A Graph-based approach for Kite recognition

Kamel Madi a,⇑, Hamida Seba a, Hamamache Khedouci a, Olivier Barge b

a Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR 5133, Archéorient, Lyon 69622, France
b CNRS, UMR 5133 Archéorient, Lyon 69622, 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 widespread 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 archaeologists 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
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 experiments 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 experiments 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 experiments 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 [29]. The edit distance between substructures is achieved in $O(n^3)$ 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.

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
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 $\text{length}_{\text{min}}$ and if it has no neighbors according to a minimum neighborhood distance $\text{neighbor}_{\text{min}}$.
- **Merging neighboring segments**: During this step, each pair of segments that are neighbors according to $\text{neighbor}_{\text{min}}$, do not cross each other and have the same angle with the horizontal line with a tolerance angle $\delta$, are merged in one segment. $\text{length}_{\text{min}}$, $\text{neighbor}_{\text{min}}$ 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 “Image” 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 relieved 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. 2.** Illustration of Kite detection.

**Fig. 3.** Examples of Kite graphs in different states of preservation.

---

**Table 1.** Characteristics of the data set.
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.

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 $(R ± ε_{R,i})$ and the angle $(Ang ± ε_{Ang,i})$ 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 descending 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.
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 GlobalGeoLocal and GeoLocalGlobal 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 GlobalGeoLocal or GeoLocalGlobal.

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 G1 and G2 is given by:

\[
Global(G_1, G_2) = w_1 \cdot d_{\text{vertices}}(G_1, G_2) + w_2 \cdot d_{\text{edges}}(G_1, G_2) + w_3 \cdot d_{\text{angles}}(G_1, G_2) + w_4 \cdot d_{\text{convex}}(G_1, G_2)
\]

where \(w_j\) is a weighting coefficient with \(\sum_{j=1}^{4} w_j = 1\). \(d_{\text{vertices}}(G_1, G_2)\) compares the order of the two graphs.

\[
d_{\text{vertices}}(G_1, G_2) = \frac{||V(G_1)|| - ||V(G_2)||}{\max(||V(G_1)||, ||V(G_2)||)}
\]

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

\[
d_{\text{edges}}(G_1, G_2) = \frac{\sum_{i=1}^{\text{edges}G_1} ||E(G_1)|| - ||E(G_2)||}{\max(\sum_{j=1}^{\text{edges}G_1} ||E(G_1)||, \sum_{j=1}^{\text{edges}G_1} ||E(G_2)||)}
\]

\(d_{\text{angles}}(G_1, G_2)\) depends on the angles of the graph and the angle between two edges. It is computed as:

\[
d_{\text{angles}}(G_1, G_2) = \frac{\sum_{i=1}^{\text{edges}G_1} \sum_{j=1}^{\text{edges}G_2} |\text{Angles}_{G_1}(i) - \text{Angles}_{G_2}(j)|}{\max(\sum_{i=1}^{\text{edges}G_1} \sum_{j=1}^{\text{edges}G_1} \text{Angles}_{G_1}(i), \sum_{i=1}^{\text{edges}G_1} \sum_{j=1}^{\text{edges}G_1} \text{Angles}_{G_2}(j))}
\]

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

\(d_{\text{convex}}(G_1, G_2)\) compares the global geometric forms of the two Kites based on the convexity of the angles and the total value of the angles, respectively:

\[
d_{\text{convex}}(G_1, G_2) = \frac{||\text{Angles}_{G_1} - \text{ConvexityTh}||}{\max(\sum_{i=1}^{\text{edges}G_1} \text{Angles}_{G_1}(i), \sum_{i=1}^{\text{edges}G_1} \text{Angles}_{G_2}(j))}
\]

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 $G(V, E)$ 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 $(\langle v_1, v_2, v_3 \rangle \in \mathbb{R}^3, \langle v_1, v_2, v_3 \rangle \in \mathbb{R}^3)$. However, if we consider angles between the horizontal axis $x$, $G_1$ and $G_2$ in (Fig. 5(d, e)) are isomorphic but not geometrical isomorphic: $\alpha_1, \alpha_2 \neq \beta_1, \beta_2, \alpha_2, \beta_2$. In order to compute the geometric local similarity, each vertex $v$ is represented by a signature $(\deg(v), \{\Delta(v) \setminus \deg(v)\}^{1\leq i \leq 3})$, where:

- $\deg(v)$ is the degree of the vertex $v$.
- $\{\Delta(v) \setminus \deg(v)\}^{1\leq i \leq 3}$ are the labels (weights) of the two edges $e_1$ and $e_2$ constituting the angle $\Delta(v)$. $\{\Delta(v) \setminus \deg(v)\}^{1\leq i \leq 3}$ are ranked in descending order.
- The triplets $\{\Delta(v) \setminus \deg(v)\}^{1\leq i \leq 3}$ are ranked according to the angle $\Delta(v)$ in descending order.
- All the vertices are represented by signatures i.e., vectors which have the same size: $\text{size} = 1 + ((\Delta(G_1, G_2) - 1) \times 3)$.
- $\Delta(G_1, G_2)$ 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:

