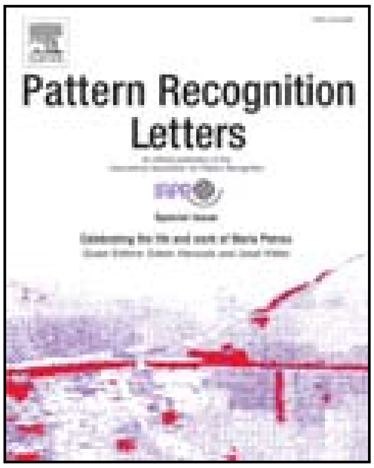
MORENO-GARCÍA, C.F., SERRATOSA, F. and XIAOYI, J. 2020. Correspondence edit distance to obtain a set of weighted means of graph correspondences. *Pattern recognition letters* [online], 134, pages 29-36. Available from: https://doi.org/10.1016/j.patrec.2018.08.027

Correspondence edit distance to obtain a set of weighted means of graph correspondences.

MORENO-GARCÍA, C.F., SERRATOSA, F. and XIAOYI, J.

2020



Correspondence edit distance to obtain a set of weighted means of graph correspondences

Carlos Francisco Moreno-García , Francesc Serratosa ,
Xiaoyi Jiang

PII: S0167-8655(18)30427-6
DOI: https://doi.org/10.1016/j.patrec.2018.08.027
Reference: PATREC 7291

To appear in: *Pattern Recognition Letters*

Received date: 21 September 2017
Revised date: 18 April 2018
Accepted date: 10 August 2018

Please cite this article as: Carlos Francisco Moreno-García , Francesc Serratosa , Xiaoyi Jiang , Correspondence edit distance to obtain a set of weighted means of graph correspondences, *Pattern Recognition Letters* (2018), doi: https://doi.org/10.1016/j.patrec.2018.08.027

Highlights

- Edit distance and graph matching
- Weighted mean and generalized median
- Definition of a distance between graph correspondences
- New algorithm to deduce the set of weighted means of correspondences

Correspondence edit distance to obtain a set of weighted means of graph correspondences

Carlos Francisco Moreno-García[a]*, Francesc Serratosa[b] and Xiaoyi Jiang[c]

[a] *School of Computer Science and Digital Media, The Robert Gordon University, Aberdeen, United Kingdom*
[b] *Department of Computer Science and Mathematics, Universitat Rovirai Virgili, Tarragona, Spain*
[c] *Department of Mathematics and Computer Science, University of Münster, Münster, Germany.*

ABSTRACT

Given a pair of data structures, such as strings, trees, graphs or sets of points, several correspondences (also referred in literature as labellings, matchings or assignments) can be defined between their local parts. The Hamming distance has been largely used to define the dissimilarity of a pair of correspondences between two data structures. Although it has the advantage of being simple in computation, it does not consider the data structures themselves, which the correspondences relate to. In this paper, we extend the definitions of a recently presented distance between correspondences based on the concept of the edit distance, which we called Correspondence edit distance. Moreover, we present an algorithm to compute the set of weighted means between a pair of graph correspondences. Both the Correspondence edit distance and the computation of the set of weighted means are necessary for the calculation of a more representative prototype between a set of correspondences. In the validation section, we show how the use of the Correspondence edit distance increases the quality of the set of weighted means compared to using the Hamming distance.

Keywords: Graph correspondence, Hamming distance, Edit distance, Weighted mean, Generalised median.

* Corresponding author. e-mail: c.moreno-garcia@rgu.ac.uk

1. Introduction

A graph correspondence (or simply a correspondence) is a concept from the graph matching domain which is defined as a bijective function that designates a set of mappings (or assignments) between the nodes and/or edges of a pair of graphs. It can be generated either manually or automatically, in order to find the similarity between the two graphs. In cases where a correspondence is obtained through an automatic method, it is usually deduced through an optimisation process called error-tolerant graph matching. Several graph matching methods have been proposed in recent years [1], [2], [3], [4], [5] to address a variety of problems such as general pattern recognition and image processing [6], text interpretation [7], symbol classification in schematic diagrams [8], chemical and protein compound association, biometrics, malware detection in networks [9], 2D to 3D process plant diagram conversion [10], among others. As a result, it is possible to generate more than one correspondence between a single pair of graphs. In these scenarios, it may be interesting to obtain a single representative prototype correspondence instead.

One of the first concepts proposed in literature as a suitable representative prototype of a set is the generalised median [4], [12], [13] mainly because of its robustness. The generalised median is defined as a representation in the same domain as the data in the set, which achieves the minimum sum of distances to the whole set. Moreover, it was proven in [14] for any metric that, if the number of representations in the set is exactly two, then any weighted mean is also a generalise median. Thus, current methodologies that compute the weighted mean [15], [16], [17] could be adapted to deduce a generalised median. However, for a set of more than two elements, this relation no longer holds therefore, these methods cannot be used.

One fundamental concept bounded to the weighted mean and the generalised median calculation is the distance function used between data structures. So far in the literature, the most commonly used distance between correspondences is the Hamming distance, which measures the number of mappings that are different between two correspondences. This distance has been used either to measure the accuracy of graph matching algorithms [30], [31] or to perform classification [13]. However, the Hamming distance fails to accurately represent the dissimilarity between a pair of correspondences and for this reason, we propose the definition of a new distance.

In the case of strings [18], graphs [19], data clusters [20] and correspondences as well, computing the generalised median of a set of multiple data structures is an NP-complete problem. Thus, some suboptimal methods have been represented to approximate to the generalised median. For instance, an embedding approach has been presented for strings [21], graphs [22] and data clusters [23]. Most recently, a strategy known as the evolutionary method [24] has proven to obtain fair approximations to the generalised median in reasonable time, specifically in the case of strings. Interestingly, the latter method relies on the use of equidistant weighted means to better approximate towards the generalised median, which could be obtained more easily with a robust weighted mean search strategy. Also, a method to deduce the optimal generalised median of a set of correspondences based on the classical Hamming distance was presented in [25]. Remarkably, some consensus methodologies for correspondences based on classical optimisation methods have been presented in [26], [27], [28], [29]. Given two correspondences [26] or several correspondences at once [27], [28], [29], these proposals learn the consensus correspondence, which is a correspondence that minimises some specific cost functions while intending to be close to the mean correspondence.

To apply any of the aforementioned frameworks properly, a more representative distance between correspondences must be defined. To justify this claim, consider the following toy example. Assume that three separate parties (human experts or automatic systems) deduce respectively three correspondences f^1 (blue), f^2 (green) and f^3 (red) between two graphs G and G' as shown in Figure 1 (numbers in nodes represent their attributes). We propose two options to define a distance between these correspondences.

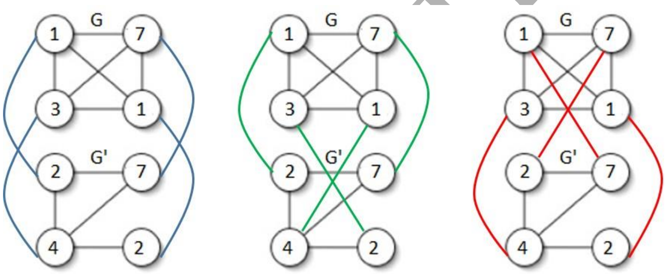


Fig. 1. Three correspondences f^1, f^2 and f^3 between two graphs.

