



**HAL**  
open science

## Object Classification Based on Graph Kernels

Amal Mahboubi, Luc Brun, François-Xavier Dupé

► **To cite this version:**

Amal Mahboubi, Luc Brun, François-Xavier Dupé. Object Classification Based on Graph Kernels. HPCS 2010, Jun 2010, Caen, France. pp.385-389. hal-00954566

**HAL Id: hal-00954566**

**<https://hal.science/hal-00954566>**

Submitted on 3 Apr 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Object Classification Based On Graph Kernels

Amal Mahboubi, Luc Brun and François-Xavier Dupé  
GREYC UMR CNRS 6072  
amal.mahboubi@unicaen.fr  
luc.brun,fdupe@greyc.ensicaen.fr

## ABSTRACT

*Automatic object recognition plays a central role in numerous applications, such as image retrieval and robot navigation. A now classical strategy consists to compute a bag of features within a sliding window and to compare this bag with precomputed models. One main drawback of this approach is the use of an unstructured bag of features which do not allow to take into account relationships which may be defined on structured objects. Graphs are natural data structures to model such relationships with nodes representing features and edges encoding relationships between them. However, usual distances between graphs such as the graph edit distance do not satisfy all the properties of a metric and classifiers defined on these distances are mainly restricted to the  $K$  nearest neighbors method. This article describes an image object classification method based on a definite positive graph kernel inducing a metric between graphs. This kernel may thus be combined with numerous classification algorithms.*

**KEYWORDS:** Object classification, Graph kernels.

## 1. INTRODUCTION

One now classical [2] strategy to recognize objects consists to compute a bag of features from a sliding window and to compare this bag to a model using scoring functions. Features of the bag may be deduced from boundary segments or SIFT points [2] defined at multiple scales.

However, such methods usually resume the image information to histograms hereby loosing any structural property of the object. Recent algorithms [10] have proposed to overcome this drawback by encoding the objects by graphs using a heuristic graph matching algorithm to classify objects. Indeed, graphs provide a natural framework to capture the structure of objects. Classical distances between graphs are either based on the computation of the maximum common sub-graph [6] or on graph edit distances [7] both problems being related and both nearly  $\mathcal{NP}$  complete. The computation of a distance be-

tween graphs using this approach requires thus either the use of an appropriate heuristic to compute efficiently an approximation of the distance on general graphs or a simplification of the graphs in order to apply more robust algorithms. Both solutions have been used. For example, Bunke [7] proposed an efficient heuristic to approximate the edit distance between two graphs. On the other hand Pelillo [6] transforms shock graphs into shock tree in order to apply an efficient maximal clique detection algorithm.

Neglecting the complexity issue one additional drawback of graph matching or graph edit distances is the lack of properties of the resulting similarity or distance measures. Indeed, such measures do not usually provide a metric. As a consequence, usual graph distances cannot be readily combined with usual classification or regression algorithms coming from the statistical machine learning framework.

Definite positive graph kernels may be understood as a measure of similarity between graphs. Using Mercer's theorem such kernels are associated to a mapping function of graphs onto an Hilbert space such that the kernel between two graphs corresponds to the scalar product between their mapping within the Hilbert space. Using the famous "Kernel trick" many usual tools defined within the statistical machine learning framework may be used by using only kernels between graphs hereby avoiding the explicit computation of the mapping function which is usually unknown.

Several graph kernels have been proposed by Neuhauss and Bunke [8] based on the edit distance. However such kernels are usually not definite positive. Dupé and Brun [1] have proposed several kernels based on the notion of bag of paths. Outside the pattern recognition framework, many graph kernels have been proposed in the bioinformatic community [9]. Most of these kernels are based on kernels between bags of paths.

