Pattern Recognition Letters 87 (2017) 186–194

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A Graph-based approach for Kite recognition☆

Kamel Madi[a,*], Hamida Seba[a], Hamamache Kheddouci[a], Olivier Barge[b]

[a] Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, Lyon 69622, France
[b] CNRS, UMR 5133 Archéorient, Lyon 69365, France

ARTICLE INFO

Article history:
Available online 24 May 2016

Keywords:
Pattern recognition
Graph matching
Graph edit distance
Graph-based image modeling
Kite recognition
Satellite image

ABSTRACT

Kites are huge archaeological structures of stone visible from satellite images. Because of their important number and their wide geographical distribution, automatic recognition of these structures on images is an important step towards understanding these enigmatic remnants. This paper presents a complete identification tool relying on a graph representation of the Kites. As Kites are naturally represented by graphs, graph matching methods are thus the main building blocks in the Kite identification process. However, Kite graphs are disconnected geometric graphs for which traditional graph matching methods are useless. To address this issue, we propose a graph similarity measure adapted for Kite graphs. The proposed approach combines graph invariants with a geometric graph edit distance computation leading to an efficient Kite identification process. We analyze the time complexity of the proposed algorithms and conduct extensive experiments both on real and synthetic Kite graph data sets to attest the effectiveness of the approach. We also perform a set of experimentations on other data sets in order to show that the proposed approach is extensible and quite general.

1. Introduction

A Kite is an archaeological structure consisting of two long walls built of stones and arranged within a funnel shape opening onto an enclosure. The walls can reach a length of several kilometers and the enclosure can cover an area of several hectares. This yields huge constructions that are visible on satellite images as depicted in Fig. 2(a). Kites have been discovered in the Middle East since 1920. They were first discovered by the British airmen who flew over the Jordanian desert during the period of the Mandate. They were thus called Kites due to the analogy of their shape with the shape of a Kite. Despite several studies, the issues related to their age and functions remain without satisfactory answers. Some rare dating attributes them to the Bronze Age but predated use of these structures is not excluded. The exact function of these structures has never been established. Many authors attribute a hunting function to the Kites, but the hypothesis of a pastoral use has not been refuted. These uncertainties are due to the extreme difficulty of obtaining reliable data during field investigations in contexts where archaeological material is most often absent [7,11]. Recently, public access to high resolution satellite images (Google Earth, Bing) has significantly expanded the number of discovered Kites and also enlarged their geographical spread from the south of the Arabian Peninsula to the Aralo-Caspian region [1]. The massive use of Kites, judging by the density of these structures, probably had territorial implications and socioeconomic importance in a region that has seen the advent of agriculture and the birth of the urban phenomenon. Kites are thus an underestimated phenomenon. Establishing the duration of their utilization, outlining their use and functioning, and trying to identify the population responsible for 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 [4]. For this purpose, automatic recognition of Kites on satellite images offers archeologists valuable help in understanding this phenomenon. This will allow a systematic and homogeneous search in the entire distribution area of Kites and then in the peripheral regions.

In this paper, we present a complete framework for Kite recognition on satellite images where Kites are modeled by graphs. This representation is motivated by the natural graph form of Kites. Kite recognition as a graph matching problem is interesting because it raises several challenges not addressed by existing methods. In fact, Kite graphs are not connected and may contain several parts. They have specific geometric forms that distinguish them from other constructions. Furthermore, each processed image can involve a large number of graphs, thus implying the use of rapid

☆ This paper has been recommended for acceptance by Cheng-Lin Liu.

* Corresponding author. Tel.: +33 426 234 482; fax: +33 472 431 537.
E-mail addresses: kamel.madi@liris.cnrs.fr, kamel.madi88@gmail.com (K. Madi).

http://dx.doi.org/10.1016/j.patrec.2016.05.005
0167-8655/

recognition algorithms. To tackle these challenges, we propose a multi-level recognition framework that first applies rapidly computed graph invariants to discard non-Kite graphs in the early stages of the recognition framework. Then, we use a local similarity measure that takes into account the geometric form of Kite graphs by considering the angles of the form. Finally, a reconstruction process allows us to consider disconnection within Kite graphs. We construct a benchmark of Kite graphs from real images to evaluate the efficiency of our framework. We also generate a synthetic data set to evaluate the resilience of the proposed method to different preservation states of Kites. We compare our work with existing methods and we also apply it to other data sets mainly characterized by the geometric form of the graphs. These experimentations show that the proposed framework is a practical and efficient Kite recognition tool that applies directly to images.

A preliminary version of the current paper appeared in [17]. The current paper has been significantly extended with respect to the underlying methodology and the experimental evaluation. Firstly, we added a detailed description of the extraction process of Kites' graphs from real images and enriched the benchmark with new Kites. We also enlarged our experimentations with a synthetic data set generated randomly with various levels of deformations. This synthetic data set allowed us to attest the resiliency of the proposed approach. Secondly, we added a comparison with existing approaches and proved that the proposed approach is quite general by performing experimentations on the well-known GREC data set [21].

The remainder of the paper is organized as follows: Section 2 presents related works. In Section 3, we explain the process of constructing and generating of the real and the synthetic data sets used to evaluate our approach. Section 4 describes the proposed similarity measure and presents its complexity analysis. Section 5 reports our experimental results and finally, Section 6 concludes the paper.

2. Related work

A graph $G = (V, E)$ is a set of vertices V (also called nodes) connected by a set of edges $E \subseteq V \times V$. A finite number of labels are associated with vertices and/or edges. Graphs are a powerful representation tool and a popular formalism used in many applications of structural pattern recognition and classification [8,27]. For these kinds of applications, graph matching and, more generally, graph comparison is a fundamental issue. Graph matching is the process of finding a correspondence between vertices and edges of two graphs that satisfies a certain number of constraints, ensuring that similar substructures in one graph are mapped to similar substructures in the other. Graph matching solutions are classified into two wide categories: exact approaches and inexact approaches. Exact approaches, such as those that test for graph isomorphism or sub-graph isomorphism [18,26], 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. Inexact graph matching computes a distance between the compared graphs. This distance measures how similar (or dissimilar) are the graphs and deals 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 [3,27] 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.

Graph edit distance (GED) is one of the most famous and powerful fault-tolerant graph matching measures to determine the distance between graphs [2,25]. It is based on a kind of graph transformation called an edit operation. An edit operation is either an insertion, a suppression or a substitution 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 [24]. This sequence is called an optimal edit path. Tolerance to noise and distortion is one of the advantages of GED. Unfortunately, computing the exact value of the edit distance between two graphs is NP-Hard for general graphs and induces an exponential computational complexity. This motivated the apparition of several heuristics that approach the exact value of GED in polynomial time using different methods such as dynamic programming and probability. We refer the reader to [9] for a detailed survey and we describe here an approach that partitions the compared graphs into smaller substructures and approximates GED by computing edit distance between substructures. These substructures are generally stars, i.e., vertices with their direct neighbors and edges as illustrated in Fig. 1. These substructures are called local descriptions in [22], stars in [30], b-stars in [29] and probe vectors in [20]. The edit distance between substructures is achieved in $\mathcal{O}(n^3)$ time steps. Another approximation called BEAM is proposed in [19], where the authors present a fast suboptimal graph edit distance search which is a variant of a standard A^* *algorithm* reducing the search space. Rather than expanding all successor vertices in the search tree, only a fixed number of vertices to be processed are kept in the set of open vertices at all times. The search space is not completely explored, only the vertices belonging to the most promising partial matches are expanded.