On the one hand, the classical Hamming distance between correspondences. In our example, this distance between f^1 and f^2 is exactly the same as between f^1 and f^3, which turns out to be 2 (there are two different node-to-node mappings). This implies that, given the Hamming distance, both f^2 and f^3 are equally dissimilar with regards to f^1.

On the other hand, we consider the cost of each correspondence as the sum of the Euclidean distance between the attributes of the mapped nodes. Then, we define the difference between correspondences as the difference between their costs. In our example, we have the following costs: $Cost(f^1) = 1 + 0 + 1 + 1 = 3$, $Cost(f^2) = 1 + 0 + 1 + 3 = 5$ and $Cost(f^3) = 6 + 5 + 1 + 1 = 13$. Therefore, we have $|Cost(f^1) - Cost(f^2)| = 2$ but $|Cost(f^1) - Cost(f^3)| = 10$. In this case, f^1 is considered to be more similar to f^2 than f^3. The second option has taken into consideration the node attributes, and thus reflects the difference between the attributes. As we will show in the experimental section, the fact of considering the attributes makes the distance more appropriate for the methods that need a distance between correspondences. A first step towards the calculation of a distance between correspondences which considers the node attributes was defined in [32], which we have called Correspondence Edit Distance (CED).

In this paper, we present a threefold extension of the work in [32]. First, we extend the definition of the CED, considering more information for its computation by means of the local substructure (i.e. Star) of the mapped nodes and verifying that the distance axioms are still applicable to this distance. Secondly, we introduce a new strategy to simultaneously obtain a set of weighted means between a pair of correspondences, which is used to solve a crucial step of the evolutionary method [24] that calculates the generalised mean correspondence. Finally, we present additional experiments to compare the accuracy and applicability of CED with respect to the Hamming distance.

The rest of the paper is structured as follows. The next section introduces the basic definitions. In Section 3, we present the newly proposed distance between a pair of correspondences. In Section 4, we explain our algorithm to deduce the set of weighted mean correspondences. In Section 5, we contrast the new distance against the Hamming distance in the case of finding the set of weighted mean correspondences. Finally, Section 6 is reserved for conclusions and further work.

2. Basic Definitions

Consider a data structure $D = (\Sigma, \gamma)$, where $v_i \in \Sigma$ represents its elements and γ is a function that assigns a set of attributes to each element. This data structure may contain null elements, which have a set of attributes that differentiate them from the rest. From now on, we shall refer to these null elements of D as $v_i \in \hat{\Sigma}$, where $\hat{\Sigma} \subseteq \Sigma$. Note that the concept of data structure may refer to a graph, tree, string or set of elements. In this paper, we consider attributed graphs even though the method we present could be applied to these other data structures, since they are concretisations of attributed graphs.

Given $D = (\Sigma, \gamma)$ and $D' = (\Sigma', \gamma')$ of the same order n (naturally or due to the aforementioned null element presence), we define the set of all possible correspondences T, so that each correspondence in T maps all elements of D to elements of D', $f: \Sigma \to \Sigma'$ in a bijective manner. Thus, Let f^1 and f^2 denote two arbitrarily selected correspondences in T. We deduce how similar these two correspondences are through the Hamming distance HD between f^1 and f^2,

$$HD(f^1, f^2) = \sum_{i=1}^{n} \left(1 - \partial(f^1(v_i), f^2(v_i))\right) \quad (1)$$

Where v_i is an element of D mapped to an element $f(v_i)$ of D' and ∂ is the well-known Kronecker Delta function,

$$\partial(a, b) = \begin{cases} 0 & if\ a \neq b \\ 1 & if\ a = b \end{cases} \quad (2)$$

One of the most widely used frameworks to evaluate the distance between two data structures is the edit distance. This concept has been concretised in the literature as string edit distance [33], tree edit distance [34] and graph edit distance [35], [36], [37] (sets of points are a special case of graphs that do not have edges). The dissimilarity is defined as the minimum amount of required operations that transform one structure into the other. To this end, a number of distortion or edit operations, consisting of the substitution, deletion or insertion of elements are defined. Edit cost functions are introduced to quantitatively evaluate the edit operations. The basic idea is to assign a penalty cost to each edit operation according to the amount of distortion that it introduces in the transformation. Deletion and insertion operations are assignments of a non-null element of the first or second structure to a null element of the second or first structure respectively. Substitutions simply indicate element-to-element mappings. Given two data structures D and D', and a correspondence f between them, the edit cost is

$$EditCost(D, D', f) = \sum_{\substack{v_i \in \Sigma - \hat{\Sigma} \\ f(v_i) \in \Sigma' - \hat{\Sigma}'}} C_s(v_i, f(v_i)) + \\ + \sum_{\substack{v_i \in \Sigma - \hat{\Sigma} \\ f(v_i) \in \hat{\Sigma}'}} C_d(v_i, f(v_i)) + \sum_{\substack{v_i \in \hat{\Sigma} \\ f(v_i) \in \Sigma - \hat{\Sigma}}} C_i(v_i, f(v_i)) \quad (3)$$

where function C_s is a distance between elements, and functions C_d and C_i are the penalty of deleting and inserting elements. Then, the edit distance, ED, is the minimum cost under any bijection in T,

$$ED(G, G') = \min_{f \in T} \{EditCost(D, D', f)\} \quad (4)$$

3. Graph edit distance and local substructures

Let $G = (\Sigma_v, \Sigma_e, \gamma_v, \gamma_e)$ be an attributed graph. $\Sigma_v = \{v_i \mid i = 1, \ldots, n\}$ is the set of vertices and $\Sigma_e = \{e_{i,j} \middle| i, j \in 1, \ldots, k\}$ is the set of edges. Functions $\gamma_v: \Sigma_v \to \Delta_v$ and $\gamma_e: \Sigma_e \to \Delta_e$ assign attribute values in any domain to vertices and edges. $\gamma_v(v_i) = v_i$ and $\gamma_e(e_{i,j}) = e_{i,j}$. The orders of the nodes and edges of the graph are n and k respectively. Similar definitions hold for G'.

The graph edit distance is defined as the minimum amount of required distortion to transform G into G'. To this end, edit operations consisting of the substitution, deletion or insertion, are applied to nodes and edges to calculate a cost $EditCost(G, G', f)$. Given graphs G and G', C_{vs} is a function that represents the cost of substituting the node v_i of G with a node v'_p of G', while C_{es} is a function that represents the cost of substituting the edge $e_{i,j}$ of G with an edge $e'_{p,q}$ of G'.

Moreover, C_{vd} and C_{vi} are the costs of deleting node v_i of G (i.e. mapping to a null node) or inserting the node $f(v_i)$ of G' (i.e. being mapped from a null node). Likewise, C_{ed} and C_{ei} are the costs of assigning edge $e_{i,j}$ of G to a null edge of G' or assigning edge $f(e_{i,j})$ of G' to a null edge of G. This results in Equation 3 applied to graphs, which is formally described as follows,

$$EditCost(G, G', f) =$$

$$\sum_{\substack{v_i \in \Sigma_v - \hat{\Sigma}_v \\ v'_p \in \Sigma'_v - \hat{\Sigma}'_v}} C_{vs}(v_i, v'_p) + \sum_{\substack{e_{i,j} \in \Sigma_e - \hat{\Sigma}_e \\ e'_{p,q} \in \hat{\Sigma}'_e - \hat{\Sigma}'_e}} C_{es}(e_{i,j}, e'_{p,q}) + \sum_{\substack{v_i \in \Sigma_v - \hat{\Sigma}_v \\ v'_p \in \hat{\Sigma}'_v}} C_{vd}(v_i, v'_p) + \sum_{\substack{e_{i,j} \in \Sigma_e - \hat{\Sigma}_e \\ e'_{p,q} \in \hat{\Sigma}'_e}} C_{ed}(e_{i,j}, e'_{p,q}) + \sum_{\substack{v_i \in \hat{\Sigma}'_v \\ v'_p \in \Sigma_v - \hat{\Sigma}_v}} C_{vi}(v_i, v'_p) + \sum_{\substack{e_{i,j} \in \hat{\Sigma}'_e \\ e'_{p,q} \in \Sigma_e - \hat{\Sigma}_e}} C_{ei}(e_{i,j}, e'_{p,q}) \tag{5}$$

