



**HAL**  
open science

## **Kite Recognition by means of Graph Matching**

Kamel Madi, Hamida Seba, Hamamache Kheddouci, Charles-Edmont Bichot,  
Olivier Barge, Christine Chataigner, Rémy Crassard, Emmanuelle Régagnon,  
Emmanuelle Vila

► **To cite this version:**

Kamel Madi, Hamida Seba, Hamamache Kheddouci, Charles-Edmont Bichot, Olivier Barge, et al..  
Kite Recognition by means of Graph Matching. Graph-based Representations in Pattern Recognition  
(GbR2015), May 2015, Beijing, China. hal-01829332

**HAL Id: hal-01829332**

**<https://hal.science/hal-01829332>**

Submitted on 4 Jul 2018

**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.

# Kite Recognition by means of Graph Matching

Kamel Madi<sup>1</sup>, Hamida Seba<sup>1</sup>, Hamamache Kheddouci<sup>1</sup>,  
Charles-Edmont Bichot<sup>2</sup>, Olivier Barge<sup>3</sup>, Christine Chataigner<sup>3</sup>,  
Remy Crassard<sup>3</sup>, Emmanuelle Reganon<sup>3</sup>, and Emmanuelle Vila<sup>3</sup>

<sup>1</sup>Université de Lyon, CNRS

Université Lyon 1, LIRIS, UMR5205, F-69622, France

<sup>2</sup>Ecole Centrale de Lyon, LIRIS, UMR5205, F-69134, France

<sup>3</sup>CNRS, UMR 5133 Archéorient, Lyon, France

hamida.seba@univ-lyon1.fr

**Abstract.** Kites are remnants of long stone walls that outline the shape of a child's kite. But the kites are huge, their big size makes them often clearly visible on high-resolution satellite images. Identified at first in the Near East, their area of distribution is getting larger and larger. This wide distribution gives new dimensions in the interpretation of these structures. Consequently, a large scale recognition of kites will help archeologists to understand the functionality of these enigmatic constructions. In this paper, we investigate how the satellite imagery can be exploited in this purpose using a graph representation of the kites. We propose a similarity measure and a kite identification process that can highlights the preservation state of the kites. We also construct from real images a benchmark of kite graphs that can be used by other researchers.

**Keywords:** Kite recognition, graph matching, edit distance, satellite image

## 1 Background an Motivation

Kites are archaeological structures made of long walls converging on an enclosure that have small cells in its periphery (see Figure 1(a)). Kites have been discovered in the Middle East since the 1920s. Despite several studies, the issues related to their age and function remain without satisfactory answers [4, 8]. The kites are thus an underestimated phenomenon. Establishing the duration of their utilization, outlining their use and functioning, and trying to identify the population who are behind these constructions are the challenges that would highlight the significance of this unknown phenomenon. However, these issues cannot be seriously addressed without an almost exhaustive inventory of these structures.

In this paper, we investigate how the satellite imagery can be exploited in this purpose using a graph representation of the kites. Kites are naturally representable with graphs. A graph  $G = (V, E)$  is a representation tool composed of a set of vertices  $V$  and a set of edges  $E$  with the cardinalities  $|V(G)| = n$  and  $|E(G)| = m$  where  $n$  is called the order of the graph and  $m$  its size. The set of

