# Image retrieval with graph kernel on regions

Justine Lebrun, Sylvie Philipp-Foliguet and Philippe-Henri Gosselin
*ETIS, CNRS, ENSEA, Univ Cergy-Pontoise, F-95000 Cergy-Pontoise*
*{justine.lebrun,philipp,gosselin}@ensea.fr*

## Abstract

*In the framework of the interactive search in image databases, we are interested in similarity measures able to learn during the search and usable in real-time. Images are represented by adjacency graphs of regions. In order to compare attributed graphs, we employ kernels on graphs built on sets of paths. In this paper, we introduce a fast kernel function whose similarity is based on several matches. We also introduce new features for edges in the graph. Experiments on a specific database having objects with heterogeneous backgrounds show the performance of our object retrieval technique.*

## 1 Introduction

The problem of graph comparison is a topic which has been widely studied in the literature for several decades. Many algorithms build isomorphisms between two graphs, which means that graphs have the same structure, the same number of nodes and the same number of edges. The problem of comparing graphs with unlabelled nodes and edges is a NP-hard problem. When the nodes and edges are labelled, the same probleme is much simpler. In our approach, in order to rank and classify a set of graphs, both nodes and edges are valued. The complexity of matching such graphs stands, then, in between the complexities of the aforemention problems.

Recent approaches propose to consider graphs as sets of paths [4]. As we are interested in matching only a part of the image (the object and not its background), this approach seems able to measure a similarity between sets of regions with their layout. Support Vector Machines (SVM) are state-of-the-art large margin classifiers which have demonstrated remarkable performances in image retrieval, when associated with adequate kernel functions. We propose to use kernels on graphs based on kernels on paths to compute the similarity between images. The classification or the search in a database from one or several examples is then performed using interactive learning with a SVM classifier and techniques of active learning for the selection of the images to be annotated by the user.

Our contribution in this paper is threefold. First we present the general framework unifying search trees and kernels on graphs to perform inexact graph matching. Secondly we introduce a fast kernel function whose similarity is based on several matches. We also introduce new features to describe the spatial relationship between two regions. Thirdly we evaluate our propositions in the context of interactive object retrieval.

## 2 Framework

### 2.1 Graph matching with kernels

Each image of the base is represented by a graph$G = (V, E)$, where $V$ is a set of vertices, and $E \subseteq V \times V$ is a set of edges. For example, if an image is segmented into regions, such a graph is built by representing each region by a vertex $v \in V$, and each adjacency between two regions by an edge $e = (v_1, v_2) \in V \times V$. A path $h$ is a set of vertices $(v_0, ..., v_n)$ linked by edges of $E$.

In many similarity measures $S(G, G')$ between two graphs $G = (V, E)$ and $G' = (V', E')$, the idea is to find the best matches between vertices and edges. For example, Sorlin [6] proposes a similarity measure which is the average value of the best similarities between vertices and edges. FReBIR [5] computes the value of the best match between a query path and any path of the other image. However this similarity measure does not possess the usual properties of a metric measure such as symmetry or triangular inequality. They are thus hardly usable by some of the powerful tools used for classification or for *"browsing"* for example.

On the contrary, some similarity measures use kernels respecting Mercer properties, allowing an easy use by the search engines. In this case, the similarity measure is a dot product $S(G, G') = K(G, G') =$

$\langle \Phi(G), \Phi(G') \rangle$ in a Hilbert space $\mathcal{H}$, with $\Phi$ a function which maps a graph to a vector of $\mathcal{H}$.

Some approaches try to explicitly build the kernel function through $\Phi(G)$. For example, Jurie[3] proposes to compute a dictionary of the graph vertex prototypes (here vertices represent points of interest) the most frequent in the database, and then to project the vertices on this dictionary in order to build histograms – these histograms will be vectors $\Phi(G)$ in the Hilbert space. Grauman[2] implicitly includes spatial constraints, through a pyramid approach. The drawback of these methods based on prototypes is their weak ability to generalize outside the training database.

The other way we are interested in, is to perform an implicit calculation of the image similarities in a Hilbert space via a kernel function. An interesting property of the Mercer kernels is their closure towards addition or product. We will use these properties to build our kernels.