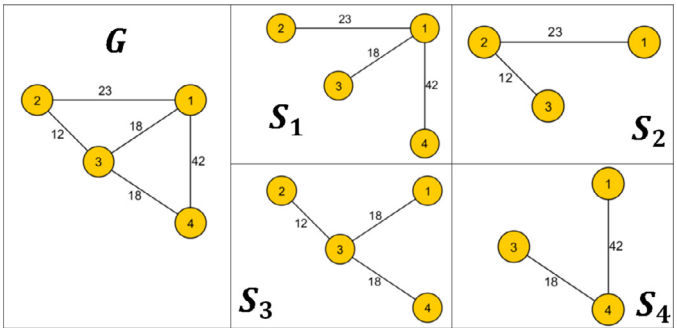


Fig. 1. Example of stars of a graph.

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 [18], which is one of the most efficient algorithms 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 [16], 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 [28] to compare biological data.

In this paper, we propose to unify the computation speed of graph invariants to the fault tolerance of GED in a similarity measure adapted to particular graphs that represent Kites, the archeological structures described in Section 1.

3. Kite graph data set construction

In this section, we present the process of Kite graphs construction from real images, and the process of generating a synthetic

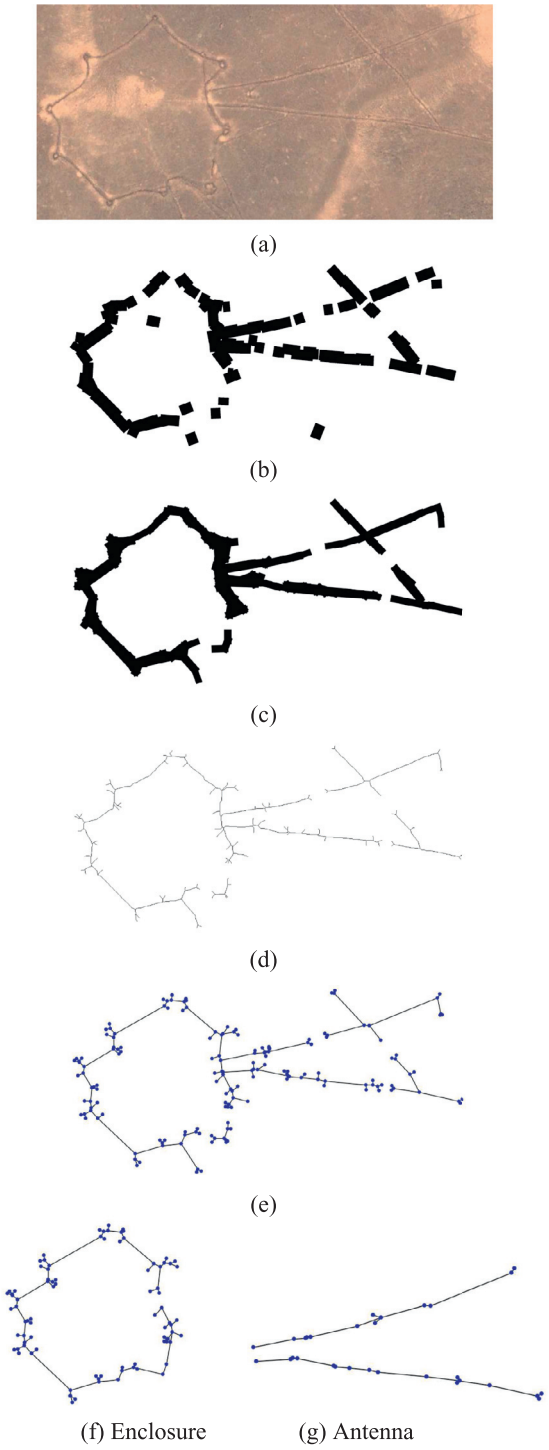


Fig. 2. Illustration of Kite detection.

data set of Kite graphs generated randomly. The two data sets (real and synthetic) are used to evaluate our approach.

3.1. Real data set construction

On satellite images, Kites appear as flat surfaces delimited by a set of lines as illustrated in (Fig. 2(a)). To convert Kites' images into attributed graphs, the first step is to extract the Kite structures from the images by edge detection. Edge (segment or line) detection in images is an intensively studied topic in image analysis [5,12]. Besides, several recent methods such as [10,14,23] give good results 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 Kites, we investigated several solutions with various settings and the LSD algorithm [10] gave us the most satisfactory set of segments (see Fig. 2(b)). The LSD algorithm is followed by four steps to obtain the final Kite graphs:

- **Deleting isolated segments**: We consider that a segment is isolated if its length is less than a threshold $length_{min}$ and if it has no neighbors according to a minimum neighborhood distance $neighbor_{min}$.
- **Merging neighboring segments**: During this step, each pair of segments that are neighbors according to $neighbor_{min}$, do not cross each other and have the same angle with the horizontal line with a tolerance angle *delta*, are merged in one segment. $length_{min}$, $neighbor_{min}$ and *delta* are set during experimentations. Deleting isolated segments and merging neighboring ones are illustrated in Fig. 2(c).
- **Thinning segments**: In this step, a skeleton is generated by reducing the width of all the segments to 1 pixel (see Fig. 2(d)) using the Skeletonize "ImageJ" method, which is the implementation of the approach described in [15].
- **Graph construction**: Finally, we construct the graph from the skeleton by representing lines by edges and ending points of lines by vertices (Fig. 2(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 more connected components (i.e., some parts of the Kite have been destroyed).

We executed our algorithm on 350 images (250 with Kites and 100 without Kites) with different states of preservation. We classified the obtained graphs into four preservation levels:

1. **State I**: The Kite is entire and well preserved. The Kite graph obtained is perfect and the few disconnections found are corrected manually with the help of the archeologists.
2. **State II**: The Kite is entire and well preserved. The Kite graph may be disconnected but the disconnections are neither frequent nor important.
3. **State III**: The Kite graph is very disconnected. Some parts of the Kite are not present.
4. **State IV**: The graph is not a Kite. These graphs are obtained by executing the algorithm on images that do not contain Kites. These images are extracted near (geographical positions) the images containing Kites, so these images have the same reliefs as the images containing the Kites, and the graphs obtained represent structures close to Kites.

Fig. 3 depicts some examples in each case. The characteristics of the data set are summarized in Table 1.