As a result, the graph edit distance $EditDist$ is defined as the minimum cost under any bijection in T:

$$EditDist(G, G') = min_{f \in T}\{EditCost(G, G', f)\} \tag{6}$$

To calculate a more accurate dissimilarity measure between two graph correspondences, we have considered the Star local substructure due to its trade-off between simplicity and robustness [38]. A Star within a graph is composed of a node, its connecting edges and incident nodes. These structures are defined as attributed graphs with their specific node and edge structure. More formally, the Star of a node v_i, named S_i, on a graph G, is another graph $S_i = (\Sigma_v^{S_i}, \Sigma_e^{S_i}, \gamma_v^{S_i}, \gamma_e^{S_i})$ composed of $\Sigma_v^{S_i} = \{v_i \cup v_j | e_{i,j} \in \Sigma_e\}$ and $\Sigma_e^{S_i} = \{e_{i,j} | e_{i,j} \in \Sigma_e\}$. Moreover, $\gamma_v^{S_i}(v_j) = \gamma_v(v_j)$, $\forall\, v_j \in \Sigma_v^{S_i}$ and $\gamma_e^{S_i}(e_{i,j}) = \gamma_e(e_{i,j})$, $\forall\, e_{i,j} \in \Sigma_e^{S_i}$.

4. Correspondence edit distance

4.1. Definition

In contrast to the classical Hamming distance, the CED aims to consider the attributes and the structure of the mapped attributed graphs. This property makes the CED more appropriate for the computation of a distance between the weighted mean of two graph correspondences as will be shown later in the present work. Given two attributed graphs G and G', and two correspondences f^1 and f^2 between them, the elements to be considered by the CED are the unary elements (mappings) within f^1 and f^2. This means that it is not our purpose to compute the distance between G and G', but rather the distance between f^1 and f^2. Thus, correspondences f^1 and f^2 are defined as sets of mappings $\mathrm{F}^1 = \{m_1^1, \dots, m_i^1, \dots, m_n^1\}$ and $\mathrm{F}^2 = \{m_1^2, \dots, m_j^2, \dots, m_n^2\}$, where $m_i^1 = (v_i, f^1(v_i))$ and $m_j^2 = (v_j, f^2(v_j))$. Notice that the CED, in contrast to calculating the distance between the graph edit distances associated to each correspondence, can be applied to other scenarios such as manually generated correspondences or correspondences generated for strings, trees, and sets of points, among others.

The key of computing the CED is based on finding a bijective function h (such as f in Equation 3) which generates the minimum cost between the mappings in the sets F^1 and F^2. Similarly to the general case of the edit distance defined in Section 2, these sets of mappings can be enlarged with null mappings, which are included in subsets $\hat{\mathrm{F}}^1$ and $\hat{\mathrm{F}}^2$. Then, we define the CED as,

$$CED(G, G', f^1, f^2) = \min_{h \in H}\{EditCost(G, G', \mathrm{F}^1, \mathrm{F}^2, h)\} \tag{7}$$

and the edit cost is defined as follows,

$$EditCost(G, G', \mathrm{F}^1, \mathrm{F}^2, h) = \sum_{\substack{m_i^1 \in F^1 - \hat{F}^1 \\ m_j^2 \in F^2 - \hat{F}^2}} C_{ms}(m_i^1, m_j^2) + \sum_{\substack{m_i^1 \in F^1 - \hat{F}^1 \\ m_j^2 \in \hat{F}^2}} C_{md}(m_i^1, m_j^2) + \sum_{\substack{m_i^1 \in \hat{\mathrm{F}}^1 \\ m_j^2 \in \mathrm{F}^2 - \hat{\mathrm{F}}^2}} C_{mi}(m_i^1, m_j^2) \quad (8)$$

where C_{ms}, C_{md} and C_{mi} represent the cost of substituting, deleting and inserting a mapping respectively. Notice that in Equation 8, h is defined as the bijective function derived from edit cost calculated between a pair of mappings, in a similar way as f in Equation 3.

The first requirement of the definition of the CED is the concretisation of C_{ms}, C_{md} and C_{mi}, which is

$$\begin{aligned} C_{ms}(m_i^1, m_j^2) &= C_{i,j} + C_{p,q} \\ C_{md}(m_i^1, m_j^2) &= C_{i,\varepsilon} + C_{p,\varepsilon} \\ C_{mi}(m_i^1, m_j^2) &= C_{\varepsilon,j} + C_{\varepsilon,q} \end{aligned} \quad (9)$$

$C_{i,j}$ is the cost of mapping star S_i to S_j and $C_{p,q}$ is the cost of mapping star S'_p to S'_q. Note that in $C_{i,j}$, both mapped stars, S_a and S_b, belong to G. Similarly, in $C_{p,q}$, both mapped stars, S'_p and S'_q, belong to G'. Recall that the computation of Star substitution is done by combining the node and edge substitutions, that is C_{vs} and C_{es}. Figure 2 shows the substitution of mappings m_i^1 with m_j^2.

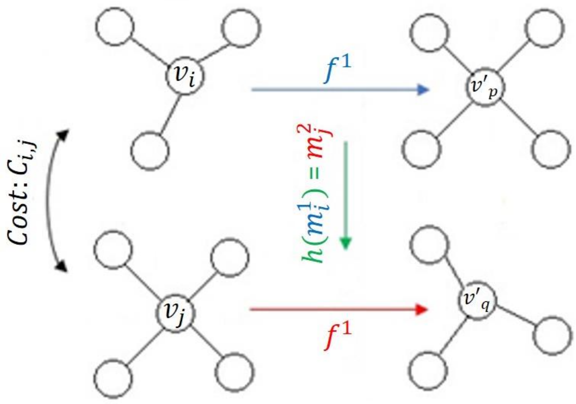


Fig. 2. Substitution of m_a^1 with m_b^2.

In cases where v_i is a null node, $v_i \in \hat{\Sigma}_v$, or v_j is a null node, $v_j \in \hat{\Sigma}_v$ and the other nodes are non null nodes, $v'_p \in \Sigma'_v - \hat{\Sigma}'_v$ or $v'_q \in \Sigma'_v - \hat{\Sigma}'_v$, then the star substitution is converted into a star deletion. Similarly, in cases where v'_i is a null node, $v'_i \in \hat{\Sigma}'_v$, or v'_j is a null node, $v'_j \in \hat{\Sigma}'_v$ and the other nodes are non null nodes, $v_i \in \Sigma_v - \hat{\Sigma}_v$ or $v_j \in \Sigma_v - \hat{\Sigma}_v$, then the star substitution is converted into a star insertion. Besides, $C_{i,\varepsilon}$ and $C_{p,\varepsilon}$ are the costs of deleting the stars of both graphs in terms of C_{vd} and C_{ed}. Similarly, $C_{\varepsilon,j}$ and $C_{\varepsilon,q}$ are the costs of inserting the stars of both graphs in terms of C_{vi} and C_{ei}. Again, it could be that the involved stars are composed of only one null node. In this case, the cost of inserting or deleting a null node is always zero.