Kashima[4] proposed to compare two graphs by comparing all possible paths of both graphs. The kernel functions then concern sets (or bags) of paths and thus involve similarities between vertices and between edges. A general model for the computation of a kernel function on graphs is defined by considering the sets of all paths in each graph, and then by computing the average value of all similarities between paths of $G$ and of $G'$ of same length. If $|h|$ is the length of path $h$, i.e. its edge number, this kernel function is expressed as, with $p_G(h)$ the probability of finding path $h$ in graph $G$ :

$$\mathbf{K}(G, G') = \sum_{\substack{h \in G \\ h' \in G' \\ |h| = |h'|}} K_C(h, h') p_G(h) p_{G'}(h') \qquad (1)$$

Kernel function $K_C(h, h')$ measures the similarity between two paths ($h = v_0 ... v_n, h' = v'_0 ... v'_n$). The minor kernels which occur in this equation are the kernel on vertices $k_V$ and the kernel on edges $k_V$. For $k_V$, we use a Gaussian kernel, which takes values between 0 and 1. Kernel $k_E$ allows to take into account the similarity between edges [7].

Other functions can increase the discrimination while reducing the computation time. The following one has been used in FReBIR [5] :

$$\mathbf{K}(G, G') = \max_{h \in G} \max_{\substack{h' \in G' \\ |h| = |h'|}} K_C(h, h') \qquad (2)$$

This function is not a kernel function strictly speaking, but in practice it respects the Mercer conditions on the databases used in the experiments. The main drawback of this function is that the similarity between two graphs is based on only one match between two paths, while with Kashima kernel (Eq. (1)) , the similarity between two graphs is based on all path similarities.

### 2.2 Optimization algorithms

Once the measure of similarity between two graphs defined, the problem of finding the matching which maximizes this similarity is very complex, especially if the search is not limited to isomorphisms between graphs. Kernel on paths $K_C$ are not limited to paths of same length since paths can include loops, i.e several time the same vertex.

There is often a compromise to do between optimal solution and computation time. Algorithms by ant colony or by taboo research [6] find optimal solutions but they are too slow for the real-time use we consider. Another very common approach uses search trees (properly speaking, rooted trees). Each node of this tree represents a couple of vertices $(v, v')$ candidates for the matching. The tree is built from an empty root node, by developing each node with candidate couples. The candidate nodes are couples $(v, v')$ compatible with the nodes already built in the (oriented) path from the root to the current node. The advantage of this representation by rooted tree is that the similarity of a path is computed during the oriented path building. In the case of a similarity function which uses a $max$, the algorithm "branch and bound" finds the optimal solution without exploring all possible solutions. The most promising solution is firstly obtained, it gives a lower bound for the similarity $\mathbf{K}(G, G')$. Then the other branches of the tree are built only if they are likely to increase the similarity value. This most promising solution is obtained by exploring the oriented path made of the nodes with the largest similarity values.

## 3 Propositions

### 3.1 Proposed graph matching kernel

Kashima kernel takes into account the similarities between all paths of both graphs. If Kashima kernel is interesting for labeled vertices, in return it needs to compute all path similarities. Thus it is slower to compute. For example, if among 100 possible matches, there are 3 matches with a high similarity value $a$ and 97 with a small similarity value $b$, then the total similarity equals $3a + 97b$. The 3 strong matching are not sufficient towards the 97 small matching. On the other hand FReBIR kernel computes the similarity of the best couple of paths$(h, h')$. Between these two extreme behaviors, we propose a new kernel :

$$
\begin{aligned}
\mathbf{K}(G,G') &= \sum_{v\in G} \max_{\substack{h\in s_G(v)\\ h'\in s_{G'}(m_{G'}(v))}} K_C(h,h') \\
&+ \sum_{v'\in G'} \max_{\substack{h'\in s_{G'}(v')\\ h\in s_G(m_G(v'))}} K_C(h,h')
\end{aligned} \quad (3)
$$

$$
\text{with} \begin{cases} h \in s_G(v) \Leftrightarrow v \text{ is the first vertex of } h \in G \\ m_G(u) = \operatorname*{argmax}_{w\in G}(k_V(w,u)) \end{cases}
$$

The computation of the similarity between two graphs can be performed using several "branch and bound" runs (one by vertex). It is not as fast as the FReBIR kernel (Eq. (2)) , but it is much more faster than Kashima kernel (Eq. (1)) , while still being based on several matches of paths.