Kite graphs Prototype(Real). With the help of the archeologists, we selected from the graphs in **State-I**, the most preserved Kites as prototype Kite graphs. Also, to be able to deal with disconnected Kite graphs without adding significant computing costs, we constructed a prototype graph for each Kite component, namely: antenna and enclosure. Fig. 2(f) and (g) give, respectively, an example of a Kite enclosure and a Kite antenna. In our experimentation, we consider a Graph Antenna, a Graph Enclosure and four different Graph Kites.

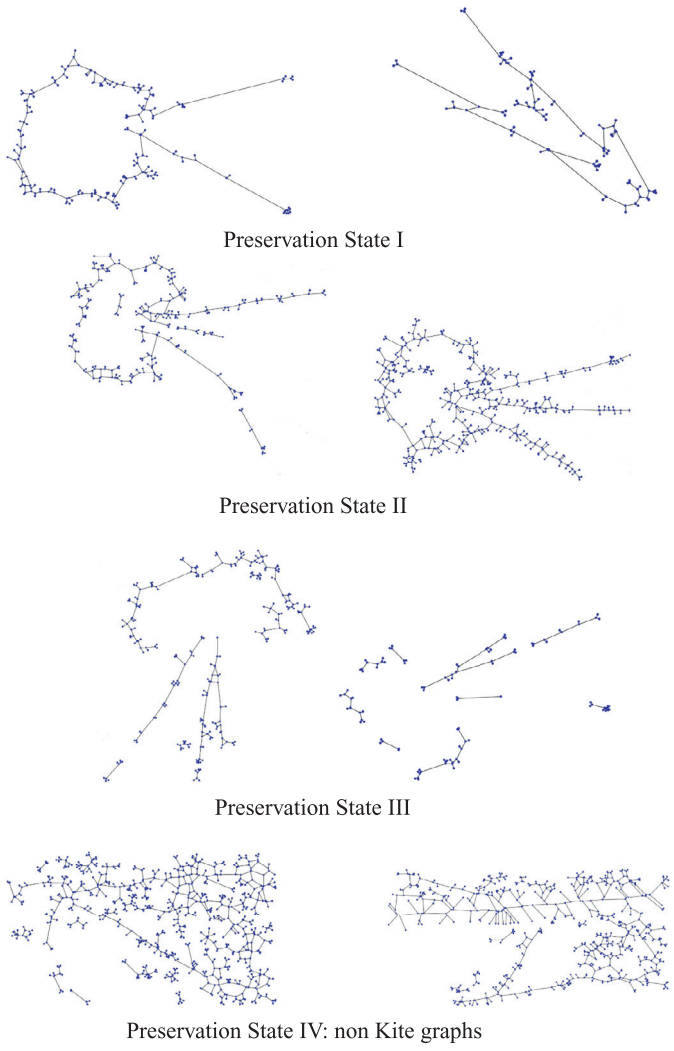


Fig. 3. Real data set.

Table 1
Real Data set Characteristics #*G*: number of graphs. #*Img*: number of images. *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.

	All the Data set	State-I	State-II	State-III	State-IV
#*G*	4081	62	129	1581	2309
#*Img*	350	50	100	100	100
avg (*V*)	26.14	110.84	113.74	30.09	15.59
max (*V*)	949	316	320	779	949
avg (*E*)	26.28	116.51	122.34	30.56	15.90
max (*E*)	1081	327	331	864	1081
avgAng	91.22	91.19	91.31	91.24	91.15
maxAng	180	180	180	180	180

3.2. Synthetic data set generation

Random generation of a synthetic data set of Kite graphs offers us the possibility of:

- obtaining Kite graphs in several possible preservation states.
- obtaining Kite graphs with numerous deformations, which may correspond to the variations in form of Kite components or the absence of one or more of these components.
- studying the scalability and resilience of our Kite recognition process.

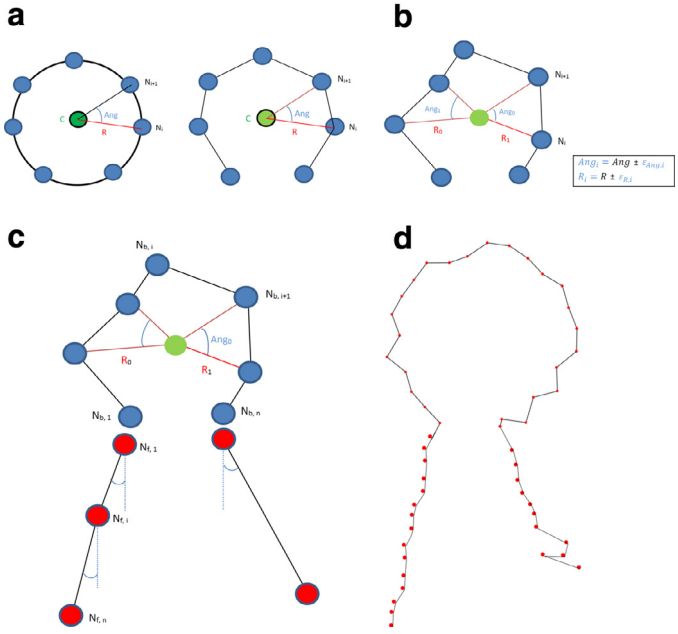


Fig. 4. Kite graphs synthetic data set generation process.

Table 2
Synthetic Graph Data set Characteristics #*G*: number of graphs. *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.

#*G*	*avg* (*V*)	*max* (*V*)	*avg* (*E*)	*max* (*E*)	*avgAng*	*maxAng*
1000	58.44	90	57.41	89	108.89	180

In order to generate a graph representing a Kite (Fig. 4 (d)), we generate the graphs of each component, namely the enclosure graphs and the antenna graphs. The different parameters used to generate the graphs of each Kite component are checked and controlled by a team of Kite expert archeologists.

Enclosure graph generation. Due to the form of the Kite enclosure which is pseudo-convex, the generation of its graph is based on a circle equation. The center position *c*, the number of vertices *N* and the radius circle *R* are generated randomly according to a minimum and a maximum limit defined by the archaeologists. An angle *Ang* is generated according to the number of vertices in the Kite enclosure (see Fig. 4(a)). The coordinates (*x*, *y*) of the vertices of the Kite enclosure are generated according to the circle equation. To obtain the *pseudo-convex* form of the enclosure, we vary the values of the radius ($R \pm \varepsilon_{R,i}$) and the angle ($Ang \pm \varepsilon_{Ang,i}$) for each generation of vertex coordinates (see Fig. 4(b)).

Antenna graph generation. A Kite antenna is represented by a graph that is an open chain of edges (at least one edge). The number of edges, the distance between two vertices constituting an edge, and the inclination angle of an edge are generated randomly depending on a set of minimum and maximum values of the condescending parameters (see Fig. 4(c)).

Using the described generation process, we obtain a synthetic data set containing 1000 graphs representing Kites. The characteristics of the synthetic data set are summarized in Table 2.

Kite graphs Prototype(Synthetic). Using the described process, we generate a set of prototype graphs representing: an antenna, an enclosure and four entire Kites.

Table 3
Notation.