edges  $E$  is a subset of  $V \times V$  such that  $(u, v) \in E$  means that vertices  $u$  and  $v$  are connected. Usually, a finite number of labels are associated with vertices and edges. Graphs are commonly used as representation tools in many applications of structural pattern recognition and classification. For these kind of applications, graph comparison is a fundamental issue. Graph matching and more generally graph comparison is a powerful tool for structural pattern recognition [1, 5, 19]. Its has been used in the recognition process of architectonic/urbanistic elements in aerial/satellite images such as building or Gable-Roof detection [2, 20]. Graph comparison solutions are classified into two wide categories: exact approaches and fault-tolerant approaches. Exact approaches, such as those that test for graph isomorphism or sub-graph isomorphism [13, 18], refer to the methods that look for an exact mapping between the vertices and edges of a query graph and the vertices and edges of a target graph. Fault-tolerant graph comparison computes a distance between the compared graphs. This distance measures how similar are the graphs and allows to deal with the errors that are introduced by the processes needed to model objects by graphs. Several similarity measures are proposed in the literature using different approaches: graph kernels, graph embedding, maximum common subgraph, graph invariants, etc. We refer the reader to [1, 19] for more exhaustive surveys. We focus here on two main approaches that we use in the rest of the paper: graph edit distance (GED) and graph invariants. GED is one of the most powerful fault-tolerant graph matching tool. GED is based on a kind of graph transformations called edit operations. An edit operation is either an insertion, a suppression or a re-labeling of a vertex or an edge in the graph. A cost function associates a cost to each edit operation. The edit distance between two graphs is defined by the minimum costing sequence of edit operations that are necessary to transform one graph into the other [17]. This sequence is called an optimal edit path. However, computing the exact value of the edit distance between two graphs is NP-Hard for general graphs and induces a high computational complexity. This motivated several heuristics that approaches the exact value of GED in polynomial time using different methods such as dynamic programming and probability (see [6] for a detailed survey). In this context, [15] proposes a polynomial-time framework based on a fast bipartite assignment procedure mapping nodes and their local structures of one graph to nodes and their local structures of another graph. Similar approaches are also presented in [14]. Graph invariants have been efficiently used to solve the graph comparison problem in general and the graph isomorphism problem in particular. They are used for example in Nauty [13] which is one of the most efficient algorithm for graph and subgraph isomorphism testing. A vertex invariant, for example, is a number  $i(v)$  assigned to a vertex  $v$  such that if there is an isomorphism that maps  $v$  to  $v'$  then  $i(v) = i(v')$ . Examples of invariants are the degree of a vertex, the number of cliques of size  $k$  that contain the vertex, the number of vertices at a given distance from the vertex, etc. Graph invariants are also the basis of graph probing [12] where a distance between two graphs is defined as the norm of their probes. Each graph probe is a vector of graph

invariants. A generalization of this concept is also used in [21] to compare biological data.

In this paper, we propose a graph comparison approach that deals with the following issues related to kite recognition:

1. to distinguish between two isomorphic graphs that have different geometry by using angles between two adjacent edges as well as distances between connected vertices in the similarity measure. In fact, almost all existing graph similarity measures compare the structures of graphs in terms of vertices, edges and their labels but they do not consider the geometric forms of these graphs. This must be very important when the graphs represent objects with specific forms as for kite graphs presented later in the paper,
2. to deal with an important number of graphs: the challenge is to be able to process an important number of images in almost real-time.
3. to deal with disconnected graphs : depending on the state of preservation of the kites, the kite graphs may have several connected components.

The proposed approach combines the speed of graph invariant computation with the resilience of GED, in a multilevel framework consisting of: (1) a fast computable global similarity measure based on graph invariants computed on the compared graphs. This similarity aims to rapidly discard the graphs that can not be kites and avoid unnecessary and more costly comparisons; (2) a more accurate local similarity measure based on the geometric form and the structured features extracted from the graphs. This similarity uses GED to deal with the state of preservation of the kites; and (3) a reconstruction process that verifies if the different connected components of the graph constitute a kite.

We also provide a benchmark of kite graphs that we constructed from real images. This benchmark can be used to evaluate other graph matching methods.

The rest of the paper is organized as follows: in Section 2 we describe how the kites are represented by graphs and the proposed similarity measure. Section 3 gives the results of our experiments. Section 4 concludes the paper and describe our future work.

## 2 A Graph Based Approach for Kite Recognition

In this section, we present our contribution which is twofold:

1. The construction of a valuable dataset of kite graphs extracted from real images. This dataset comes enriching the existing graph databases used by researchers to evaluate graph matching and similarity measures.
2. A graph-based similarity measure for kite recognition.

We will use the notations summarized in Table 1 in the remaining of the paper. We consider simple undirected graphs with labels on both vertices and edges. Furthermore, we use the angles between two consecutive edges to record the geometric form of the graphs. We first describe the process of extracting kite graphs from images. Then, we give the detailed algorithms of our similarity measure.

Table 1: Notation

Symbol	Description
$G(V, E)$	undirected labeled graph with $V$ its vertex set and $E$ its edge set. Both vertices and edges are labeled.
$V(G)$	vertex set of graph $G$ .
$E(G)$	edge set of graph $G$ .
$deg(v)$	degree of vertex $v$ .
$\Delta(G)$	the greatest vertex degree in graph $G$ .
$\ell(e)$	the label of edge $e$ .
$\mathcal{A}(G)$	the greatest angle in graph $G$ .
$\mathcal{L}(G)$	the greatest edge label in graph $G$ .
$\ S\ $	size of the set $S$ .

