ associate to each path $h$ a sequence of reduced paths $(h, \kappa(h), \ldots, \kappa^D(h))$. Each $\kappa^k(h)$ is associated to a cost: $cost_k(h)$ defined as the sum of the costs of the $k$ operations yielding $\kappa^k(h)$ from $h$. Using $K_{classic}$ for the path comparison, we define the kernel $K_{edit}$ as a sum of kernels between reduced paths. Given two paths $h$ and $h'$, the kernel $K_{edit}(h, h')$ is defined as:

$$K_{edit}(h,h') = \frac{1}{D+1}\sum_{k=0}^{D}\sum_{l=0}^{D} \exp\left(-\frac{\text{cost}_k(h)+\text{cost}_l(h')}{2\sigma_{cost}^2}\right) K_{classic}(\kappa^k(h), \kappa^l(h')), \quad (4)$$

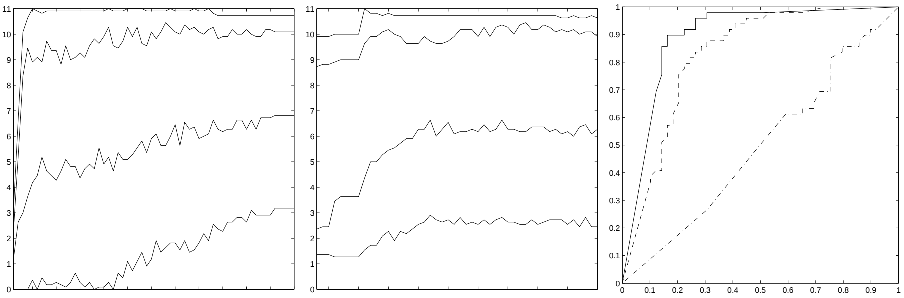
where $\sigma_{cost}$ is a tuning variable. This kernel is composed of two parts: a scalar product of the edition costs in a particular space and a path kernel. For a small value of $\sigma_{cost}$ the behavior of the kernel will be close to $K_{classic}$ as only low editions cost will contribute to $K_{edit}(h, h')$. Conversely, for a high value every editions will contribute to $K_{edit}(h, h')$ with an approximately equal importance.

The kernel $K_{classic}$ is a tensor product kernel based on positive-definite kernels (Section 2.2), so it is positive-definite. The kernel over edition costs is constructed from a scalar product and is thus positive-definite. These two last kernels form a tensor product kernel. Finally $K_{edit}$ is proportional (by a factor $D+1$) to a $R$-convolution kernel [18, Lemma 1], thus this kernel is positive-definite.

## 5 Experiments

For the following experiments, we defined the importance of a path as the sum of the weights of its edges. For each graph, we first consider all its paths composed of at most 7 nodes and we sort them according to their importance using a descending order. The bag of paths is then constructed using the first 5 percent of the sorted paths. For all the experiments, the tuning variable of the deformation cost $\gamma$ (Section 4.2) is set to 0.5.

The first experiment consists in an indexation of the shapes using the distances induced by the kernels, i.e. $d(G, G') = k(G, G) + k(G', G') - 2k(G, G')$ where $k$ is a graph kernel. The different $\sigma$ of the attributes RBF kernels involved in $K_{classic}$ (Section 3.2) are fixed as follows: $\sigma_{perimeter} = \sigma_{radius} = \sigma_{orientation} = 0.1$ and $\sigma_{gravity\ center} = 0.2$. Note that $K_{classic}$ constitutes the basis of all the kernels defined below. The parameters of $K_{change}$ are set to: $\sigma_{mean} = 1.0$, $\sigma_{origin} = 20$ and $\nu = 0.9$. The maximal number of editions is fixed to 6. Let us consider the class tool from the LEMS database [19] of 99 shapes with 11 elements per class. Two kind of robustness are considered: robustness against ligatures and perturbations and robustness against erroneous slope nodes. Ligatured skeletons of the shapes are created by varying the threshold parameter $\zeta$ of the skeletonization algorithm [17], high values lead to ligatured skeletons while low value tend to remove relevant branches. Skeletons with erroneous slope nodes are created by varying the parameter of our slope detection algorithm. This detection is based on the BIC criterion which uses the standard error of the noise



(a) Slope changes (b) Ligatures (c) ROC curves

**Fig. 3.** Resistance to spurious slope changes (a) and spurious branches(b). For (a) and (b) the kernels are from top to bottom: $K_{change,edit2}$(—), $K_{change,edit1}$(⁞⁞⁞), random walk kernel(- -), and $K_{change,classic}$(···). (c) ROC curves for the classification of dogs and cats using: $K_{change,edit}$ (—), random walk kernel (- -) and $K_{change,classic}$ (···).