Symbol	Description
$G(V, E)$	undirected labeled graph where V is 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.
$(\angle v_1v_2v_3)_{G_i}$	the angle between the two edges $(v1, v2)$ and $(v2, v3)$ in the graph G_i.

4. Algorithm overview

In this section, we describe the proposed Kite recognition solution, which is a hierarchical graph-based approach consisting of: approaches measuring the distance between two graphs and a reconstruction process. Firstly we present the proposed approaches measuring the distance between two graphs: a global similarity measure denoted *Global*, a geometric local similarity measure denoted *GeoLocal* and two varieties of hierarchical measures that we call $Global_{GeoLocal}$ and $GeoLocal_{Global}$ which are the result of combining *Global* and *GeoLocal* depending on the defined order. The global similarity *Global* is a fast computable measure based on graph invariants. This similarity aims to rapidly discard the graphs that cannot be Kites and avoid unnecessary and more costly comparisons. The geometric local similarity *GeoLocal* is a more accurate measure based on the geometric form and the structured features extracted from the graphs. This similarity is based on graph edit distance *GED* to deal with the state of preservation of the Kites. Secondly, we present the reconstruction process, which aims to verify if the different connected components of the graph identified as Kite components (enclosure and antenna) constitute a Kite. Identification of the different connected components of the graph as Kite components is realized using one of the proposed approaches of graph similarity measure, namely: *Global*, *GeoLocal* or one of two hierarchical measures $Global_{GeoLocal}$ or $GeoLocal_{Global}$. Finally, we present the computational complexity of the proposed algorithm.

Table 3 summarizes the notations that we use in the remainder of the paper.

4.1. Global similarity

Global similarity computes graph invariants. We consider the number of vertices of the compared graphs, the labels of the edges, which correspond to the length of the Kite walls, and the angles between edges. So, the global similarity between two graphs G_1 and G_2 is given by:

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

where w_i is a weighting coefficient with $\sum_{i=1}^{i=4} w_i = 1$, $d_{Vertices}(G_1, G_2)$ compares the order of the two graphs.

$$d_{Vertices}(G_1, G_2) = \frac{\big|\|V(G_1)\| - \|V(G_2)\|\big|}{Max\big(\|V(G_1)\|, \|V(G_2)\|\big)} \quad (2)$$

$d_{Edges}(G_1, G_2)$ compares the global size of the two Kites by comparing the distances reported on the edges of the corresponding graphs.

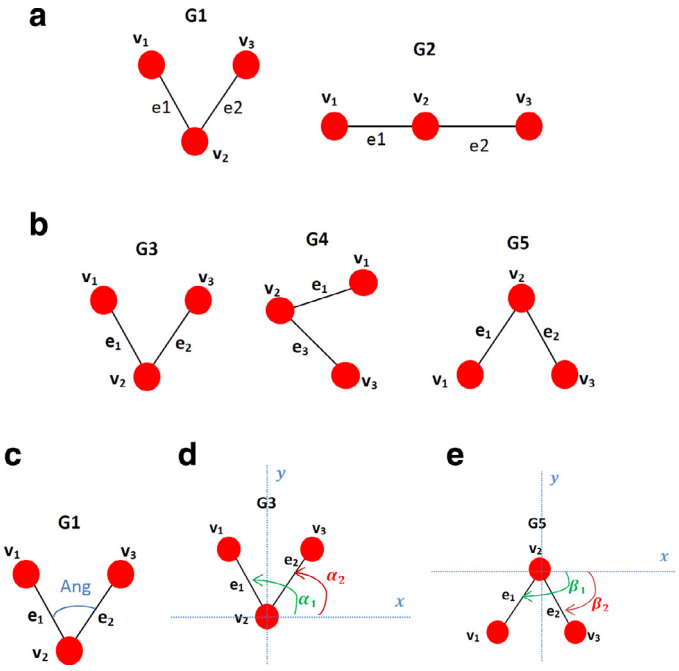


Fig. 5. Example of isomorphic graphs.

$$d_{Edges}(G_1, G_2) = \frac{\Big|\sum_{i=1}^{\|E(G_1)\|} \ell(e_i) - \sum_{j=1}^{\|E(G_2)\|} \ell(e_j)\Big|}{Max\Big(\sum_{i=1}^{\|E(G_1)\|} \ell(e_i), \sum_{j=1}^{\|E(G_2)\|} \ell(e_j)\Big)} \quad (3)$$

$d_{Convex}(G_1, G_2)$ 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:

$$d_{Convex}(G_1, G_2) = \Big|\frac{\|Angles_{G_1} < ConvexityTh\|}{\|Angles_{G_1}\|} - \frac{\|Angles_{G_2} < ConvexityTh\|}{\|Angles_{G_2}\|}\Big| \quad (4)$$

where $Angles_{G_i}$ denotes the set of angles of graph G_i and $ConvexityTh$ is an angle threshold at most equal to 180°. However, it will be defined according to the form of the Kites.

$$d_{Angles}(G_1, G_2) = \frac{|\sum Angle_{i,G_1} - \sum Angle_{j,G_2}|}{Max(\sum Angle_{i,G_1}, \sum Angle_{j,G_2})} \quad (5)$$

where $Angle_{i,G}$ denotes the ith angle of graph G.

The algorithm takes as inputs a set of prototype graphs, which are: $G_{Antenna}$, $G_{Enclosure}$, four different G_{Kite} and a query graph. For each connected component of the query graph, the algorithm returns the most similar Kite component.

4.2. Geometric local similarity

The geometric local similarity measure *GeoLocal* is a distance based on the approximation of the graph edit distance that compares the graphs using local descriptions of substructures (see Fig. 1). However, unlike the approaches proposed in [20,22,30], in our approach we extended local descriptions by considering angles in addition to degrees of vertices and labels of the edges. This allows us to distinguish between two isomorphic graphs with different geometry (Fig. 5(a)). 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 form of these graphs. Some authors even use the

angle as an attribute or a label associated with an edge [21]. This attribute represents the angle between the considered edge and a horizontal or a vertical line landmark (Fig. 5(d, e)). The drawback of this representation is that the value of the angle may change if a rotation is applied. This is not a problem in some graph representations such as graphs representing letters, digits, etc. However, considering a model that resists rotation and other deformations is very important when the graphs represent objects with specific forms as is the case of Kite graphs. Thus, in our framework two graphs are isomorphic if they also have the same form according to the following definition.

Definition 1 (geometrical isomorphic)**.** Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ be two graphs. G_1 and G_2 are *geometrical isomorphic* if they are *isomorphic* and have the same geometric form.