### 3.2 Topological similarity between sets of regions

Adjacency graphs of regions carry two types of information : region information and topological information. If regions are commonly described by color and texture distributions, spatial relationships between regions have been less studied. We propose a description composed of 4 features to build a kernel on edges $k_E(e_i, e'_i)$. An edge is an oriented link between two regions. It is described by the 4 features : A(above), B(below), L(left) and R(right).

For two adjacent regions $R_i$ and $R_j$, we consider the set $F_{ij}$ of pixel couples $(p_i, p_j) \in R_i \times R_j$ neighbors in 4-connectivity. Then we define the following features :

$$
T_{ij}^{\mathcal{R}_t} = \frac{|\{(p_i,p_j) \in F_{ij}\ ,\ p_j \mathcal{R}_t p_i\}|}{|F_{ij}|}
$$

with spacial relations $p_i\mathcal{R}_1p_j \Leftrightarrow p_i$ is above $p_j$, $p_i\mathcal{R}_2p_j \Leftrightarrow p_i$ is below $p_j$, $p_i\mathcal{R}_3p_j \Leftrightarrow p_i$ is left of $p_j$, $p_i\mathcal{R}_4p_j \Leftrightarrow p_i$ is right of $p_j$.

These four features are then packed into a vector $(T_{ij}^{\mathcal{R}_1}T_{ij}^{\mathcal{R}_2}T_{ij}^{\mathcal{R}_3}T_{ij}^{\mathcal{R}_4})$ which represents the edge between vertices $v_i$ and $v_j$. We have tested several classical kernel functions $k_E$ with these attributes and our results showed that a Gaussian kernel with $\chi 2$ distance is the best choice.

## 4 Experiments

### 4.1 Comparison of graph kernels

We compared the three graph kernels we presented in this paper (Eq. (1), (2) and (3)) in the context of interactive object retrieval. Each retrieval session is initialized with one image containing the searched object. Then, the user labels images containing or not the searched object, and the system updates the ranking of the database according to these new labels. It is a weakly supervised learning context, since the user does not give the position of the objects in the image.


**Figure 1. Synthetic database.**

The experiments were performed on a synthetic database of 600 images (*cf.* Fig. 1). It is made of 50 objects of 12 views each and put on a random background. The objects come from the Columbia database, but the background is replaced by an image of landscape issued from the ANN database. The graphs issued from the segmentation of the images have between 3 and 15 vertices. Regions are represented by histograms of colors and textures. Evaluation is measured with the Mean Average Precision metric [1]. In all cases, we used the following kernel on paths:

$$
K_C(h,h') = k_V(v_0,v'_0) + \sum_{i=1}^{|h|} k_E(e_i,e'_i)k_V(v_i,v'_i)
$$

The results presented on Fig. 2 show that for the three kernels performances increase with path length, especially for the FReBIR kernel (Eq. (2)) . It comes from the fact that the similarity is computed on the best match of two paths of both graphs. If we compare now the best MAP for each of the three kernels on graphs, the best result is obtained using Kashima kernel (Eq. (1)) .

However, when considering the very high computational complexity of Kashima kernel, the proposed kernel is the most interesting since it gives better results than FReBIR while having a comparable computational complexity.