This paper aims to define a kernel between graphs deduced from points of interest detected in two image to be compared. Within such a framework the main feature of both graphs is the point of interest together with

its neighborhood. The notion of path is thus less relevant and we propose to associate to each graph a bag of vertices encoding the points of interest. The structure of the graph is captured by adding to each vertex the ordered sequence of its neighbors. Our graph construction scheme together with our new graph kernel are presented in Section . The proposed approach is validated through experiments in section .

## 2. GRAPH KERNEL CONSTRUCTION SCHEME

### 2.1. Graph Construction

As in the usual bag of word model, the first step of our method consists to extract a set of features from an image. The Scale Invariant Feature Transform (SIFT) [5] provides a list of key points robust to changes in image scale, noise, illumination and local geometric distortion. Each key point is expressed by its  $x$  and  $y$  coordinates, the scale, the orientation and 128 numbers (in the range 0-255) representing the descriptor. Figure 1 illustrates a key point detection by the SIFT method on a face image. This unstructured set of points is then organized into a graph in order to take into account spatial relationships between key points. Given an image, we build a labeled graph  $G = (V, E, \mu, \mathcal{F}, w)$  where  $V, E$  and  $\mathcal{F}$  respectively denote the set of vertices, edge and vertex's features while  $\mu$  and  $w$  respectively correspond to a vertex labelling function from  $V$  to  $\mathcal{F}$  and a weighing function from  $V$  to  $\mathbb{R}^+$ . Each key point is associated to a vertex  $v \in V$ , while its features  $\mu(v) \in \mathcal{F}$  are stored as a label of this vertex. The weighing function  $w$  encodes the relevance of each vertex within the graph. Within our framework, this weigh is defined for each vertex as the scale of its associated key point. This choice corresponds to the usual assumption that relevant and stable key points should be detected at large scales. Note however, that any relevant measure of the relevance of key points may be used as weigh. The set  $E$  of edges is defined from the  $k$  mutual nearest neighbors based on the  $x, y$  coordinates of key points: One edge  $(v, v')$  belongs to  $E$  if  $v$  belongs to the  $k$  nearest neighbors of  $v'$  while  $v'$  belongs to the  $k$  nearest neighbors of  $v$ . The degree of each vertex is thus bounded by  $k$ .

In order to take into account the local arrangement of vertices incident to a given vertex we explicitly encode for each vertex  $v$  the sequence of its neighbors encountered when turning counter-clockwise around  $v$ . The neighborhood  $\mathcal{N}(v)$  of vertex  $v$  is thus defined as an *ordered set* (a sequence)  $(v_1, \dots, v_n)$  of vertices. The first vertex of this sequence is arbitrary chosen as the upper right vertex (minimum of the sum of both coordinates).



Figure 1. SIFT Feature Points

### 2.2. Graph Kernel Construction

One classical way to define a kernel between graphs is to consider finite [1] or infinite [4] bags of paths. Such kernels allow to consider global or semi global properties of graphs using walks traversing large part of them. However, using SIFT key point detector the information is mainly local and is thus essentially captured by graph's vertices and their immediate neighborhoods.

We thus define our kernel between graphs as a kernel between bags of oriented neighborhoods. The design of such a kernel implies the definition of a basic kernel between oriented neighborhoods. Considering the oriented neighborhoods  $\mathcal{N}(u)$  and  $\mathcal{N}(v)$  of two vertices  $u$  and  $v$ , with  $l_u = |\mathcal{N}(u)|$  and  $l_v = |\mathcal{N}(v)|$ , one straightforward solution consists to use a kernel between vectored data hereby considering each sequence as a vector:

$$K_{seq}(u, v) = \begin{cases} 0 & \text{if } l_u \neq l_v \\ \prod_{i=1}^{l_u} K_g(u_i, v_i) & \text{otherwise} \end{cases} \quad (1)$$

where  $K_g(u, v)$  is a RBF kernel between the features of input vertices:  $K_g(u, v) = e^{-\frac{d(\mu(u), \mu(v))}{\sigma}}$ ,  $\sigma$  denoting a tuning parameter while  $d(\cdot, \cdot)$  is the Euclidean distance, between the features  $\mu(u)$  and  $\mu(v)$  of vertices  $u$  and  $v$ . Sequences  $(u_i)_{i \in \{1, \dots, l_u\}}$  and  $(v_i)_{i \in \{1, \dots, l_v\}}$  denote respectively the oriented neighborhoods  $\mathcal{N}(u)$  and  $\mathcal{N}(v)$ .