Example 1. Let $G_i(V, E)$ where $i = 1, \ldots, 5$, five graphs, such that: $V = \{v_1, v_2, v_3\}$ and $E = \{e_1, e_2, e_3\}$ (Fig. 5(a, b)). We can easily find a mapping between the set of vertices of G_1 and G_2 ensuring edge preservation, thus G_1 and G_2 are isomorphic. However, they are not *geometrical isomorphic* because they do not have the same geometric form ($(\angle v_1v_2v_3)_{G_1} \neq (\angle v_1v_2v_3)_{G_2}$), (see Fig. 5(a)). G_3, G_4 and G_5 (see Fig. 5(b)) are *geometrical isomorphic*: because they are *isomorphic* and have the same geometric form ($(\angle v_1v_2v_3)_{G_3} = (\angle v_1v_2v_3)_{G_4} = (\angle v_1v_2v_3)_{G_5}$). However, if we consider angles between edges and the horizontal axis x, G_3 and G_5 in (Fig. 5(d, e)) are *isomorphic* but not *geometrical isomorphic*: ($\alpha_{1,G_3} \neq \beta_{1,G_5}$ and $\alpha_{2,G_3} \neq \beta_{2,G_5}$).

In order to compute the geometric local similarity, each vertex v is represented by a signature *i.e.*, a vector defining its local structure as follows: $s(v) = \{deg(v), \{Ang_i, \ell(e_{i,1}), \ell(e_{i,2})\}_{i=1}^{i=(deg(v)-1)}\}$, where:

- $deg(v)$ is the degree of the vertex v.
- $\ell(e_{i,1})$ and $\ell(e_{i,2})$ are the labels (weights) of the two edges e_1 and e_2 constituting the angle Ang_i. $\ell(e_{i,1})$ and $\ell(e_{i,2})$ are ranked in descending order.
- The triplets $\{Ang_i, \ell(e_{i,1}), \ell(e_{i,2})\}$ are ranked according to the angle Ang_i in descending order.
- All the vertices are represented by signatures *i.e.,* vectors which have the same size: $size = 1 + ((\Delta(G_1, G_2) - 1) * 3)$.
- $\Delta(G_1, G_2)$ is the greatest vertex degree in the compared graphs G_1 and G_2.
- If a vertex v has a degree less than $\Delta(G_1, G_2)$, the rest of the vector is completed with *zeros*.

The similarity measure d between two signatures s_1 and s_2 is computed as follows:

$$d(s_1, s_2) = 1 - \sum_{i=1}^{i=3} \omega_i * F_i \tag{6}$$

The functions F_i are defined as follows:

$$F_1 = \frac{|deg(v_1) - deg(v_2)|}{Max(\Delta(G_1), \Delta(G_2))} \tag{7}$$

$$F_2 = \sum_{k=1}^{Max(\Delta(G_1),\Delta(G_2))} \frac{|\ell(e_{1,k}) - \ell(e_{2,k})|}{Max(\Delta(G_1), \Delta(G_2)) * Max(\mathcal{L}(G_1), \mathcal{L}(G_2))} \tag{8}$$

$$F_3 = \sum_{k=1}^{Max(\Delta(G_1),\Delta(G_2))-1} \frac{|Ang_{1,k} - Ang_{2,k}|}{(Max(\Delta(G_1), \Delta(G_2)) - 1) * Max(\mathcal{A}(G_1), \mathcal{A}(G_2))} \tag{9}$$

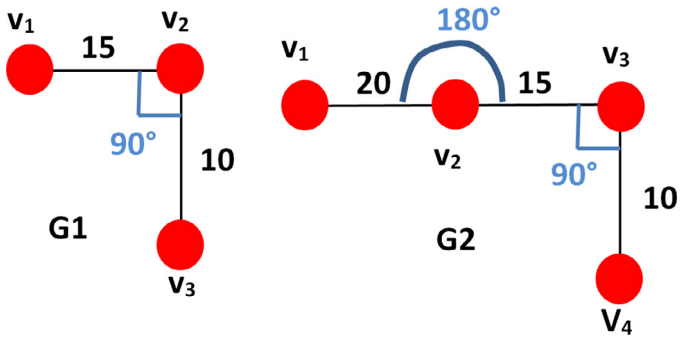


Fig. 6. The graphs G1 and G2 in Example 2.

where ω_i are weighting coefficients with $\sum_{i=1}^{i=3} \omega_i = 1$, $Ang_{i,k}$ is the kth angle of vertex v_i. $F_{i=1\ldots3}$ compares, respectively, the degree of the vertices, the labels of edges and the angles.

The Geometric Local Similarity *GeoLocal* aims to determine the best matching between the signatures (defining the local structure of each vertex) associated with the two compared graphs. Formally, let G_1 and G_2 be two graphs, S_1 and S_2 their corresponding sets of signatures, and M the set of all possible matching between S_1 and S_2. The similarity $GeoLocal(S_1, S_2)$ is formulated as follows:

$$GeoLocal(S_1, S_2) = 1 - \frac{\max\limits_{m \in M} \sum\limits_{s_i \in S_1,\, m(s_i) \in S_2} d(s_i, m(s_i))}{\max(\|S_1\|, \|S_2\|)} \tag{10}$$

Computation of $GeoLocal(S_1, S_2)$ is equivalent to solving the assignment problem which is a fundamental combinatorial optimization problem that aims to find the minimum/maximum weight matching in a weighted bipartite graph. To solve this assignment problem, we define a $n \times n$ matrix D, where $n = \max(\|S_1\|, \|S_2\|)$. Each element $D_{i,j}$ of the matrix represents the similarity measure $d(s_i, s_j)$ between a signature s_i in S_1 and a signature s_j in S_2. In the case of $\|S_1\| \neq \|S_2\|$, the smallest set of signatures is completed by $(max(||S_1||, ||S_2||) - min(||S_1||, ||S_2||))$ empty signatures ε. The similarity between an empty signature ε and a signature s is computed by the formula (Eq. (6)) and corresponds to the cost of adding s to the small graph (or of deleting s from the large graph).

We apply the Hungarian algorithm [13] on the matrix D in order to find the best assignment in $\mathcal{O}(n^3)$ time.

The resulting distance (dissimilarity) is compared to a threshold $th \in [0, 1]$ defined by an expert or by experimentation, in order to decide if the compared graphs are similar or not. Like the algorithm of *Global*, the algorithm of *GeoLocal* takes as inputs a set of prototype graphs, which are: $G_{Antenna}$, $G_{Enclosure}$, four different G_{Kite} and a query graph. For each connected component of the query graph, the algorithm returns the most similar Kite component.

Example 2. Let G_1 and G_2 be two graphs (see Fig. 6), S_1 and S_2 their corresponding sets of signatures, such that $\|S_1\| = 3$ and $\|S_2\| = 4$. Let D, be the matrix of similarities between S_1 and S_2. Where $D_{i,j} = d(s_i, s_j)$. $\|S_1\| < \|S_2\|$, thus we add an ε signature to S_1 and we complete the matrix D by $d(\varepsilon, s_{2,j=0\ldots3})$.

$$\begin{array}{c} \\ s_{1,0} \\ s_{1,1} \\ s_{1,2} \\ \varepsilon \end{array} \begin{array}{c} \begin{array}{cccc} s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \end{array} \\ \begin{pmatrix} 0.96 & \mathbf{0.33} & 0.58 & 0.96 \\ 0.54 & 0.75 & \mathbf{1} & 0.54 \\ 0.92 & 0.29 & 0.54 & \mathbf{1} \\ \mathbf{0.92} & 0.04 & 0.29 & 0.75 \end{pmatrix} \end{array}$$

The max sum is 3.25. The normalized dissimilarity is $GeoLocal(S_1, S_2) = 1 - \frac{3.25}{4} = 0.1875$.