Notice that in the first experimentation stages of this work, we computed the distances between correspondences as $|EditCost(G, G', f^1) - EditCost(G, G', f^2)|$. Although it might be the simplest distance between graph correspondences, we realised that it did not properly incorporated the structural difference that related both correspondences. This is because, in some occasions, completely different correspondences return distances close to zero since different correspondences can have equal or similar edit costs.

4.2. Proof of CED as a valid distance

As explained in the previous section, the CED is based on Star edit costs corresponding to each of the mappings that compose each correspondence. Since the Star edit operations are done through additions of node and edge edit operations, the CED is related to the edit cost operations of the GED. In this section, we demonstrate that the distance axioms are applicable to the CED in the same way that they are to the GED [39].

1) *Non-negativity*: $CED(G, G', f^1, f^2) \geq 0$.

For the GED to be a distance, costs C_{vs}, C_{vd}, C_{vi}, C_{es}, C_{ed} and C_{ei} have to be defined non-negative [39]. Thus, since C_{ms}, C_{md} and C_{mi} are defined as additions of these costs, for sure they are non-negative and therefore, the CED is also non-negative.

2) *Identity of indiscernible elements*: $CED(G, G', f^1, f^2) = 0 \Leftrightarrow f^1 = f^2$.

The costs C_{vs} and C_{es} have to be zero if the mapped nodes and edges have the same attributes to be the GED a distance [39]. Then, suppose that h is the identity. In this case, there are not any costs C_{md} or C_{mi} involved in h. Moreover, the whole involved costs C_{md} or C_{mi} becomes zero since they are mapping nodes and edges that have the same attributes. In this case, we have that $EditCost(G, G', \mathrm{F}^1, \mathrm{F}^1, h) = 0$. Due to the Non-negativity of CED, previously defined, for sure that this is the minimum cost and thus, being the identity the optimal correspondence in H.

3) *Symmetry*: $CED(G, G', f^1, f^2) = CED(G, G', f^2, f^1)$

If GED is defined as a distance, costs C_{vs} and C_{vd} have to fulfil the symmetry restriction, as well as the insertion and deletion costs of nodes and edge, more formally, $C_{vi} = C_{vd}$ and $C_{ei} = C_{ed}$ [39]. Thus, $C_{ms}(m_i^1, m_j^2) = C_{ms}(m_j^2, m_i^1)$ and $C_{md}(m_i^1, m_j^2) = C_{mi}(m_j^2, m_i^1)$. Therefore, the CED also complies with the symmetry property, since it is computed as the addition of these costs.

4) *Triangle inequality*:
$CED(G, G', f^1, f^2) \leq CED(G, G', f^1, f^3) + CED(G, G', f^3, f^2)$

For GED to be a distance, costs C_{vs} and C_{es} have to fulfil the triangle inequality [39]. More formally, costs are defined as $C_{vs}(v_a, v_b) \geq C_{vs}(v_a, v_c) + C_{vs}(v_c, v_b)$ and $C_{es}(v_{a,a'}, v_{b,b'}) \geq C_{es}(v_{a,a'}, v_{c,c'}) + C_{es}(v_{c,c'}, v_{b,b'})$. Moreover, $C_{vs} \leq C_{vd} + C_{vi}$ and $C_{es} \geq C_{ed} + C_{ei}$. In this case, for sure that the minimum value achieved by $CED(G, G', f^1, f^2)$ is lower o equal than $CED(G, G', f^1, f^3) + CED(G, G', f^3, f^2)$.

4.3. Other distance frameworks for correspondences

The optimal transport as a general framework for modelling the distance between probability distributions. Several models for its calculation have been defined in literature, such as the Wasserstein distance [40] or the Earth-Mover's distance [41]. The optimal transport deduces the probability distribution distance through considering as a whole the difference between the distributions and the distance between their positions in the sample domain. Intuitively, if each distribution is viewed as an amount of earth, the distance is the minimum cost of turning one pile into the other, which is assumed to be the amount of earth that needs to be moved times the distance it has to be moved.

Note that, the CED could be interpreted as a concrete case of this general framework. This is because the two correspondences to be compared can be seen as different displacements of the earth and the graphs can be seen as the structural and discrete version of the probability densities.

5. Search for a set of weighted means

Given f^1 and f^2 in T and a distance $Dist$ between them, a weighted mean $\bar{f}$ is defined as a correspondence in T that holds the following condition

$$Dist(f^1, f^2) = Dist(f^1, \bar{f}) + Dist(\bar{f}, f^2) \quad (10)$$

Clearly, f^1 and f^2 are also weighted means of themselves. Moreover, the set of weighted mean correspondences is usually not unique depending on the distance used. This aspect, which has an effect on the weighted mean search, is discussed in depth in the experimental validation. Our goal is to present a weighted mean search strategy that is able to find a reasonable amount of weighted mean correspondences simultaneously while reducing the computational cost.

Given two sets of elements of order n, the number of possible bijective correspondences $f \in \mathrm{T}$ between them is $n!$. From this combinatorial space, the search for the weighted mean correspondences can be restricted in two ways. The first way is related to the distance metric used and the second way deals with the exploration space.

Firstly, a distance between correspondences $Dist$ must be defined as an addition of local distances (i.e. subadditivity)

$$Dist(f^1, f^2) = \sum_{i=1}^{n} d(f^1(v_i), f^2(v_i)) \quad (11)$$

where d is a distance measure between mappings. Both the Hamming distance (Equation 1) and the CED (Equation 7) hold this restriction, since both accomplish the triangle inequality axiom and thus.

With regard to the second limitation, we force the weighted mean search space to be within $\mathcal{W} \subseteq \mathrm{T}$. Correspondences in $\mathcal{W}$ have the property that their element-to-element mappings are equal to the element-to-element mappings on either f^1 or f^2, in other words,

$$\bar{f} \in \mathcal{W} \text{ if } \bar{f}(v_i) = f^1(v_i) \text{ or } \bar{f}(v_i) = f^2(v_i);\ \forall\, v_i \in G \quad (12)$$

Theorem 1 demonstrates that all correspondences in $\mathcal{W}$ are indeed weighted means of f^1 and f^2.

Theorem 1. If $\bar{f} \in \mathcal{W}$, then correspondence $\bar{f}$ is a weighted mean of f^1 and f^2.