The kernel between oriented neighborhood defined by equation 1 compares the features of two oriented neighborhood of *equal length*. However due to acquisition noise or small changes between two images some SIFT points may be added or removed between two similar images. Such addition or removal of SIFT points may modify the neighbourhood of vertices belonging to the  $k$  nearest neighbors of the added or removed points. Using equation 1 such vertices become incomparable with their original version in the initial graph.

In order to overcome this drawback we introduce a rewriting rule on oriented neighborhood, which removes the vertex of lower weight of an oriented neighborhood. Given one vertex  $v$ , the rewriting of its oriented neighborhood  $\mathcal{N}(v)$ , denoted  $\kappa(v)$  is defined as:

$$\kappa(v) = (v_1, \dots, \hat{v}_i, \dots, v_{l_v}) \quad (2)$$

where  $\hat{v}_i = \operatorname{argmin}_{j \in \{1, \dots, l_v\}} w(v_j)$  is the neighbor of  $v$  with lowest weight. Note that, using a more rigorous notation, the rewriting of  $\mathcal{N}(v)$  should be denoted  $\kappa(\mathcal{N}(v))$ . We nonetheless use the simplified notation  $\kappa(v)$  in order to avoid cumbersome notations.

The function  $\kappa$  may be iterated thus leading to a sequence of successively simplified oriented neighborhoods:  $(\kappa^1(v), \kappa^2(v), \dots, \kappa^D(v))$  where  $D$  is fixed to 50% of the maximal vertex's degree  $k$  (Section ). Each oriented neighborhood of such a sequence is obtained from the previous oriented neighborhood by removing one vertex. the dissimilarity with the initial neighborhood  $\mathcal{N}(v)$  increases thus along the sequence. In order to quantify such a dissimilarity we associate a cumulative weigh to each element of this sequence of rewritings. The cumulative weigh of the  $i^{\text{th}}$  rewriting of the initial oriented neighborhood  $\mathcal{N}(v)$ , denoted  $Cw(\kappa^i(v))$  corresponds to the sum of the weigh of the vertices removed to obtain  $\kappa^i(v)$  from  $\mathcal{N}(v)$  and is formally defined as:

$$\begin{aligned} Cw(v) &= 0 \\ Cw(\kappa^i(v)) &= w(v_i) + Cw(\kappa^{i-1}(v)) \end{aligned} \quad (3)$$

where  $v_i$  is the vertex removed between  $\kappa^{i-1}(v)$  and  $\kappa^i(v)$ .

Based on these rewriting rules, we extend the initial vertex's kernel defined by equation 1 to a convolution kernel [3] between two sets of rewrote oriented neighborhoods:

$$\begin{aligned} K_{rewriting}(u, v) &= \sum_{i=1}^D \sum_{j=1}^D \\ W(\kappa^i(u), \kappa^j(v)) &* K_{seq}(\kappa^i(u), \kappa^j(v)) \end{aligned} \quad (4)$$

where the kernel  $W$  penalizes costly rewritings corresponding to the removal of important key points. This kernel is defined as a kernel between the cumulative weighs of both rewritings:

$$W(\kappa^i(u), \kappa^j(v)) = e^{-\frac{Cw(\kappa^i(u)) + Cw(\kappa^j(v))}{\sigma'}} \quad (5)$$

where  $\sigma'$  is a tuning variable.