The signature $s_{2,0}$ (of the node v_1 in G_2) is deleted ($s_{2,0} \rightarrow \varepsilon$).

4.3. Hierarchical similarity measure

In this section, we present two hierarchical measures that we call $Global_{GeoLocal}$ and $GeoLocal_{Global}$ which are the result of combining the global similarity measure *Global* and the geometric local similarity measure *GeoLocal* depending on the defined order.

Global geometric-Local similarity. The Global geometric-Local similarity $Global_{GeoLocal}$ is a hierarchical similarity measure, which aims to measure the distance between two graphs using firstly the global similarity measure *Global*, then using the geometric local similarity measure *GeoLocal* if necessary. The main idea is to measure the distance between the two graphs using *Global*. If the distance obtained is less than a specific threshold, which means that the two graphs are similar according to *Global*, then we check this result using *GeoLocal*. Otherwise, the two graphs are not similar, which means that we do not need to use *GeoLocal*. $Global_{GeoLocal}$ aims to enhance time processing of Kite graphs by first computing invariants on the graphs. Formally, let: G_1 and G_2 be two graphs, S_1 and S_2 the set of signatures of G_1 and G_2 respectively, $th \in [0, 1]$ a threshold and $d_1 = Global(G_1, G_2)$.

$$Global_{GeoLocal}(G_1, G_2) = \begin{cases} d_1, \ if\, d_1 > th \\ GeoLocal(S_1, S_2), \ otherwise \end{cases} \tag{11}$$

Geometric-Local Global similarity. Like $Global_{GeoLocal}$, the Geometric-Local Global similarity $GeoLocal_{Global}$ is a hierarchical similarity measure, which aims to measure the distance between two graphs using firstly *GeoLocal*, then using *Global* if necessary. The main idea is to measure the distance between the two graphs using *GeoLocal*. If the distance obtained is less than a specific threshold, which means that the two graphs are similar according to *GeoLocal*, then we check this result using *Global*. Otherwise, the two graphs are not similar, which means we do not need to use *Global*. However, only the vertices assigned in the phase of *GeoLocal* will be considered in the second phase using *Global*. In the case where the two graphs have the same number of vertices, all the vertices will be considered in the second phase, *i.e.*, *Global*. $GeoLocal_{Global}$ aims to improve the graph invariants in the second level by only considering the assigned vertices in the first level using *GeoLocal*. Formally, let: G_1 and G_2 be two graphs, G_2 is the graph prototype, S_1 and S_2 the sets of signatures of G_1 and G_2 respectively, $th \in [0, 1]$ a threshold, $d_2 = GeoLocal(S_1, S_2)$ and G'_1 is the subgraph induced by the vertices of G_1 assigned in the first phase using *GeoLocal*.

$$GeoLocal_{Global}(G_1, G_2) = \begin{cases} d_2, \ if\, d_2 > th \\ Global(G'_1, G_2), \ otherwise \end{cases} \tag{12}$$

4.4. Reconstruction process

Each connected component from the whole graph representing the query image is compared to the set of prototype graphs ($G_{Antenna}$, $G_{Enclosure}$ and four different G_{Kite}), using the proposed similarity measure (*GeoLocal*, *Global*, $Global_{GeoLocal}$ or $GeoLocal_{Global}$). Consequently, each connected component (a query graph) is classified as Kite, a part of Kite or not a Kite nor a part of Kite. When a query graph passes the considered similarity measure (*GeoLocal*, *Global*, $Global_{GeoLocal}$ or $GeoLocal_{Global}$) with more than one connected component classified as a Kite part and at least one of them is classified as an enclosure, we need to know if these Kite parts are parts of the same Kite or belong 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. The principle is to associate a subset of Kite parts classified as antennas to a Kite part classified as an enclosure, taking into account the distance between them and their orientations. The aggregated similarity Sim_{aggre} of the reconstructed Kite *i*, is given by:

$$Sim_{aggre}(Kite_i) = \psi * Sim(E_i) + \sum_{j=1}^{j=n} \mu * Sim(A_{i,j}) \tag{13}$$

Where: $\psi + \mu = 1$, $Sim(E_i)$ is the similarity attributed to the enclosure of the reconstructed Kite *i*, *n* is the number of antennas and $Sim(A_{i,j})$ is the similarity attributed to an antenna *j* of the reconstructed Kite *i*.

4.5. Complexity study

For the Geometric local similarity measure *GeoLocal*, the most important part, in term of complexity, is the one solving the assignment problem. We used the Hungarian algorithm [13] to find the best assignment in $\mathcal{O}(n^3)$ time, where *n* is the *maximum* number of vertices in the two compared graphs. Consequently, the time complexity of *GeoLocal* is $\mathcal{O}(n^3)$. The Global similarity measure *Global* is based on a graph invariant, which is linear in terms of computational complexity. Thus, the time complexity of *Global* is $\mathcal{O}(n)$, where *n* is the *maximum* number of vertices in the two compared graphs.

5. Experimental results

For evaluation, we used all the available graphs in the real data set described in Section 3.1 and the synthetic data set described in Section 3.2. We also used a well-known graph data set of symbols from architectural and electronic drawings named GREC [21], which is one of the data sets of the IAM graph database repository. The GREC data set is composed of 1100 undirected graphs distributed over 22 classes from the original GREC database [6]. The GREC data set is split into a training and a validation set, each of size 286, and a test set of size 528.

We conducted four series of experiments to evaluate the robustness and accuracy of our similarity measures. The first three series of experiments are realized on the real and synthetic Kite graph database, while the fourth experimentation is realized on the GREC data set. We compared our approach with two approaches from the state-of-the-art based on local structure comparison:

- $\boldsymbol{GED_{Bipartite}}$: a *GED* based on a bipartite assignment of vertices and their local structures [22].
- $\boldsymbol{Beam_{GED}}$: a simple and fast suboptimal GED based on beam search [19].

The proposed distances *GeoLocal* and *Global* are parameterized distances having a set of parameters α_k allowing different configurations. The default value is: $\sum \alpha_k = 1, \forall k$ and $ConvexityTh = 150^\circ$. In addition, we defined a threshold in order to improve classification accuracy. The parameters α_k and the threshold may be specified by inspection or by using machine learning techniques. In this paper, for simplicity, we attribute to all the parameters of our methods and the methods with which we compare ($GED_{Bipartite}$ and $Beam_{GED}$) their default values. However, for each approach we choose the threshold giving the best accuracy. The default parameters of $GED_{Bipartite}$ and $Beam_{GED}$ are: the same cost for vertex/edge deletions/insertions which is 1, the weighting parameters per vertex/edge is the same 1, the same cost for vertices and edges ($vertexCost = edgeCost$) and for $Beam_{GED}$, the size of the OPEN set is 10.