Proof. Considering the distance definition in Equation 11, we have the following definitions,

$$\begin{aligned} Dist(f^1, \bar{f}) &= \sum_{i=1}^{n} d(f^1(v_i), \bar{f}(v_i)) \\ Dist(f^2, \bar{f}) &= \sum_{i=1}^{n} d(f^2(v_i), \bar{f}(v_i)) \\ Dist(f^1, f^2) &= \sum_{i=1}^{n} d(f^1(v_i), f^2(v_i)) \end{aligned} \tag{13}$$

Therefore, to verify that $\bar{f}$ is a weighted mean correspondence, we need to demonstrate that Equation 14 holds,

$$\sum_{i=1}^{n} d(f^1(v_i), f^2(v_i)) = \\ = \sum_{i=1}^{n} d(f^1(v_i), \bar{f}(v_i)) + \sum_{i=1}^{n} d(f^2(v_i), \bar{f}(v_i)) \tag{14}$$

If the output of f^1 and f^2 is the same, then it is certain that all correspondences in $\mathcal{W}$ contain this mapping and therefore $d(f^1(v_i), f^2(v_i)) = d(\bar{f}(v_i), f^1(v_i)) = d(\bar{f}(v_i), f^2(v_i)) = 0$ and Equation 14 holds. Otherwise, the correspondences in $\mathcal{W}$ must have either option, in other words $\bar{f}(v_i) = f^1(v_i)$ or $\bar{f}(v_i) = f^2(v_i)$. For the first case, $d(\bar{f}(v_i), f^1(v_i)) = 0$, thus $d(f^1(v_i), f^2(v_i)) = 0 + d(f(v_i), f^2(v_i))$ and Equation 14 holds. For the second case, a similar demonstration can be deduced ■

Algorithm 1 returns the set $\mathcal{W}$ of weighted means of f^1 and f^2 by recursively calling function All_Means, which compares the mappings in f^1 and f^2 (if $i \leq n$), keeping the similar ones and searching until all combinations of different partial mappings $partial\_f$ have been computed.

```
Algorithm 1. Weighted Mean Search
Input: f^1, f^2
Output: W
Begin
    W = All_Means(1, ~)    //~: empty correspondence
End Algorithm
Function All_Means (1, partial_f)
Begin
if i > n
    return partial_f;
end if
if f^1(i) = f^2(i)                //two mappings are equal        partial_f(i) = f^1(i)        //mapping is kept
    All_Means(i + 1, partial_f)   //continue search
else                              //two mappings are different
    for (j = f^1(i) ∨ j = f^2(i))  //search for j^th mapping
        if j not in partial_f
            partial_f(i) = j    //mapping is kept
            All_Means(i + 1, partial_f)  //continue search
            partial_f(i) = 0  //clear i^th mapping in partial_f
        end if
    end for
end if
End Function
```

The worst computational cost of this algorithm is $O(2^n)$. This is because the exploration tree has n levels and each level has, at the most, two branches.

6. Validation

6.1. Comparison between distances while computing the set of weighted means

As commented in the introduction, the approximation of the generalised median can be performed through approaches such as the evolutionary method [24]. One of the crucial steps of this method is to obtain a set of equidistant weighted means $\bar{f}_i$ between a pair of graphs. In this section, we analyse the capability of our search strategy to generate said equidistant set of weighted means. This analysis is performed using both the Hamming distance and the CED.

To this end, the following example is provided. Using the first two entries of the "EASTPARK" sequence in the "Tarragona Rotation Zoom" graph repository [42] (shown in Figure 3), we select for each entry the seven strongest SIFT features out of the 50 contained nodes. Afterwards, a graph is constructed using these nodes with the edges conformed through the Delaunay triangulation.

Fig. 3. First two entries of the "EASTPARK" sequence in the "Tarragona Rotation Zoom" graph repository. ●: Seven strongest SIFT features (i.e. nodes) used to generate graphs.

With these two graphs, two node-to-node correspondences

$$f^1 = \{1 \to 6', 2 \to 2', 3 \to 3', 4 \to 4', 5 \to 5', 6 \to 7', 7 \to \Phi', \Phi \to 1'\}$$

and

$$f^2 = \{1 \to \Phi', 2 \to 5', 3 \to 7', 4 \to 4', 5 \to 2', 6 \to 3', 7 \to 1', \Phi \to 7'\}$$

are generated at random. Correspondences f^1 and f^2 have a total of eight mappings, with seven of them being different one from each other. Moreover, both correspondences have null nodes Φ and Φ' which to force them to be bijective.

Algorithm 1 finds twelve correspondences $\mathcal{W} = \bar{f}_1, \dots \bar{f}_{12}$, two of them being the original f_1 and f_2, thus $\bar{f}_1 = f^1$ and $\bar{f}_{12} = f^2$ (an example of how weighted mean correspondences are depicted in a similar case has been previously presented in [32]). Figure 4 shows the distance α_i between each of the twelve weighted means towards f_1, normalised by the distance between f_1 and f_2. That is

$$\alpha_i = \frac{Dist(f_1, \bar{f}_i)}{Dist(f_1, f_2)}, 1 \le i \le 12 \quad (15)$$

In this experiment, we have defined the substitution costs with nodes as the normalised Euclidean distance between the SURF features [43] of the salient points that conform the nodes of the graphs and the insertion and deletion costs of nodes as 0.2. The distances between f_1 and f_2 are: $HD(f_1, f_2) = 6$ and $\mathrm{CED}(f_1, f_2) = 5.85$. As expected, the values obtained by the first and the last weighted means are 0 and 1, respectively.

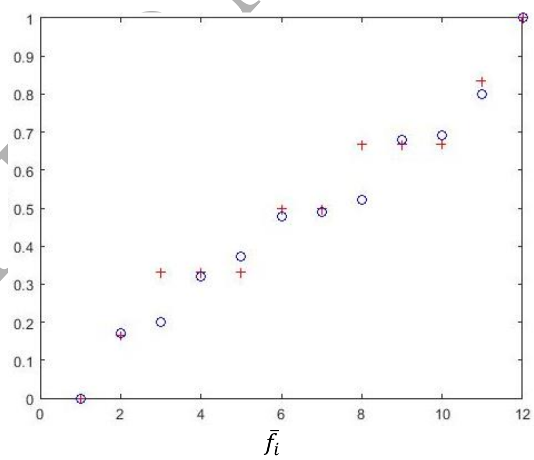


Fig. 4. Normalised distances of the twelve weighted means considering the Hamming distance (+) and CED (**O**). The horizontal axis represents the different weighted means $\bar{f}_i$; $1 \le i \le 12$.

Notice that using the Hamming distance for the computation of the set of weighted means $\mathcal{W}$ achieves seven different values, with $\alpha_3 = \alpha_4 = \alpha_5 = 0.1\bar{6}$ and $\alpha_8 = \alpha_9 = \alpha_{10} = 0.\bar{3}$. Conversely, all weighted means in $\mathcal{W}$ deliver a different distance when the CED is used. The main conclusion drawn from this theoretical validation is that the CED is able to deduce more diverse distances than the Hamming distance, due to the fact that it considers the attributes of the nodes and edges of the graphs being mapped.

Suppose that we are interested in the weighted mean closest to $\alpha_i = 0.25$. Therefore, if the Hamming distance is considered, it is observed from Figure 4 that four correspondences ($\bar{f}_2$, $\bar{f}_3$, $\bar{f}_4$ and $\bar{f}_5$) approach equally to this value, however in the case of the CED, only $\bar{f}_4$ approaches the best. In the case of the Hamming distance, the error committed by $\bar{f}_2$ is $|0.1\bar{6} - 0.25| = 0.083$, which is the same as the one committed by $\bar{f}_3$, $\bar{f}_4$ and $\bar{f}_5$, in other words $|0.\bar{3} - 0.25| = 0.083$. By contrast, in the case of CED, the error committed by $\bar{f}_4$ is $|0.2011 - 0.25| = 0.049$.

Table 1. Mean error and selected correspondences as different number of equidistant weighted means Ω are requested.