Our final kernel between vertices combines  $K_{rewriting}$  with one kernel between the features of the two central vertices defining both oriented neighborhoods:

$$K(u, v) = K_g(u, v) K_{rewriting}(u, v) \quad (6)$$

Our final kernel between both graphs corresponds to a kernel between both bags of oriented neighborhoods using equation 6. One usual design of such a kernel [1] would consist to perform a sum of kernels between all couples of oriented neighborhoods belonging to both bags. However, such a solution do not take into account the relevance of vertices within both graphs. In order to take such a relevance into account we introduce a function  $\varphi$  from  $V$  to  $\mathbb{R}+$  which increases with

respect to vertex's weigh and is formally defined as:

$$\varphi(v) = e^{-\frac{1}{\sigma'(1+w(v))}}.$$

Our final graph kernel is thus defined as a weighed sum of kernels between oriented neighborhoods:

$$K_{graph}(G_1, G_2) = \sum_{u \in V_1} \sum_{v \in V_2} \varphi(u) \varphi(v) K(u, v) \quad (7)$$

### 3. EXPERIMENTS

#### 3.1. Dataset

We choose to test our kernel against a database of 41 faces composed of 11 females and 30 males, extracted from the FERET face database<sup>1</sup>. Each of the 41 subject of the database is associated with 5 images: one regular facial expression, an alternative facial expression, two half left and half right rotations of  $-67.5$  degrees and one right profile with a rotation of  $+90$  degrees. The image is fixed to  $256 \times 384$  pixels. Some images from the database are displayed in Figure 2.

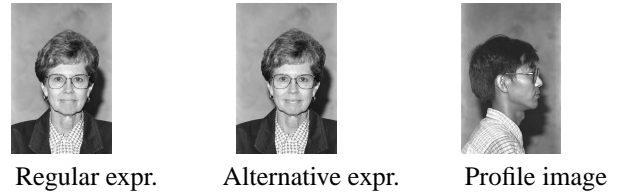


Figure 2. FERET Database

#### 3.2. Experimental Setup

Our first experiment illustrates the importance of the rewriting step within our kernel computation scheme. To this end, we randomly select 5 images among our database and corrupt them by an additive Gaussian noise with increasing standard deviation  $\sigma$ . The similarity between each original image and its corrupted version is measured both with our kernel  $K_{graph}$  (equation 7) and by a kernel without any rewritings denoted  $K_{graphSeq}$ . This last kernel is defined from equation 7 by setting  $K(u, v) = K_g(u, v) \cdot K_{seq}(u, v)$  (equation 1). In order to compare both kernels we normalize them using the following formula [3]:

$$\tilde{K}(G, G') = \frac{K(G, G')}{\sqrt{K(G, G)K(G', G')}} \quad (8)$$

where  $\tilde{K}$  represent the normalized version of a kernel  $K$ . Using such a normalisation step, both kernels  $\tilde{K}_{graph}$  and  $\tilde{K}_{graphSeq}$  vary between 0 and 1. Fig. 3 represents the mean values of  $\tilde{K}_{graph}$  and  $\tilde{K}_{graphSeq}$  for increasing noise levels. Vertical bars on each curve represent

<sup>1</sup>[http://www.itl.nist.gov/iad/humanid/feret/feret\\_master.html](http://www.itl.nist.gov/iad/humanid/feret/feret_master.html)

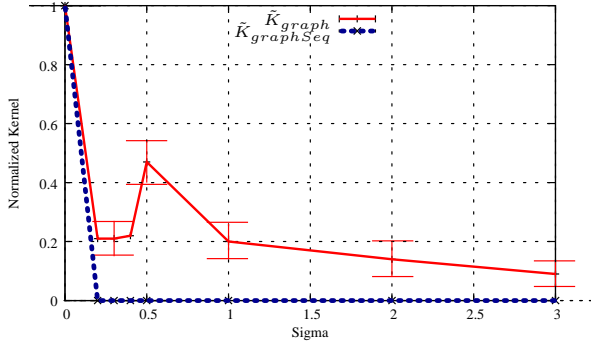


Figure 3. Normalized Kernels  $\tilde{K}_{graph}$  and  $\tilde{K}_{graphSeq}$ .