## 2.1 From Kite images to kite Graphs: construction of the dataset

The structure of a Kite can be naturally represented by a graph. On satellite images, Kites appears as flat surfaces delimited by a set of lines. To convert kites' images into attributed graphs, the first stage is to extract the kites from the images by edge detection. Edge (segment or line) detection in images is an intensively studied topic in image analysis [3,9]. Besides, several recent methods such as [7, 11, 16] give good result on satellite images. The main difficulty with such methods is to find the adequate settings to obtain an acceptable segment detection for a specific application. For the kites, we investigated several solutions with various settings and the LSD algorithm [7] gave us the most satisfactory set of segments (see Figure 1(b)). However, we added a refinement step to delete isolated segments and to merge adjacent ones (see Figure 1(c)). Then, a skeleton is generated by thinning the segments (see Figure 1(d)). The graph is constructed from the skeleton by representing lines by edges and ending points of lines by vertices (Figure1(e)). Each vertex is labeled with a two-dimensional attribute giving its position and an  $n$ -dimensional attribute containing the angles between every pair of consecutive incident edges. According to the state of preservation of the kite, a graph obtained by this process can have a single connected component (i.e., the kite is totally preserved) or it can be composed by two or several connected components (i.e., some parts of the kite have been destroyed).

We executed our algorithm on an important number of images containing kites with different state of preservation. We classified the obtained graphs into three preservation levels:

1. State I: the kite is entire and well preserved. The kite graph may be disconnected but the disconnections are neither frequent nor important.
2. State II: the kite graph is very disconnected. Some parts of the kite are not present.
3. State III: The graph is not a kite. These graphs are obtained by executing the algorithm on images that do not contain kites.

Figure 2 depicts some examples in each case. We have actually 200 graphs in each dataset with the characteristics summarized in Table 2.

With the help of the archeologists, we selected from the graphs in State-I dataset, the most preserved kites and modified them manually to obtain prototype kite graphs. Also, to be able to deal with disconnected kite graphs without

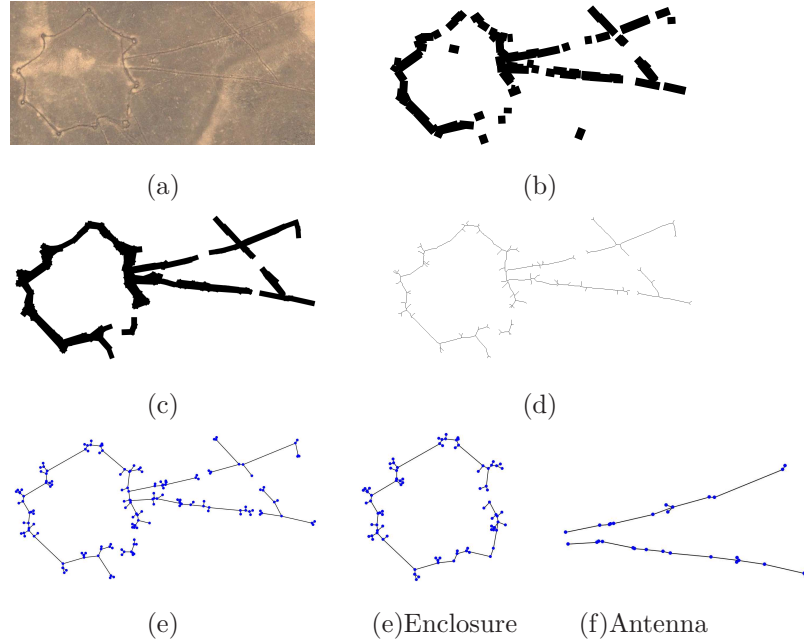


Fig. 1: Illustration of kite detection

Table 2: Graph Dataset Characteristics

$avg|V|$ : average number of vertices.  $avg|E|$ : average number of edges.  $max|V|$ : maximum number of vertices.  $max|E|$ : maximum number of edges.  $avgAng$ : average value of the angles.  $maxAng$ : maximum angle value.