In the first experiment, we show the impact of using the reconstruction process in the obtained accuracy. Table 4 depicts the results obtained depending on the use or not of the process of reconstruction ($Threshold = 0.28$). These experiments are conducted on the real Kite data set.

Table 4
The impact of the reconstruction process on the classification.

Methods	*Reconstruction*	State-I	State-II	State-III	State-IV
Global	Yes	93.87%	96%	83%	77%
	No	89.79%	95%	81%	80%
GeoLocal	Yes	100%	98%	91%	78%
	No	93.87%	93%	87%	78%
$Global_{GeoLocal}$	Yes	91.83%	94%	78%	78%
	No	87.75%	92%	76%	82%
$GeoLocal_{Global}$	Yes	93.87%	95%	85%	78%
	No	89.79%	93%	83%	77%

Table 5
Classification on the Kite data set.

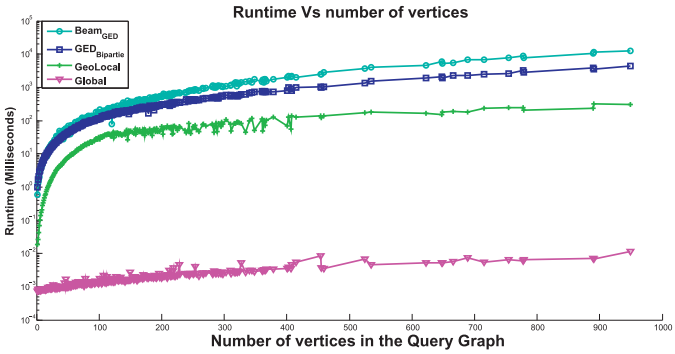
Methods	*Threshold*	State I	State II	State III	State IV	Synthetic data set
Global	0.28	93.87%	96%	83%	77%	98%
GeoLocal	0.28	100%	98%	91%	77%	100%
$Global_{GeoLocal}$	0.28	91.83%	94%	78%	78%	98%
$GeoLocal_{Global}$	0.28	93.87%	95%	85%	78%	98%
$GED_{Bipartite}$	0.40	36.53%	41%	75%	11%	41.3%
$Beam_{GED}$	0.10	20.20%	28%	75%	44.44%	75%

We can see that the four methods are globally more accurate with considering the process of reconstruction. This shows the importance of using the reconstruction process.

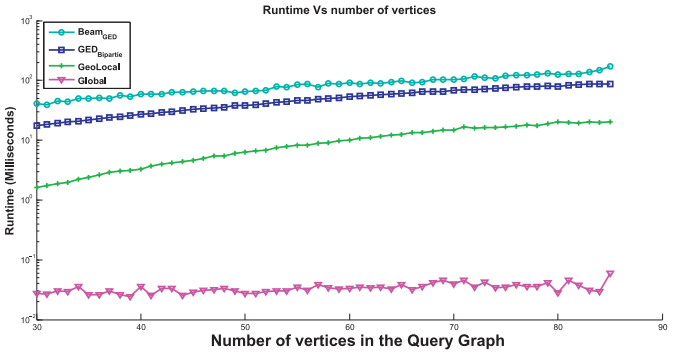
In the second series of experiments, we evaluated the accuracy of the proposed approach by performing classification. These experiments are realized on both the real and the synthetic Kite data set. Table 5 depicts the results obtained by our approaches and the approaches with which we compare, using the adequate threshold. We can see that our approaches *GeoLocal*, *Global*, $Global_{GeoLocal}$ and $GeoLocal_{Global}$ are more accurate than $GED_{Bipartite}$ and $Beam_{GED}$ at all the levels of the real and the synthetic Kite data set. This confirms that considering the geometric form (angles) has a high added value for Kite recognition. We can also see that *GeoLocal* is more accurate than *Global*, $Global_{GeoLocal}$ and $GeoLocal_{Global}$ at all the levels of the real and the synthetic Kite data set. However, $Global_{GeoLocal}$ and $GeoLocal_{Global}$ are slightly better in the negative data set (**State IV**) of the real data set. Although, *GeoLocal* achieves better classification accuracy compared to $Global_{GeoLocal}$ and $GeoLocal_{Global}$. However, use of the hierarchical measures $Global_{GeoLocal}$ and $GeoLocal_{Global}$ avoids unnecessary comparison in the second level, thus the general runtime on the data set is better. We note also that $GeoLocal_{Global}$ achieves better classification accuracy compared to $Global_{GeoLocal}$ at all the levels of the real and the synthetic Kite data sets. However, $Global_{GeoLocal}$ achieves a better general runtime on the data set, due to the fact that *Global* is faster than *GeoLocal*.

In the third series of experiments we evaluated the scalability of our approach over an increasing number of vertices in the query graphs. These experiments are realized on both the real Kite data set and the synthetic Kite data set. From the 4081 graphs of the real Kite data set and 1000 graphs of the synthetic Kite data set, we constructed a set of query groups with the same number of vertices. The number of vertices vary from 2 vertices to 949 vertices in the real data set and from 30 vertices to 85 vertices in the synthetic data set.

Fig. 7 shows the average runtime performance of *Global*, *GeoLocal*, $GED_{Bipartie}$ and $Beam_{GED}$ in both the real Kite data set (Fig. 7(a)) and the synthetic Kite data set (Fig. 7(b)). 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 set of Kite prototype graphs. This figure clearly shows the interest of using



(a) Real Kite data set



(b) Synthetic Kite data set

Fig. 7. Runtime vs. number of vertices.

Table 6
Classification on the GREC data set.

Methods	*GeoLocal*	$GED_{Bipartite}$	$Beam_{GED}$
GREC data set	96.19%	86.30%	76.70%

the global similarity measure *Global*, which is largely faster than the geometric local similarity measure *GeoLocal*. Fig. 7 also shows that *GeoLocal* is faster compared to $GED_{Bipartie}$ and $Beam_{GED}$. The approaches with which we compare ($GED_{Bipartie}$ and $Beam_{GED}$) are approximatively equivalent with a little difference making $GED_{Bipartie}$ slight faster than $Beam_{GED}$. The runtime performance shown in the figure confirms the theoretical time complexity, which is linear for *Global* and polynomial for *GeoLocal* and $GED_{Bipartie}$. However, *GeoLocal* has a better time complexity, which is $\mathcal{O}((\max(n, m))^3)$ compared to $GED_{Bipartie}$ with $\mathcal{O}((n + m)^3)$, where n and m are the number of vertices of the two compared graphs. Finally, we evaluated the accuracy of the proposed approach by performing classification on the GREC data set. We compare the results obtained by our approach *GeoLocal* with the results obtained by $GED_{Bipartie}$ and $Beam_{GED}$ in [22]. Table 6 depicts the results obtained by our approach *GeoLocal* using the adequate threshold (0.07) and $GED_{Bipartie}$ and $Beam_{GED}$.