### 4.2 Topological similarity

We evaluated the influence of the features we introduced in this paper. Experiments were carried out on the Columbia database, which is composed of 7,200 images

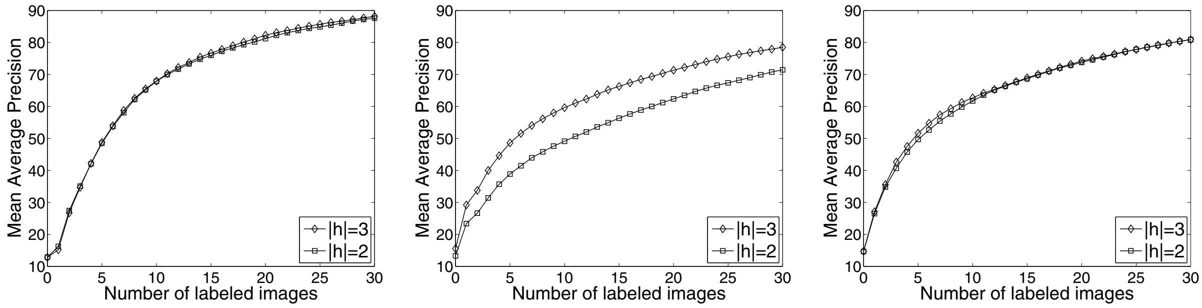


**Figure 2. Results (%) on Columbia database for path length from 1 to 3. Left : Kashima kernel; Middle : FReBIR kernel ; Right : proposed kernel.**

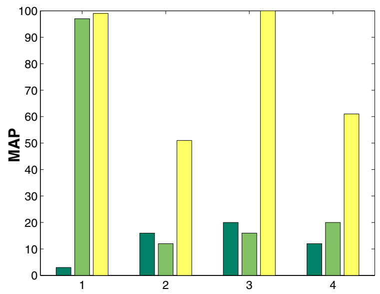


**Figure 3. Results(%) for four objects from the Columbia database. Left/darkest bars : only using edges ; Middle bars : only using vertices ; Right/lightest bars : edges and vertices.**

of 100 objects from different viewpoints. Each image is automatically segmented into 4 to 15 regions by image (the black background is sometimes split into several regions). When we use the only information on vertices, we use a kernel on edge $k_E(e, e') = 1$ for all $e, e' \in E \times E'$. When we use the only information on edges, we use a kernel on vertices $k_V(v, v') = 1$ for all $v, v' \in V \times V'$ and we assume that all vertices are connected. The evaluation protocol is the same than the one used in the previous section.

Fig. 3 shows the results using only edges, only vertices and both $k_E$ and $k_V$, for four representative objects. In all cases, the combination of edges and vertices provide significantly better results.

## 5 Conclusion

We have shown in this paper that inexact graph matching can be achieved by using kernel theory. We introduced a fast and efficient graph kernel, but also new features on edge which, combined with features on vertices, give significantly better results. The combination of region information and topological information of the layout of regions within image improves the retrieval of images, specially for a object retrieval task. When the semantic is carried only by several regions of the image the new kernel on graphs we introduced gives results almost as good as Kashima kernel but in a time compatible with a real-time task.

## References


[1] TREC Video Retrieval Evaluation Campain. http://www-nlpir.nist.gov/projects/trecvid/.

[2] K. Grauman and T. Darell. The pyramid match kernel : Discriminative classification with sets of image features. In *IEEE International Conference on Computer Vision (ICCV)*, Beijing, China, October 2005.

[3] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision and Pattern Recognition*, 2005.

[4] H. Kashima and Y. Tsuboi. Kernel-based discriminative learning algorithms for labeling sequences, trees and graphs. In *International Conference on Machine Learning (ICML)*, Banff, Alberta, Canada, 2004.

[5] S. Philipp-Foliguet and J. Gony. FReBIR : Fuzzy regions-based image retrieval. In *Information Processing and Management of Uncertainty (IPMU)*, Paris, France, July 2006.

[6] S. Sorlin, O. Sammoud, C. Solnon, and J.-M. Jolion. Mesurer la similarité de graphes. In *Extraction de COnnaissance à partir d'Images (ECOI'06), Atelier de Extraction et Gestion de Connaissances (EGC'06)*, pages 21–30, Juanary 2006.

[7] F. Suard, V. Guigue, A. Rakotomamonjy, and A. Bensrhair. Pedestrian detection using stereo-vision and graph kernels. In *Intelligent Vehicles Symposium*, pages 267–272, Las Vegas, Nevada, June 2005.