Ω	3	4	5	6	7	8	9	10	11	12
HD Mean Error	0	0	0.056	0.050	0.015	0.048	0.048	0.042	0.044	0.045
HD $\mathcal{W}_\Omega$	$\bar{f}_1,\bar{f}_5,\bar{f}_{12}$	$\bar{f}_1,\bar{f}_3,\bar{f}_8,\bar{f}_{12}$	$\bar{f}_1,\bar{f}_2,\bar{f}_5,\bar{f}_8,\bar{f}_{12}$	$\bar{f}_1,\bar{f}_2,\bar{f}_3,\bar{f}_8,\bar{f}_{10},\bar{f}_{12}$	$\bar{f}_1,\bar{f}_2,\bar{f}_3\,\bar{f}_5,\bar{f}_8,\bar{f}_{10},\bar{f}_{12}$	$\bar{f}_1,\bar{f}_2,\bar{f}_3,\bar{f}_5\,\bar{f}_5,\bar{f}_8,\bar{f}_{10},\bar{f}_{12}$	$\bar{f}_1,\bar{f}_2,\bar{f}_2,\bar{f}_3,\bar{f}_5,\bar{f}_8,\bar{f}_8\,\bar{f}_{10},\bar{f}_{12}$	$\bar{f}_1,\bar{f}_2,\bar{f}_2,\bar{f}_3\,\bar{f}_5,\bar{f}_5,\bar{f}_8,\bar{f}_{10},\bar{f}_{10},\bar{f}_{12}$	$\bar{f}_1,\bar{f}_2,\bar{f}_2,\bar{f}_3\,\bar{f}_3,\bar{f}_5,\bar{f}_8\,\bar{f}_8,\bar{f}_{10},\bar{f}_{10},\bar{f}_{12}$	$\bar{f}_1,\bar{f}_2,\bar{f}_2,\bar{f}_3,\bar{f}_3\,\bar{f}_5,\bar{f}_5,\bar{f}_8\,\bar{f}_8,\bar{f}_{10},\bar{f}_{10},\bar{f}_{12}$
CED Mean Error	0.009	0.013	0.036	0.027	0.015	0.041	0.039	0.036	0.035	0.038
CED $\mathcal{W}_\Omega$	$\bar{f}_1,\bar{f}_7,\bar{f}_{12}$	$\bar{f}_1,\bar{f}_4,\bar{f}_9,\bar{f}_{12}$	$\bar{f}_1,\bar{f}_3,\bar{f}_7,\bar{f}_{11},\bar{f}_{12}$	$\bar{f}_1,\bar{f}_3,\bar{f}_5,\bar{f}_8,\bar{f}_{11},\bar{f}_{12}$	$\bar{f}_1,\bar{f}_2,\bar{f}_4\,\bar{f}_7,\bar{f}_9,\bar{f}_{11},\bar{f}_{12}$	$\bar{f}_1,\bar{f}_2,\bar{f}_4,\bar{f}_6\,\bar{f}_8,\bar{f}_{10},\bar{f}_{11},\bar{f}_{12}$	$\bar{f}_1,\bar{f}_2,\bar{f}_3,\bar{f}_5,\bar{f}_7,\bar{f}_9,\bar{f}_{10}\,\bar{f}_{11},\bar{f}_{12}$	$\bar{f}_1,\bar{f}_2,\bar{f}_3,\bar{f}_4\,\bar{f}_6,\bar{f}_8,\bar{f}_9,\bar{f}_{10}\,\bar{f}_{11},\bar{f}_{12}$	$\bar{f}_1,\bar{f}_2,\bar{f}_3,\bar{f}_4\,\bar{f}_5,\bar{f}_7,\bar{f}_8\,\bar{f}_9,\bar{f}_{10},\bar{f}_{11},\bar{f}_{12}$	*All*

The evolutionary method that seeks for a generalised median [24] requires a set of Ω equidistant weighted means $\mathcal{W}_\Omega = \bar{f}_{\alpha_1},\dots,\bar{f}_{\alpha_w},\dots,\bar{f}_{\alpha_\Omega}$ so that $Dist(f^1,\bar{f}_{\alpha_1})$ is close to α_1, $Dist(f^1,\bar{f}_{\alpha_2})$ is close to α_2, and so on, using the following definition:

$$\alpha_w = Dist(f^1,f^2)\cdot\frac{w-1}{\Omega-1};1\le w\le\Omega$$
$$\bar{f}_{\alpha_w} = \underset{\bar{f}\in T}{argmin}\{|Dist(f^1,\bar{f})-\alpha_w|\} \tag{16}$$

Table 1 shows the average error of the weighted mean search algorithm for $3\le\Omega\le 12$ while using the Hamming distance and the CED. The error is computed using Equation 17:

$$Error = \frac{|Dist(f^1,\bar{f}_{\alpha_w})-\alpha_w|}{Dist(f^1,f^2)} \tag{17}$$

Since the errors of the first and last weighted means are always 0, the mean error was computed only with the rest of them. Moreover, it is shown which weighted means have been selected. Note that as Ω increases, the error through the use of the CED is lower than when considering the Hamming distance. Furthermore, it is possible to observe that the weighted means obtained through the Hamming distance are sometimes repeated in the same search, while weighted means obtained through the CED are always different.

6.2. Error and Runtime Analysis using Synthetic Graphs

In a second round of tests, we analyse the average error, using Equation 17, and the runtime of the new weighted mean search strategy given both distances. To do so, we randomly generated four experimental sets $\mathbf{S}^k$. Each set is composed of 60 tuples. Each tuple is composed of a pair of graphs and a pair of correspondences between them, $\{G,G',f^1,f^2\}$, with the particularity of each set being the order of its graphs ($\mathbf{S}^1$ with graph orders [5,10], $\mathbf{S}^2$ with [10,15], $\mathbf{S}^3$ with [15,20] and $\mathbf{S}^4$ with [20,25]). The final experimental dataset is composed of 4×60=240 pairs of graphs and their respective pairs of correspondences.

The experiments were carried out as follows. Per each tuple, a random number α_w was generated so that $0<\alpha_w<Dist(f^1,f^2)$. Then, $\bar{f}_{\alpha_w}$ is computed using Equation 16, and the error is deduced using Equation 17 given both distances. Figure 5 shows the relation between the error committed by the Hamming distance and the CED. We observed that in the majority of the cases, the CED error is lower than the Hamming distance error. Furthermore, the error magnitudes are reduced as graph size increases. Note that, in general, as the order of the graphs increases, the order of set $\mathcal{W}$ also increases.

Figure 6 shows the relationship between the runtime spent to compute our algorithm using the Hamming distance and the CED. The runtime was measured in seconds using a PC with an Intel 3.4 GHz CPU and a Windows 7 operating system. Notice the Hamming is faster in all the experimented cases, and this difference is particularly significant in low and medium size graphs. However, there is a tendency of reducing the gap between both distances as the graph order increases. This is because as the order of the graph increases, the runtime of All_Means, which has a maximum computational cost of $O(2^n)$, becomes larger than the time spent on the distance calculation.

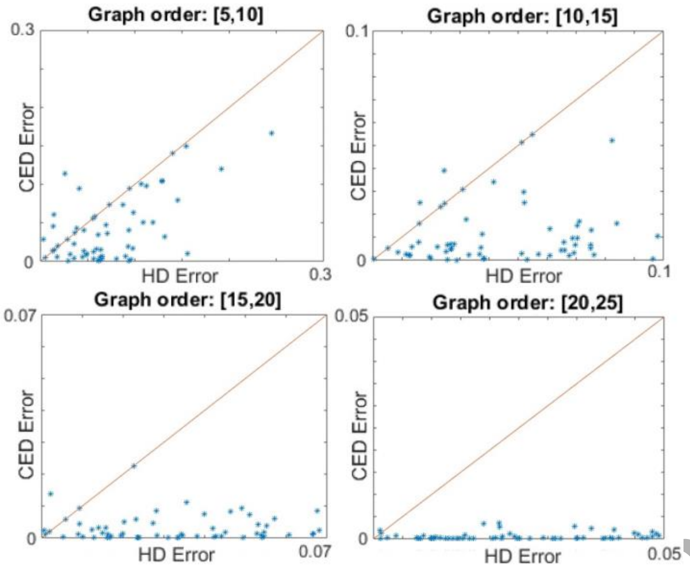


Fig. 5. Mean error committed by the Hamming distance and the CED, given different graph orders (note the scales are different).