$avg$ $ V $	$max$ $ V $	$avg$ $ E $	$max$ $ E $	$avg$ $Ang$	$max$ $Ang$
13	622	14	754	124	180

adding significant computing costs, we constructed a prototype graph for each kite component, namely : antenna and enclosure. Figures 1(e) and 1(f) give respectively an example of a kite enclosure and a kite antenna.

## 2.2 A Multi-level Graph Comparison Approach for Kite Recognition

The proposed kite recognition solution consists of three levels: a global similarity measure, a local similarity measure and a reconstruction process. The global similarity measure aims to rapidly discard objects that cannot be kites based on the size and the form of the objects compared to the prototypes. It computes graph invariants. The global similarity between two graphs  $G_1$  and  $G_2$  is given by:

$$\begin{aligned}
 G\_Sim(G_1, G_2) = & w_1 * d_{Convex}(G_1, G_2) + w_2 * d_{Edges}(G_1, G_2) \\
 & + w_3 * d_{Angles}(G_1, G_2) + w_4 * d_{Vertices}(G_1, G_2)
 \end{aligned}
 \tag{1}$$

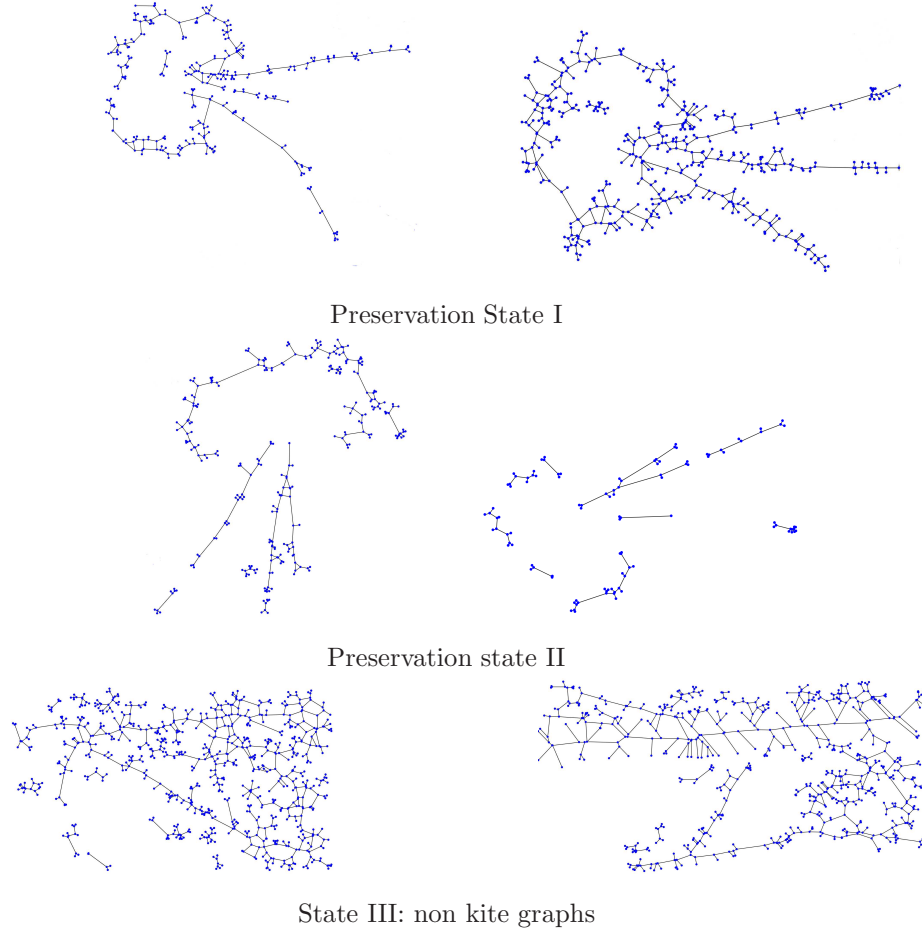


Fig. 2: Dataset.