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

The functions $F_i$ are defined as follows:

$$F_i = \frac{|\deg(v_i) - \deg(v_j)|}{\max(\Delta(G_1), \Delta(G_2))}$$

$$F_2 = \frac{|\ell(e_{1,k}) - \ell(e_{2,k})|}{\max(\Delta(G_1), \Delta(G_2)) \times \max(\ell(G_1), \ell(G_2))}$$

$$F_3 = \frac{|\Delta(G_1, G_2)| \cdot \max(\Delta(G_1), \Delta(G_2))}{\sum_{k=1}^{3} \frac{|\deg(v_k) - \Delta(G_1, G_2)|}{\max(\Delta(G_1), \Delta(G_2)) - 1 \times \max(\deg(G_1), \deg(G_2))}}$$

$$d_{GeoLocal}(S_1, S_2) = 1 - \frac{\sum_{i=1}^{3} \max_{m \in M} \sum_{s_1, s_2, s_3} d(s_i, m(s_i))}{\max(\|S_1\|, \|S_2\|)}$$

Computations of $d_{GeoLocal}(S_1, S_2)$ 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 $D_{ij}$ 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 $\varepsilon$. The similarity between an empty signature $\varepsilon$ and a signature $s_j$ 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 $O(n^3)$ 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 $D_{ij} = d(s_i, s_j)$. $\|S_1\| < \|S_2\|$, thus we add an $\varepsilon$ signature to $S_1$ and we complete the matrix $D$ by $d(\varepsilon, s_{2,j}) = 0, 325$. The max sum is 3.25. The normalized dissimilarity is $GeoLocal(S_1, S_2) = \frac{1 - \frac{2.25}{3.25}}{0.1875}$. 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 GlobalGeoLocal and GeoLocalGlobal 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 GlobalGeoLocal 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. GlobalGeoLocal aims to enhance time processing of Kite graphs by first computing invariants on the graphs. Formally, let: G1 and G2 be two graphs, S1 and S2 the set of signatures of G1 and G2 respectively, th ∈ (0, 1] a threshold and d1 = Global(G1, G2).

\[
\text{GlobalGeoLocal}(G_1, G_2) = \begin{cases} 
  d_1, & \text{if } d_1 > \text{th} \\
  \text{GeoLocal}(S_1, S_2), & \text{otherwise} 
\end{cases} \tag{11}
\]

Geometric-Local Global similarity. Like GlobalGeoLocal, the Geometric-Local Global similarity GeoLocalGlobal is a hierarchical similarity measure, which aims to measure the distance between two graphs using firstly GeoLocal, then using Global if necessary. The 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., GlobalGlobalGlobalGlobal aims to improve the graph invariants in the second level by only considering the assigned vertices in the first level using GeoLocal. Formally, let: G1 and G2 be two graphs, G2 is the graph prototype, S1 and S2 the sets of signatures of G1 and G2 respectively, th ∈ (0, 1] a threshold, d2 = GeoLocal(S1, S2) and G1 is the subgraph induced by the vertices of G1 assigned in the first phase using GeoLocal.

\[
\text{GeoLocalGlobal}(G_1, G_2) = \begin{cases} 
  d_2, & \text{if } d_2 > \text{th} \\
  \text{Global}(G_1', G_2), & \text{otherwise} 
\end{cases} \tag{12}
\]

4.4. Reconstruction process

Each connected component from the whole graph representing the query image is compared to the set of prototype graphs (GAntenna, GEnclosure, and four different GKite), using the proposed similarity measure (GeoLocal, Global, GlobalGeoLocal or GeoLocalGlobal). 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, GlobalGeoLocal or GeoLocalGlobal) 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 Simagg of the reconstructed Kite i, is given by:

\[
\text{Simagg}(\text{Kite}_i) = \psi \ast \text{Sim}(E_i) + \sum_{j=1}^{n} \mu \ast \text{Sim}(A_{ij}) \tag{13}
\]

Where: \( \psi + \mu = 1 \). Sim(Ei) is the similarity attributed to the enclosure of the reconstructed Kite i, n is the number of antennas and Sim(Aij) 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 \( O(n^3) \) time, where n is the maximum number of vertices in the two compared graphs. Consequently, the time complexity of GeoLocal is \( O(n^3) \). 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 \( O(n) \), 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:

- **GEDBipartite**: a GED based on a bipartite assignment of vertices and their local structures [22].
- **BeamGED**: 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 \( \alpha_k \) allowing different configurations. The default value is: \( \sum \alpha_k = 1 \). In addition, we defined a threshold in order to improve classification accuracy. The parameters \( \alpha_k \) 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 (GEDBipartite and BeamGED) their default values. However, for each approach we choose the threshold giving the best accuracy. The default parameters of GEDBipartite and BeamGED are: the same cost for vertex/edge deletions/insertions which is 1, the weighting parameters per vertex/edge is the same, the same cost for vertices and edges (vertexCost = edgeCost) and for BeamGED, 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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Reconstruction</th>
<th>State-I</th>
<th>State-II</th>
<th>State-III</th>
<th>State-IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>Yes</td>
<td>93.87%</td>
<td>96%</td>
<td>83%</td>
<td>77%</td>
</tr>
<tr>
<td>GeoLocal</td>
<td>No</td>
<td>89.79%</td>
<td>95%</td>
<td>81%</td>
<td>80%</td>
</tr>
<tr>
<td>GlobalGeoLocal</td>
<td>Yes</td>
<td>93.87%</td>
<td>93%</td>
<td>87%</td>
<td>78%</td>
</tr>
<tr>
<td>GeoLocalGeoLocal</td>
<td>Yes</td>
<td>91.83%</td>
<td>94%</td>
<td>78%</td>
<td>78%</td>
</tr>
<tr>
<td>GeoLocalGlobal</td>
<td>No</td>
<td>87.75%</td>
<td>92%</td>
<td>76%</td>
<td>82%</td>
</tr>
<tr>
<td>GEDbipartite</td>
<td>Yes</td>
<td>93.87%</td>
<td>95%</td>
<td>85%</td>
<td>78%</td>
</tr>
<tr>
<td>BeamGED</td>
<td>No</td>
<td>89.79%</td>
<td>93%</td>
<td>83%</td>
<td>77%</td>
</tr>
</tbody>
</table>

Table 5
Classification on the Kite data set.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Threshold</th>
<th>State I</th>
<th>State II</th>
<th>State III</th>
<th>State IV</th>
<th>Synthetic data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>0.28</td>
<td>93.87%</td>
<td>96%</td>
<td>83%</td>
<td>77%</td>
<td>98%</td>
</tr>
<tr>
<td>GeoLocal</td>
<td>0.28</td>
<td>100%</td>
<td>98%</td>
<td>91%</td>
<td>77%</td>
<td>100%</td>
</tr>
<tr>
<td>GlobalGeoLocal</td>
<td>0.28</td>
<td>91.83%</td>
<td>94%</td>
<td>78%</td>
<td>78%</td>
<td>98%</td>
</tr>
<tr>
<td>GeoLocalGlobal</td>
<td>0.28</td>
<td>93.87%</td>
<td>95%</td>
<td>85%</td>
<td>76%</td>
<td>98%</td>
</tr>
<tr>
<td>GEDbipartite</td>
<td>0.40</td>
<td>36.53%</td>
<td>41%</td>
<td>75%</td>
<td>11%</td>
<td>41.33%</td>
</tr>
<tr>
<td>BeamGED</td>
<td>0.10</td>
<td>20.20%</td>
<td>28%</td>
<td>75%</td>
<td>44.44%</td>
<td>75%</td>
</tr>
</tbody>
</table>

We can see that the four methods are globally more accurate when 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, GlobalGeoLocal and GeoLocalGlobal are more accurate than GEDbipartite and BeamGED 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, GlobalGeoLocal and GeoLocalGlobal at all the levels of the real and the synthetic Kite data set. However, GlobalGeoLocal and GeoLocalGlobal are slightly better in the negative data set (State IV) of the real data set. Although, GeoLocal achieves better classification accuracy compared to GlobalGeoLocal and GeoLocalGlobal. However, use of the hierarchical measures GlobalGeoLocal and GeoLocalGlobal avoids unnecessary comparison in the second level, thus the general runtime on the data set is better. We note also that GeoLocalGlobal achieves better classification accuracy compared to GlobalGeoLocal at all the levels of the real and the synthetic Kite data sets. However, GlobalGeoLocal 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, GEDbipartite and BeamGED 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 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 GEDbipartite and BeamGED. The approaches with which we compare (GEDbipartite and BeamGED) are approximately equivalent with a little difference making GEDbipartite slightly faster than BeamGED. The runtime performance shown in the figure confirms the theoretical time complexity, which is linear for Global and polynomial for GeoLocal and GEDbipartite. However, GeoLocal has a better time complexity, which is \( O((\max(n,m))^3) \) compared to GEDbipartite with \( O(n+m)^3) \), 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 GEDbipartite and BeamGED in [22]. Table 6 depicts the results obtained by our approach GeoLocal using the adequate threshold (0.07) and GEDbipartite and BeamGED.

We can see that our method GeoLocal is more accurate than the two methods with which we compare GEDbipartite and BeamGED 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-Edmond 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