the standard deviations of kernel values computed on the 5 images for each noise levels. As shown by Fig. 3, the curve representing  $\tilde{K}_{graphSeq}$  quickly drops to 0. This results means that even for low noise level, any SIFT point in a corrupted image has in its neighborhood at least one added or removed point compared to the same SIFT point in the original image. Using equation 1, the similarity between the oriented neighborhoods of both SIFT points is equal to 0. On the other hand, the curve associated to  $\tilde{K}_{graph}$  quickly drops to 0.2 for a value of  $\sigma$  equal to 1, and then softly decreases as the noise level increases hereby showing the resistance to noise bring by our rewriting step.

Our second experiment investigates the relationships between the number of rewritings  $D$  and the maximal size of the neighborhood  $k$ . To this end, we randomly selected 4 images belonging to different classes within the FERET database. We then iteratively removed each of the 4 selected images from the FERET database and computed its 4 nearest neighbors among the 201 remaining images. For a given choice of  $k$  and  $D$ , if our kernel correctly captures the class of each image, the 4 closest neighbors of each query should correspond to the remaining images of its class. We thus computed for each of the 4 query images the number of images belonging to its class among its 4 closest neighbors within the database. This score computed on each query image is sum up in order to obtain a global result for each couple  $(k, D)$  which varies between 0 and 16. Table 1 summarizes this experiment for different choice of  $k$  and  $D$ . The maximal value of each line is highlight by bold fonts. One can note that for each value of  $k$  the score passes through a single maximum. This phenomenon may be interpreted as follows: For low value of  $D$  the number of rewritings do not compensate small image perturbations due to noise and view change. Conversely, for high, value of  $D$ , our rewriting step may remove important points with slightly similar weigh in both neighborhoods. Such removals may lead to different neighborhoods with a low similarity according to  $K_{seq}$  (equation 1). The weighted sum of this low values of  $K_{seq}$  leads to a low overall value of  $K_{rewriting}$  (equation 4). The optimal value of

	1	2	3	4	5		1	2	3	4	5
1	3	1	1	0	0	1	2	2	0	0	1
2	0	5	0	0	0	2	0	4	1	0	0
3	0	0	5	0	0	3	0	0	5	0	0
4	0	1	2	2	0	4	0	1	3	1	0
5	0	0	2	0	3	5	0	0	2	0	3

(a)  $K=1$

(b)  $K=4$

Figure 4. Confusion Matrices Using The KNN Classification Algorithm, 5 Classes Of The FERET Database And Two Values Of  $K$ .

$D$  is equal to 50% of  $k$  for  $k = 10$  and  $k = 15$ . This optimal value remains equal to 5 for  $k = 7$  and 13 hence representing respectively 71% and 38% of  $k$ . Therefore, as expected, the heuristic setting  $D = \frac{K}{2}$  made in Section does not provide the optimal value of  $D$  in all cases. This choice provides however results close to the optimal solution and may be used as an alternative to an explicit estimation of the optimal value of  $D$  using a cross validation step.

Table 1. Classification Scores Obtained On The FERET Database For Various Values Of  $k$  And  $D$ .