where  $w_i, i = 1, 4$  are weighting coefficients with  $w_1 + w_2 + w_3 + w_4 = 1$ ,  $d_{Convex}(G_1, G_2)$  and  $d_{Angles}(G_1, G_2)$  compare the global geometric forms of the two kites based on the convexity of the angles and the total value of the angles respectively:

$$d_{Convex}(G_1, G_2) = \left| \frac{\|Angles_{G_1} < ConvexityThreshold\|}{\|Angles_{G_1}\|} - \frac{\|Angles_{G_2} < ConvexityThreshold\|}{\|Angles_{G_2}\|} \right| \quad (2)$$

where  $ConvexityThreshold$  is at most equal to  $180^\circ$  but it will be defined according to the form of the kites,  $Angles_{G_i}$  denotes the set of angles of graph  $G_i$ .

$$d_{Angles}(G_1, G_2) = \frac{|\sum Angle_{i,G_1} - \sum Angle_{j,G_2}|}{Max(\sum Angle_{i,G_1}, \sum Angle_{j,G_2})} \quad (3)$$

where  $Angle_{i,G}$  denotes the  $i^{th}$  angle of graph  $G$ .  $d_{Edges}(G_1, G_2)$  compares the global size of the two kites by comparing the distances reported on the edges of the corresponding graphs.

$$d_{Edges}(G_1, G_2) = \frac{\frac{\|E(G_1)\|}{\sum_{i=1}^{\|E(G_1)\|} \ell(e_i)} - \frac{\|E(G_2)\|}{\sum_{j=1}^{\|E(G_2)\|} \ell(e_j)}}{\text{Max}(\frac{\|E(G_1)\|}{\sum_{i=1}^{\|E(G_1)\|} \ell(e_i)}, \frac{\|E(G_2)\|}{\sum_{j=1}^{\|E(G_2)\|} \ell(e_j)})} \quad (4)$$

$d_{Vertices}(G_1, G_2)$  compares the order of the two graphs.

$$d_{Vertices}(G_1, G_2) = \frac{\|V(G_1)\| - \|V(G_2)\|}{\text{Max}(\|V(G_1)\|, \|V(G_2)\|)} \quad (5)$$

The algorithm takes as inputs the three prototype graphs, i.e.,  $G_{Antenna}$ ,  $G_{Enclosure}$  and  $G_{Kite}$  and a query graph. For each connected component of the query graph, the algorithm returns the most similar kite component.

The second level of the proposed approach is a local similarity measure where the graphs are compared based on their local descriptions using edit operations as in [15] and [14]. However, we extended local descriptions by considering angles degree and labels of the edges. We denote the obtained similarity measure by  $GeoL\_Sim$  for Geometric Local similarity measure to distinguish it from  $L\_Sim$  the local similarity measure of [15] and [14] that do not uses angles. Thus, with  $GeoL\_Sim$  each vertex of the compared graphs is represented by a signature, i.e. a vector, containing the degree of the vertex, the labels of its incident edges and the angles between each pair of consecutive incident edges. The similarity between two vertices  $v_1$  and  $v_2$  is computed as a distance between their signatures  $S(v_1)$  and  $S(v_2)$  as follows:

$$\begin{aligned} d(S(v_1), S(v_2)) = & \omega_1 * \frac{|deg(v_1) - deg(v_2)|}{\text{Max}(\Delta(G_1), \Delta(G_2))} \\ & + \omega_2 * \sum_{k=1}^{\text{Max}(\Delta(G_1), \Delta(G_2))} \frac{|\ell(e_{1,k}) - \ell(e_{2,k})|}{\text{Max}(\Delta(G_1), \Delta(G_2)) * \text{Max}(\mathcal{L}(G_1), \mathcal{L}(G_2))} \\ & + \omega_3 * \sum_{k=1}^{\text{Max}(\Delta(G_1), \Delta(G_2)) - 1} \frac{|Ang_{1,k} - Ang_{2,k}|}{(\text{Max}(\Delta(G_1), \Delta(G_2)) - 1) * \text{Max}(\mathcal{A}(G_1), \mathcal{A}(G_2))} \end{aligned} \quad (6)$$

where where  $\omega_i, i = 1, 4$  are weighting coefficients with  $\omega_1 + \omega_2 + \omega_3 = 1$ ,  $Ang_{i,k}$  is the  $k^{th}$  angle of vertex  $v_i$ .