We can see that our method *GeoLocal* is more accurate than the two methods with which we compare $GED_{Bipartie}$ and $Beam_{GED}$ on the GREC data set. This confirms that considering the geometric form (angles) has a high added value for object recognition with specific geometric structures. This also shows that our method is extensible on other types of data and proves that the proposed approach is quite general.

6. Conclusions

In this paper, we proposed a graph-based approach for Kite recognition. We presented a complete Kite recognition process in satellite images. We introduced a graph representation of Kites and proposed a novel geometric hierarchical graph matching based on graph edit distance and graph invariants. The proposed method takes into account the geometric form of the graphs in addition to their structures. We also proposed an automatic process for extracting and transforming Kites in satellite images into a set of graphs. Using this process, we construct from real images a benchmark of Kite graphs that can be used by other researchers. Both the theoretical time complexity and the experimental results on real and synthetic Kite data sets confirm the high performance of our approach. Furthermore, the experimentation performed on the GREC data set proves that the proposed approach is extensible and quite general.

Acknowledgments

Many thanks to Remy Crassard, Emmanuelle Vila, Emmanuelle Regagnon, Christine Chataigner and Charles-Edmont Bichot for their expertise.

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

References

[1] O. Barge, J. Brochier, Visible from space, understood during the fieldwork: the example of desert kites in Armenia, in: Cultural heritage and new technologies,Vienne, 2011.
[2] H. Bunke, K. Shearer, A graph distance metric based on the maximal common subgraph, Pattern Recogn. Lett. 19 (3–4) (1998) 255–259, doi:10.1016/S0167-8655(97)00179-7.
[3] D. Conte, P. Foggia, C. Sansone, M. Vento, Thirty years of graph matching in pattern recognition, IJPRAI 18 (3) (2004) 265–298, doi:10.1142/S0218001404003228.
[4] R. Crassard, O. Barge, C.-E. Bichot, J. Brochier, J. Chahoud, M.-L. Chambrade, C. Chataigner, K. Madi, E. RÈ¨gagnon, H. Seba, E. Vila, Addressing the desert kites phenomenon and its global range through a multi-proxy approach, J. Archaeol. Method Theory 22 (4) (2014) 1093–1121, doi:10.1007/s10816-014-9218-7.
[5] A. Desolneux, L. Moisan, J.-M. Morel, Meaningful alignments, Int. J. Comput. Vis. 40 (1) (2000) 7–23.
[6] P. Dosch, E. Valveny, Report on the second symbol recognition contest(2005) 381–397. doi:10.1007/11767978_35.
[7] J.-C. Échallier, F. Braemer, Nature and function of 'desert kites': new datta and hypothesis, Paléorient 21 (1) (1995) 35–63.
[8] P. Foggia, G. Percannella, M. Vento, Graph matching and learning in pattern recognition in the last 10 years, Int. J. Pattern Recognit. Artif. Intell. 28 (01) (2014) 1450001.
[9] X. Gao, B. Xiao, D. Tao, X. Li., A survey of graph edit distance, Pattern Anal. Appl. 13 (2010) 113–129.
[10] R. von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, Lsd: a fast line segment detector with a false detection control, Pattern Anal. Mach. Intell. IEEE Trans. 32 (4) (2010) 722–732.
[11] S. Helms, A. Betts, The desert 'kites' of the badiyat esh-sham and north arabia, Paléorient 13 (1) (1987) 41–67.
[12] J. Illingworth, J. Kittler, A survey of the hough transform, Comput. Vis. Graph. Image Process. 44 (1) (1988) 87–116.
[13] H.W. Kuhn, The Hungarian method for the assignment problem, Naval Res. Logist. Q. 2 (1955) 83–97.
[14] C. Lacoste, X. Descombes, J. Zerubia, Point processes for unsupervised line network extraction in remote sensing, Pattern Anal. Mach. Intell. IEEE Trans. 27 (10) (2005) 1568–1579, doi:10.1109/TPAMI.2005.206.
[15] T. Lee, R.L. Kashyap, C. Chu, Building skeleton models via 3-d medial surface/axis thinning algorithms, CVGIP: Graph. Model Image Process. 56 (6) (1994) 462–478, doi:10.1006/cgip.1994.1042.
[16] D.P. Lopresti, G.T. Wilfong, Comparing Semi-Structured Documents via Graph Probing, in: Workshop on Multimedia Information Systems, 2001, pp. 41–50.
[17] K. Madi, H. Seba, H. Kheddouci, C. Bichot, O. Barge, C. Chataigner, R. Crassard, E. Réganon, E. Vila, Kite recognition by means of graph matching(2015) 118–127. doi:10.1007/978-3-319-18224-7_12.
[18] B. McKay, Practical graph isomorphism, Congr. Numer. 87 (1981) 30–45.
[19] M. Neuhaus, K. Riesen, H. Bunke, Fast suboptimal algorithms for the computation of graph edit distance (2006) 163–172. doi:10.1007/11815921_17.
[20] R. Raveaux, J.-C. Burie, J.-M. Ogier, A graph matching method and a graph matching distance based on subgraph assignments, Pattern Recognit. Lett. 31 (2010) 394–406, doi:10.1016/j.patrec.2009.10.011.
[21] K. Riesen, H. Bunke, IAM graph database repository for graph based pattern recognition and machine learning (2008) 287–297. doi:10.1007/978-3-540-89689-0_33.
[22] K. Riesen, H. Bunke, Approximate graph edit distance computation by means of bipartite graph matching, Image Vis. Comput. 27 (2009) 950–959.
[23] M. Rochery, I.H. Jermyn, J. Zerubia, Higher order active contours, Int. J. Comput. Vis. 69 (1) (2006) 27–42.
[24] A. Sanfeliu, K. Fu, A distance measure between attributed relational graphs for pattern recognition, IEEE Trans. Syst. Man Cybern. (Part B) 13 (3) (1983) 353–363.
[25] S. Sorlin, C. Solnon, J.-M. Jolion, A generic graph distance measure based on multivalent matchings. 52 (2007) 151–181.
[26] J.R. Ullmann, An algorithm for subgraph isomorphism, J. ACM 23 (1) (1976) 31–42.
[27] M. Vento, A long trip in the charming world of graphs for pattern recognition, Pattern Recognit. 48 (2) (2015) 291–301.
[28] Y. Xiao, H. Dong, W. Wu, M. Xiong, W. Wang, B. Shi, Structure-based graph distance measures of high degree of precision, Pattern Recognit. 41 (2008) 3547–3561, doi:10.1016/j.patcog.2008.06.008.
[29] S. Yahiaoui, M. Haddad, B. Effantin, H. Kheddouci, Coloring based approach for matching unrooted and/or unordered trees, Pattern Recognit. Lett. 34 (6) (2013) 686–695, doi:10.1016/j.patrec.2013.01.017.
[30] Z. Zeng, A.K.H. Tung, J. Wang, J. Feng, L. Zhou, Comparing stars: on approximating graph edit distance, Proc. VLDB Endow. PVLDB 2 (1) (2009) 25–36.