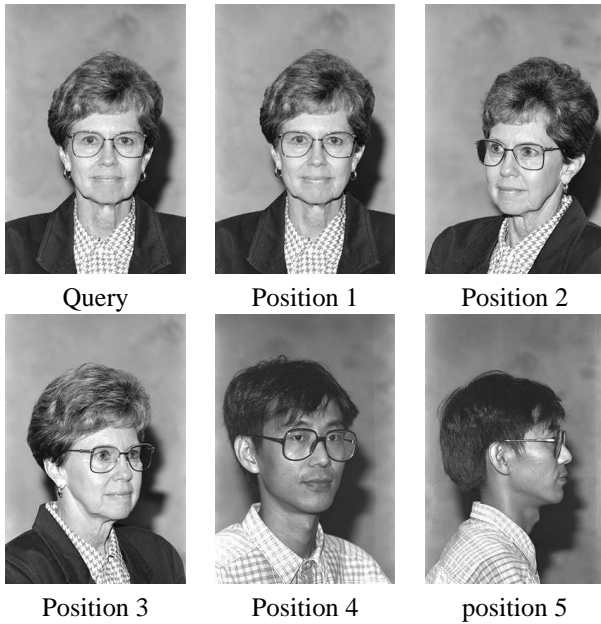
	D=1	D=3	D=4	D=5	D=6	D=7	D=12
k=7	3	5	5	<b>7</b>	6	-	-
k=10	3	5	5	<b>6</b>	5	6	-
k=13	3	5	5	<b>7</b>	5	6	6
k=15	6	5	5	5	5	<b>7</b>	6

### 3.3. Object Classification

Given a query image of an object of interest, object classification consists to retrieve its class from a set of objects already classified called a training set. Our graph kernel has been tested within this framework on 5 classes of the FERET database using the classical KNN classification algorithm. Figure 4 represents the confusion matrices obtained on these five classes for  $K = 1$  and  $K = 4$ . As illustrated by Figure 4 both matrices have only few out diagonal values, the best results being obtained for  $K = 1$ . An example of the 5 nearest neighbors of a query image is represented in Fig. 5. Three of the four images belonging to the same class as the query and encoding the same person in different views are correctly retrieved. The missing image corresponds to the profile view which induces a very different graph.

## 4. CONCLUSION

In this study, a graph kernel method for object retrieval has been proposed with an application to face recognition. We defined a graph kernel based on bags of oriented



**Figure 5. Ranked Five Closest Images From the Query**

neighborhoods. Our first results are promising. Further work will be devoted to further improvements of kernels between oriented neighborhoods.

## REFERENCES

- [1] F.X Dupé and L. Brun, “Shape classification using a flexible graph kernel”, In X. Jiang, editor, *Proceedings of CAIP 2009*. LNCS, September 2009.
- [2] H. Harzallah, F. Jurie and C. Schmid, “Combining efficient object localization and image classification”, In *Proceedings of ICCV, 2009*.
- [3] D. Haussler, “Convolution Kernels on Discrete Structures”, Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.
- [4] H. Kashima, K. Tsuda and A. Inokuchi, “Marginalized kernel Between Labeled Graphs”, In *Proc. of the Twentieth International conference on machine Learning*, 2003.
- [5] D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, *International Journal for Computer Vision*, 2:91-110, 2004.
- [6] M. Pelillo, K. Siddiqi and S. Zucker, “Machine hierarchical structures using association graphs”, *PAMI*, 21(11):1105-1120, Nov. 1999.
- [7] K. Riesen, M. Neuhaus and H. Bunke, “Bipartite graph matching for computing the edit distance of graphs”, In *Gbr’2007*, volume 4538 of *LNCS*, pages 1-12, IAPR-TC15, 2007.
- [8] K. Riesen, M. Neuhaus and H. Bunke, “Graph Embedding in Vector Spaces by Means of Prototype Selection”, In *Gbr’2007*, LNCS, pages 383-393, IAPR-TC15, 2007.
- [9] J.P. VERT, “Classification of Biological Sequences with Kernel Methods”, In *Grammatical Inference: Algorithms and Applications*, pages 7-18, 2006.
- [10] S. Xia, E.R Hancock, “Pairwise Similarity Propagation based Graph Clustering for Scalable Object Indexing and Retrieval”, In *Gbr’2009*, volume 5534 of *LNCS*, pages 184-194, Venice 2009. IAPR-TC15.