Using these signatures, we construct a matrix  $M$  where  $M[i, j] = d(S(v_i), S(v_j))$  for each pair of vertices  $v_i$  and  $v_j$  such that  $v_i \in V(G_1)$  and  $v_j \in V(G_2)$ . We define the local similarity measure between  $G_1$  and  $G_2$  by:

$$GeoL\_Sim(G_1, G_2) = \min_h \sum_{v_i \in V(G_1), h(v_i) \in V(G_2)} d(S(v_i), S(h(v_i))) \quad (7)$$

where  $h : V(G_1) \rightarrow V(G_2)$  is a bijection.

Computing this distance is then equivalent to solve the affectation problem in the matrix  $|V(G_1)| \times |V(G_2)|$  in which each element represents the distance between the signature of the  $i^{th}$  vertex of ( $G_1$ ) and the signature of  $j^{th}$  vertex of  $G_2$ .



To do so, we use the Hungarian algorithm [10] to obtain the minimum cost in  $O(n^3)$  time.

Similarly to the global similarity algorithm, the local similarity algorithm takes as input a query graph representing an image and determines for each connected component of the query graph the most similar kite component.

When a query graph passes the local similarity measure with more than one connected component classified as a kite part, we need to know if these kite parts are parts of the same kite or belongs to different kites. This is the aim of the reconstruction step that uses the coordinates of the vertices to eventually reconstruct the entire kite from different components: i.e, enclosure and antennas.

### 3 Evaluation

For evaluation, we used all the available graphs in the real dataset described in Section 2. We undertook three series of experiments to evaluate the robustness and the accuracy of our similarity measure. We compared our approach with the initial framework based on local structures comparison presented in [15] and [14] and described in Section 1. This approach is denoted  $L\_Sim$  and do not use angles in the local structures of the vertices. In the first experiment, we focused on determining the optimal values for  $Form_{threshold}$  and the weighting coefficients of the global and the local similarity measures. Based on these experiments that concerned mainly the well preserved kite graphs (State-I dataset) and the non kite graphs (State-III dataset), we set  $Form_{threshold} = 0.5$  and gave an important value to the weight of the labels of the edges, i.e., the distances.  $ConvexityThreshold$  was set to  $150^\circ$ .

In the second series of experiments we evaluated the scalability of our approach over an increasing number of nodes in the query graphs. From the 600 graphs of the dataset, we constructed a set of query groups :  $Q_{10}, \dots, Q_{620}$  each of which contains 10 graphs. All the graphs of the same query group  $Q_i$  have the same number of vertices, i.e.,  $i$ . This means that the number of vertices of the query graphs varies from 10 to 620. This variation is obtained by adding 10 graphs till  $Q_{120}$  than by adding 100 graphs. Figure 3 shows the average runtime performance of  $G\_Sim$ ,  $GeoL\_Sim$  and  $L\_Sim$ . The  $X$ -axis shows the number of vertices contained in the query graph and the  $Y$ -axis the average runtime, in log scale, obtained over the query group of the corresponding graph size when compared to the 4 kite prototype graphs. This figure shows clearly the interest of using the global similarity level which is largely faster than the local similarity measure. The runtime performance of the two local similarity measures  $GeoL\_Sim$  and  $L\_Sim$  are equivalent and confirm their theoretic polynomial time complexity.

Finally, we evaluated the accuracy of the proposed approach by performing classification. Table 3 depicts our results. We can see that  $GeoL\_Sim$  is more accurate than  $L\_Sim$  on the three datasets. This confirms that considering angles has an important added value for kite recognition. We can also see that  $GeoL\_Sim$  and  $L\_Sim$  achieve better than  $G\_Sim$  in the positive datasets (those

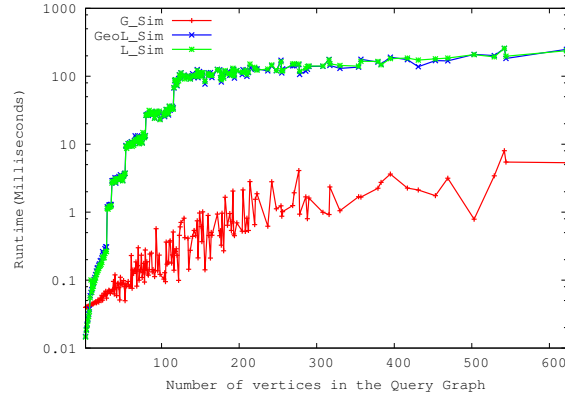


Fig. 3: Runtime Vs number of vertices.