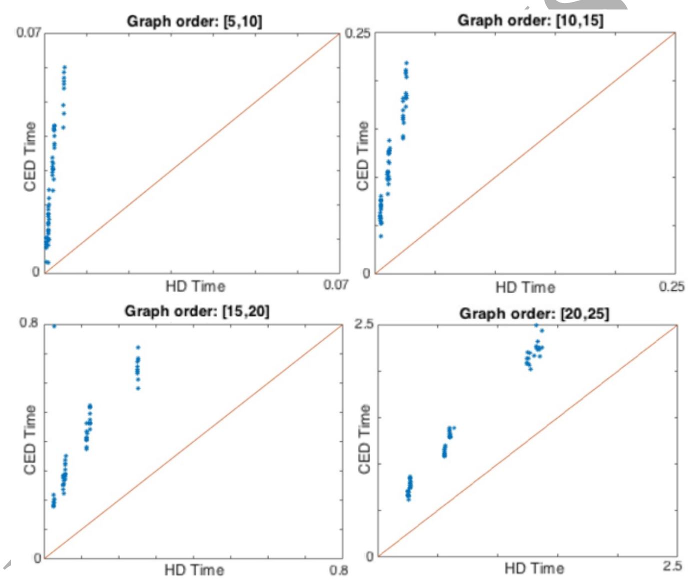


Fig. 6. Runtime of our algorithm while obtaining weighted means using the Hamming distance and the CED (note the scales are different).

From these results, we can conclude that in applications with a low number of nodes, using the Hamming distance could be a better choice since the runtime is reduced and the error gap is not so important. Conversely, when the number of nodes is large or the weighted mean search requires high precision, the CED would be the most suitable option.

Finally, we observed that in some cases, both algorithms produced exactly the same error using both distances (dots in the red diagonal in Figure 5). However, the mean correspondence could be different, since as mentioned earlier, the use of a determined distance has an effect on the mean.

6.3. Evaluation of the Space Search Limitation

Finally, we compared the runtime of our algorithm (which is $O(2^n)$ due to search space reduction) with regard to the runtime of a brute force algorithm (which is $O(n!)$ due to the exploration of the whole search space). Since the complexity of this second algorithm is exponential, experiments were only performed with the first set of graphs (the order of graphs is 5 to 10 nodes). Figure 7 shows the runtime in logarithmic scale. We observed that the runtime of our method is several orders of magnitude faster than the brute force algorithm.

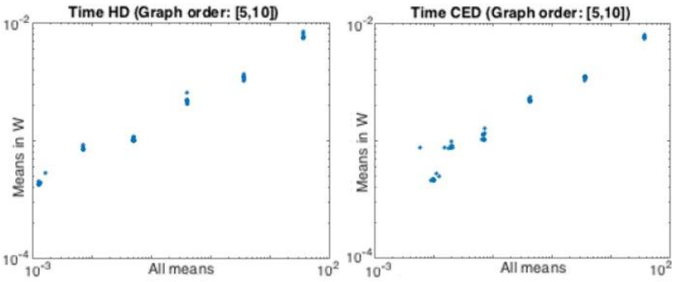


Fig. 7. Runtime of our method compared to a brute force algorithm that searches for weighted means in the whole search space, either using Hamming distance or the CED.

7. Conclusions and Future Work

In this work, we have extended the definition of a distance between a pair of correspondences presented in previous work [32] called CED. This distance aims at finding the dissimilarity between two correspondences which have been computed using different graph matching strategies. Moreover, we have also presented a new algorithm to obtain a set of weighted means given two correspondences. This algorithm has the particularity of considering the CED, which conveys a computational cost of $O(2^n)$, being n the order of the bijective correspondences. Although this complexity is still exponential and thus its application is restricted to the order of the correspondence, we have shown that the obtained dissimilarity measure is more meaningful than any other existing distance between correspondences. We have empirically tested this search strategy for small correspondences resulting in the computation of all possible weighted means, nonetheless it is still not demonstrated that $\mathcal{W}$ contains all possible weighted means for any type of correspondences.

The aim of this distance and the weighted mean calculation algorithm is to compute the generalised median of a set of correspondences through an evolutionary algorithm in a future work. The generalised median of a set of correspondences has been computed using sequential algorithms. However, in the case of sets and strings, the evolutionary algorithm has been shown to produce accurate generalised medians. This is the reason why we wish to explore this strategy applied to finding the generalised median of a set of graph correspondences.

This new distance, even though we computed it using an adaptation of the sub-optimal algorithm called Bipartite graph matching, obtains improved results compared to the classical Hamming distance. This is because, using the CED, we take into consideration not only the correspondences themselves, but also the attributes of the mapped nodes and edges.

We have presented an experimental validation based on two possible distance options: the Hamming distance and the new distance. We have seen that the error produced by the new distance is smaller than the one produced by the Hamming distance in cases where that a set of equidistant weighted means are required. Although the Hamming distance is faster than the CED, this tendency is reduced when the order of the graphs increases. Finally, we have analysed how accurate our algorithm is while deducing the set of weighted mean correspondences since the exploration space has been reduced from $O(n!)$ to $O(2^n)$. We concluded the errors of both methods are similar, but our method is clearly faster than a brute force search.

In a near future, we intend to use the new weighted mean search strategy in order to enhance the evolutionary framework for the generalised median approximation on a set of correspondences. Currently, we are analysing in depth the relevance of CED applied to other methods, for instance, learning the edit costs of the GED. These analyses cannot be included in this paper due to space reasons.

Acknowledgments

This research is supported by the Spanish projects TIN2016-77836-C2-1-R, DPI2016-78957-R and also the European project AEROARMS, H2020-ICT-2014-1-644271.

References

1. D. Conte, P. Foggia, C. Sansone, and M. Vento, “Thirty years of graph matching,” *International Journal of Pattern Recognition and Artificial Intelligence.*, vol. 18, no. 3, pp. 265–298, 2004.
2. P. Foggia, G. Percannella, and M. Vento, “Graph matching and learning in pattern recognition on the last ten years,” *International Journal of Pattern Recognition and Artificial Intelligence.*, vol. 28, no. 1, 1450001 (40 pages), 2014.
3. M. Vento, “A long trip in the charming world of graphs for pattern recognition,” *Pattern Recognition*, vol. 48, no. 2, pp. 291–301, 2015.
4. F. Emmert-Streib, M. Dehmer, and Y. Shi, “Fifty years of graph matching, network alignment and network comparison,” *Information Science (NY)*, vol. 346, pp. 180–197, 2016.
5. J. Yan, X.-C. Yin, W. Lin, C. Deng, H. Zha, and X. Yang, “A Short Survey of Recent Advances in Graph Matching,” *Proceedings of the 2016 ACM International Conference on Multimedia Retrieval (ICMR ‘16)*, pp. 167–174, 2016.
6. D. Conte, P. Foggia, C. Sansone, and M. Vento, “Graph matching applications in pattern recognition and image processing,” *International Conference on Image Processing (ICIP ‘16)*, vol. 3, no. 2, pp. 21–24, 2003.
7. M. I. Bhat and B. Sharada, “Recognition of Handwritten Devanagiri Numerals by Graph Representation and SVM,” *International Conference on Advances in Computing, Communications and Informatics*, 2016, pp. 1930–1935.
8. C. Howie, J. Kunz, T. Binford, T. Chen, and K. H. Law, “Computer Interpretation of Process and Instrumentation Drawings,” *Advances in Engineering Software*, vol. 29, no. 7–9, pp. 563–570, 1998.