$\sigma_{BIC}$. A small value of $\sigma_{BIC}$ makes the criterion sensitive to small changes of slopes and gives many slope nodes, while high value makes the criterion insensitive to slope changes. Four kernels are compared: random walk kernel [8], $K_{change}$ with $K_{classic}$ (denoted as $K_{change,classic}$) and 2 kernels using $K_{change}$ with $K_{edit}$ ( with $\sigma_{cost} = 0.1$ for $K_{change,edit1}$ and $\sigma_{cost} = 0.2$ for $K_{change,edit2}$ ). Using the distances induced by the kernels, shapes are sorted in ascending order according to their distance to the perturbed tool. Fig. 3(a) shows the mean number of tools inside the first 11 sorted shapes for an increasing value of $\sigma_{BIC}$. Fig. 3(b) shows the same number but for a decreasing threshold value $\zeta$. The two edition kernels show a good resistance to perturbations and ligatures as they get almost all the tools for each query. Their performances slightly decrease when shapes become strongly distorted. The kernel $K_{change,classic}$ gives the worst results as the reduction of the bag of paths leads to paths of different lengths which cannot be compared with $K_{classic}$ (Section 2.2). The random walk kernel is robust against slight perturbations of the shapes but cannot deal with severe distortion.

In the second experiment, we strain kernels by separating 49 dogs from 49 cats using a $\nu$-SVM. The three considered kernels are $K_{change,classic}$, $K_{change,edit}$ (with $\sigma_{cost} = 0.5$) and random walk. The different $\sigma$ of the attributes RBF kernels (Section 3.2) are fixed as follows: $\sigma_{perimeter} = \sigma_{radius} = \sigma_{orientation} = 0.1$ and $\sigma_{gravity\ center} = 0.5$. The parameters of $K_{change}$ are set to: $\sigma_{mean} = 5.0$, $\sigma_{origin} = 20$ and $\nu = 0.9$. We compute the ROC curves produced by kernels using a 10-fold cross-validation. Fig 3(c) presents the three ROC curves. The random walk kernel gives correct results, whilst the $K_{change,classic}$ kernel confirms its poor performance. The $K_{change,edit}$ kernel shows the best performances and a behaviour similar to the random walk kernel. Furthermore, on our computer a Core Duo 2 at 2GHz, the computational burden of the 98x98 Gram matrix is of approximately 23 minutes for $K_{change,edit}$ and of 2.5 hours for the random walk kernel.

## 6 Conclusion

We have defined in this paper a positive-definite kernel for shape classification which is robust to perturbations. Our bag of path contains the more important paths of a shape below a given length in order to only capture the main information about the shape. Only the $K_{edit}$ kernel provides enough flexibility for path comparison and gives better results then the classical random walk kernel. In a near future, we would like to improve the selection of paths. An extension of the edition process on graphs is also planned.

## 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. Sebastian, T., Klein, P., Kimia, B.: Recognition of shapes by editing their shock graphs. IEEE Trans. on PAMI **26**(5) (2004) 550–571
3. Goh, W.B.: Strategies for shape matching using skeletons. Computer Vision and Image Understanding **110** (2008) 326–345
4. Ruberto, C.D.: Recognition of shapes by attributed skeletal graphs. Pattern Recognition **37**(1) (2004) 21–31
5. Siddiqi, K., Shokoufandeh, A., Dickinson, S.J., Zucker, S.W.: Shock graphs and shape matching. Int. J. Comput. Vision **35**(1) (1999) 13–32
6. Leymarie, F.F., Kimia, B.B.: The shock scaffold for representing 3d shape. In: In Proc. of 4th International Workshop on Visual Form (IWVF4, Springer-Verlag (2001) 216–229
7. Bai, X., Latecki, J.: Path Similarity Skeleton Graph Matching. IEEE PAMI **30**(7) (2008)
8. Vishwanathan, S., Borgwardt, K.M., Kondor, I.R., Schraudolph, N.N.: Graph kernels. Journal of Machine Learning Research **9** (2008) 1–37
9. Suard, F., Rakotomamonjy, A., Bensrhair, A.: Mining shock graphs with kernels. Technical report, LITIS (2006) http://hal.archives-ouvertes.fr/hal-00121988/en/.
10. Neuhaus, M., Bunke, H.: Edit-distance based kernel for structural pattern classification. Pattern Recognition **39** (2006) 1852–1863
11. Dupé, F.X., Brun, L.: Hierarchical bag of paths for kernel based shape classification. In: SSPR 2008. (2008) 227–236
12. Desobry, F., Davy, M., Doncarli, C.: An online kernel change detection algorithm. IEEE Transaction on Signal Processing **53**(8) (August 2005) 2961–2974
13. Berg, C., Christensen, J.P.R., Ressel, P.: Harmonic Analysis on Semigroups. Springer-Verlag (1984)
14. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernel between labeled graphs. In: In Proc. of the Twentieth International conference on machine Learning. (2003)
15. DiMatteo, I., Genovese, C., Kass, R.: Bayesian curve fitting with free-knot splines. Biometrika **88** (2001) 1055–1071
16. Meyer, F.: Topographic distance and watershed lines. Signal Proc. **38**(1) (1994)
17. Torsello, A., Hancock, E.R.: A skeletal measure of 2d shape similarity. CVIU **95** (2004) 1–29
18. Haussler, D.: Convolution kernels on discrete structures. Technical report, Department of Computer Science, University of California at Santa Cruz (1999)
19. LEMS: shapes databases. http://www.lems.brown.edu/vision/software/