Table 3: Classification

Dataset	<i>G_Sim</i>	<i>GeoL_Sim</i>	<i>L_Sim</i>
State - I	61	94	90
State - II	46	80	60
State - III	71	67.5	43

that contains kites). However *G\_Sim* is better in the negative dataset. So, by using successively *G\_Sim* and *GeoL\_Sim* we achieve good performance in both cases.

## 4 Conclusions

In this paper, we make use of the structural information of kites in a graph-based recognition framework, in which we take a query image, retrieve a graph representation of the image, and perform detailed matching to verify the similarity of the query graph with a database of kite prototype graphs. We propose a graph comparison framework that deals with the geometric form of the kite graphs and their disconnection. The key idea is to first discard the non kite graphs using a rapidly computable similarity measure based mainly on the geometric form of the kites. Our preliminary experimentation results are acceptable and encouraging. As a perspective, we are interested by comparing our approach with other graph similarity measures mainly those based on kernels and also with non graph based recognition solutions by implementing a feature and classifier based approach.

## Acknowledgment

This work was supported by the LABEX IMU (ANR-10-LABX-0088) of Universit de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

## References

1. Bunke, H., Riesen, K.: Recent advances in graph-based pattern recognition with applications in document analysis. *Pattern Recognition* 44, 1057–1067 (2011)
2. Chung, Y.C., Han, T., He, Z.: Building recognition using sketch-based representations and spectral graph matching. In: *Computer Vision, 2009 IEEE 12th International Conference on*. pp. 2014–2020 (Sept 2009)
3. Desolneux, A., Moisan, L., Morel, J.M.: Meaningful alignments. *Int. J. Comput. Vision* 40(1), 7–23 (Oct 2000)
4. Échallier, J.C., Braemer, F.: Nature and function of 'desert kites': new data and hypothesis. *Paléorient* 21(1), 35–63 (1995)
5. Foggia, P., Percannella, G., Vento, M.: Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence* 28(01), 1450001 (2014)
6. Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. *Pattern Analysis Applications* (13), 113–129 (2010)
7. von Gioi, R., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A fast line segment detector with a false detection control. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32(4), 722–732 (April 2010)
8. Helms, S., Betts, A.: The desert 'kites' of the badiyat esh-sham and north arabia. *Paléorient* 13(1), 41–67 (1987)
9. Illingworth, J., Kittler, J.: A survey of the hough transform. *Computer Vision, Graphics, and Image Processing* 44(1), 87 – 116 (1988)
10. Kuhn, H.W.: The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 83–97 (1955)
11. Lacoste, C., Descombes, X., Zerubia, J.: Point processes for unsupervised line network extraction in remote sensing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27(10), 1568–1579 (Oct 2005)
12. Lopresti, D.P., Wilfong, G.T.: Comparing Semi-Structured Documents via Graph Probing. In: *Workshop on Multimedia Information Systems*. pp. 41–50 (2001)
13. McKay, B.: Practical graph isomorphism. *Congress Numerantium* 87, 30–45 (1981)
14. Raveaux, R., Burie, J.C., Ogier, J.M.: A graph matching method and a graph matching distance based on subgraph assignments. *Pattern Recognition Letters* 31, 394–406 (2010)
15. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision Computing* 27, 950–959 (2009)
16. Rochery, M., Jermyn, I.H., Zerubia, J.: Higher order active contours. *Int. J. Comput. Vision* 69(1), 27–42 (Aug 2006)
17. Sanfeliu, A., Fu, K.: A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics (Part B)* 13(3), 353–363 (1983)
18. Ullmann, J.R.: An Algorithm for Subgraph Isomorphism. *J. ACM* 23(1), 31–42 (Jan 1976), <http://dx.doi.org/10.1145/321921.321925>
19. Vento, M.: A long trip in the charming world of graphs for pattern recognition. *Pattern Recognition* 48(2), 291–301 (2015)
20. Wang, Q., Jiang, Z., Yang, J., Zhao, D., Shi, Z.: A hierarchical connection graph algorithm for gable-roof detection in aerial image. *Geoscience and Remote Sensing Letters, IEEE* 8(1), 177–181 (Jan 2011)
21. Xiao, Y., Dong, H., Wu, W., Xiong, M., Wang, W., Shi, B.: Structure-based graph distance measures of high degree of precision. *Pattern Recognition* 41, 3547–3561 (2008)