9. M. Stauffer, T. Tschachtli, A. Fischer, and K. Riesen, “A Survey on Applications of Bipartite Graph Edit Distance,” *Graph-Based Representations in Pattern Recognition (GbR-PR)*, 2017, vol. 10310, pp. 242–252.
10. R. Wen, W. Tang, and Z. Su, “A 2D engineering drawing and 3D model matching algorithm for process plant,” *Graphical Models*, vol. 0, pp. 1–15, 2017.
11. X. Jiang, A. Münger, and H. Bunke, “Synthesis of representative graphical symbols by computing generalized median graph,” *Graphics Recognition (GREC)*, vol. 1941, pp. 183–192, 2000.
12. X. Jiang and H. Bunke, “Learning by generalized median concept,” *Pattern Recognition and Machine Vision*, River Publishers (Aalborg), pp. 1–16, 2010.
13. R. Chaieb, K. Kalti, M. M. Luqman, M. Coustaty, J. M. Ogier, and N. E. Amara, “Fuzzy generalized median graphs computation: application to content-based document retrieval,” *Pattern Recognition*, vol. 72, pp. 266–284, 2017.
14. T. Kohonen, “Median Strings,” *Pattern Recognition Letters*, vol. 3, pp. 309-3013, 1985.
15. H. Bunke, X. Jiang, K. Abegglen, and A. Kandel, “On the weighted mean of a pair of strings,” *Pattern Analysis and Applications*, vol. 5, no. 1, pp. 23–30, 2002.
16. H. Bunke and S. Günter, “Weighted mean of a pair of graphs,” *Computing*, vol. 67, no. 3, pp. 209–224, 2001.
17. L. Franek, X. Jiang, and C. He, “Weighted mean of a pair of clusterings,” *Pattern Analysis and Applications*, vol. 17, no. 1, pp. 153–166, 2014.
18. X. Jiang, K. Abegglen, H. Bunke, and J. Csirik, “Dynamic computation of generalised median strings,” *Pattern Analysis and Applications*, vol. 6, no. 3, pp. 185–193, 2003.
19. X. Jiang, A. Münger, and H. Bunke, “On median graphs: properties, algorithms, and applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1144–1151, 2001.
20. S. Vega-Pons and J. Ruiz-Shulcloper, “A survey of clustering ensemble algorithms,” *International Journal of Pattern Recognition and Artificial. Intelligence,* vol. 25, no. 3, pp. 337–372, 2011.
21. X. Jiang, J. Wentker, and M. Ferrer, “Generalized median string computation by means of string embedding in vector spaces,” *Pattern Recognition Letters*, vol. 33, no. 7, pp. 842–852, 2012.
22. M. Ferrer, E. Valveny, F. Serratosa, K. Riesen, and H. Bunke, “Generalized median graph computation by means of graph embedding in vector spaces,” *Pattern Recognition*, vol. 43, no. 4, pp. 1642–1655, 2010.
23. L. Franek and X. Jiang, “Ensemble clustering by means of clustering embedding in vector spaces,” *Pattern Recognition*, vol. 47, no. 2, pp. 833–842, 2014.
24. L. Franek and X. Jiang, “Evolutionary weighted mean based framework for generalized median computation with application to strings,” *Structural, Syntactic and Statistical Pattern Recognition (S+SSPR)*, LNCS vol. 7626, pp. 70–78, 2012.
25. C. F. Moreno-García, F. Serratosa, and X. Cortés, “Generalised median of a set of correspondences based on the hamming distance,” *Structural, Syntactic, and Statistical Pattern Recognition (S+SSPR)*, LNCS vol. 4109, pp. 507–518, 2016.
26. C. F. Moreno-García and F. Serratosa, “Correspondence consensus of two sets of correspondences through optimisation functions,” *Pattern Analysis and Applications*, vol. 20, no. 1, pp. 201–213, 2017.
27. C. F. Moreno-García and F. Serratosa, “Online learning the consensus of multiple correspondences between sets,” *Knowledge-Based Systems*, vol. 90, pp. 49–57, 2015.
28. C. F. Moreno-García and F. Serratosa, “Consensus of multiple correspondences between sets of elements,” *Computer Vision and Image Understanding*, vol. 142, pp. 50–64, 2016.
29. C. F. Moreno-García and F. Serratosa, “Obtaining the consensus of multiple correspondences between graphs through online learning,” *Pattern Recognition Letters*, vol. 87, no. 1, pp. 79–86, 2017.
30. T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola, “Learning graph matching,” *IEEE Transactions on Pattern Analysis and Machine Intell*igence, vol. 31, no. 6, pp. 1048–1058, 2009.
31. F. Zhou and F. De la Torre, “Factorized graph matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1774-1789, 2016.
32. C. F. Moreno-García, F. Serratosa, and X. Jiang, “An edit distance between graph correspondences,” *Graph-Based Representations in Pattern Recognition (GbR-PR)*, vol. 10310, pp. 232–241, 2017.
33. R. A. Wagner and M. J. Fischer, “The string-to-string correction problem,” *Journal of the Association for Computer Machinery*, vol. 21, no. 1, pp. 168–173, 1974.
34. P. Bille, “A survey on tree edit distance and related problems,” *Theoretical Computing Science*, vol. 337, no. 9, pp. 217–239, 2005.
35. A. Sanfeliu and K. S. Fu, “A distance measure between attributed relational graphs for pattern recognition,” *IEEE Transactions on Systems, Man and Cyber*netics, vol. 13, no. 3, pp. 353–362, 1983.
36. X. Gao, B. Xiao, D. Tao, and X. Li, “A survey of graph edit distance,” *Pattern Analysis and. Applications*, vol. 13, no. 1, pp. 113–129, 2010.
37. A. Solé-Ribalta, F. Serratosa, and A. Sanfeliu, “On the graph edit distance cost: properties and applications,” *International Journal of Pattern Recognition and Artificial. Intelligence*, vol. 26, no. 5, 1260004 (24 pages), 2012.
38. F. Serratosa and X. Cortés, “Graph Edit Distance: Moving from global to local structure to solve the graph-matching problem,” *Pattern Recognition Letters*, vol. 65, pp. 204–210, 2015.
39. F. Serratosa, X. Cortés and C. Moreno, “Graph Edit Distance or Graph Edit Pseudo-Distance?,” *Structural, Syntactic and Statistical Pattern Recognition (S+SSPR)*, LNCS vol. 10029, pp. 530–540, 2016.
40. L. Rüschendorf, “Wasserstein metric,” *Encyclopaedia of Mathematics*, Springer Science+Business Media B.V. / Kluwer Academic Publishers, 2001.
41. F. Serratosa and G. Sanromà, “A Fast Approximation of the Earth-Movers Distance between Multi-Dimensional Histograms,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 8, pp. 1539 -1558, 2008.
42. C. F. Moreno-García, X. Cortés, and F. Serratosa, “A graph repository for learning error-tolerant graph matching,” *Structural, Syntactic and Statistical Pattern Recognition (S+SSPR)*, LNCS vol. 10029, pp. 519–529, 2016.
43. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.