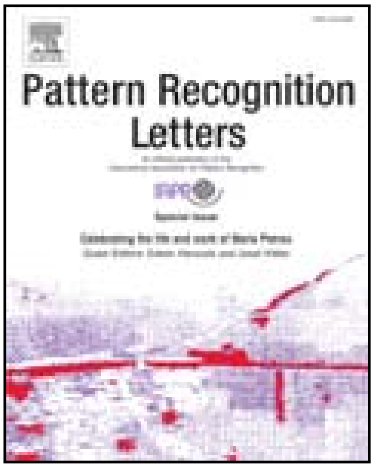
MORENO-GARCÍA, C.F. and SERRATOSA, F. 2019. Generalised median of graph correspondences. *Pattern recognition letters* [online], 125, pages 389-395. Available from: https://doi.org/10.1016/j.patrec.2019.05.015

Generalised median of graph correspondences.

MORENO-GARCÍA, C.F., SERRATOSA, F.

2019



Generalised median of graph correspondences

Carlos Francisco Moreno-García, Francesc Serratosa

PII: S0167-8655(19)30164-3
DOI: https://doi.org/10.1016/j.patrec.2019.05.015
Reference: PATREC 7526

To appear in: *Pattern Recognition Letters*

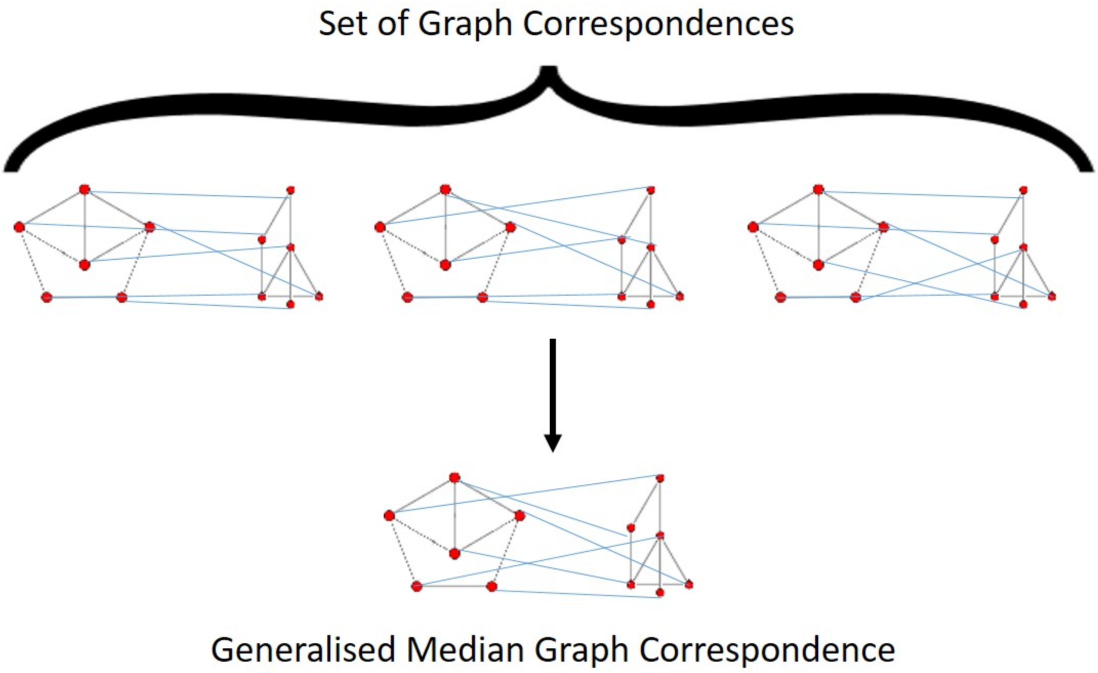
Received date: 16 February 2019
Revised date: 19 April 2019
Accepted date: 19 May 2019

Please cite this article as: Carlos Francisco Moreno-García, Francesc Serratosa, Generalised median of graph correspondences, *Pattern Recognition Letters* (2019), doi: https://doi.org/10.1016/j.patrec.2019.05.015

Highlights

- We propose a method to obtain the generalised median between graph correspondences.
- By considering graph attributes, more useful medians are obtained.
- Experiments show effectiveness in comparison to state-of-the-art methods.

Graphical Abstract



ELSEVIER

Pattern Recognition Letters
journal homepage: www.elsevier.com

Generalised median of graph correspondences

Carlos Francisco Moreno-García[a,**], Francesc Serratosa[b]

[a]The Robert Gordon University, Garthdee Road, Aberdeen, Scotland, UK
[b]Universitat Rovira i Virgili, Av. Paisos Catalans 26, Tarragona, Catalonia, Spain

ABSTRACT

A graph correspondence is defined as a function that maps the elements of two attributed graphs. Due to the increasing availability of methods to perform graph matching, numerous graph correspondences can be deducted for a pair of attributed graphs. To obtain a representative prototype for a set of data structures, the concept of the median has been largely employed, as it has proven to deliver a robust sample. Nonetheless, the calculation of the exact (or generalised) median is known to be an *NP*-complete problem for most domains. In this paper, we present a method based on an optimisation function to calculate the generalised median graph correspondence. This method makes use of the Correspondence Edit Distance, which is a metric that considers the attributes and the local structures of the graphs to obtain more interesting and meaningful results. Experimental validation shows that this approach is capable of obtaining the generalised median in a comparable runtime with respect to state-of-the-art methods on artificial data, while maintaining the success rate for a real-application case.

1. Introduction

A correspondence is defined as the result of a bijective function which designates a set of one-to-one mappings between elements representing the local information of two structures i.e. sets of points, strings, trees, graphs or data clusters. Each element (a point for sets of points; a character for strings, or a node and its edges for trees or graphs) has a set of attributes that contain specific information. Correspondences are usually generated, either manually or automatically, with the purpose of finding the similarity or a distance between two structures. In the case that correspondences are deduced through an automatic method, this is most commonly done through an optimisation process called matching. Several matching methods have been proposed for the aforementioned structures, compiled by Zitová and Flusser (2003) for sets of points, Navarro (2001) for strings, and Vento (2015) for trees and graphs.

Correspondences are used for a number of different purposes. For instance, Caetano et al. (2009) and Zhou and De La Torre (2016) used them to measure the accuracy of different graph matching algorithms. Cortés et al. (2013) considered ground-truth correspondences to improve the quality of other correspondences. In addition, Cortés and Serratosa (2016) learned the edit costs to implement them in their matching algorithms. Cortés et al. (2016) applied correspondences between images to estimate the pose of a fleet of robots. Moreover, Moreno-García et al. (2016a) put together a repository of graphs and correspondences to test classification methods. Interestingly, work by Moreno-García and Serratosa (2017a), Moreno-García and Serratosa (2016), Moreno-García and Serratosa (2015) and Moreno-García and Serratosa (2017b) proposed the calculation of the consensus between a set of correspondences based on optimisation functions and other parameters. All of these approaches use the classical Hamming distance (HD) to calculate the dissimilarity between a pair of correspondences. Nonetheless, recent work presented by Moreno-García et al. (2017) showed that this distance does not always reflect the true dissimilarity between a pair of correspondences, and thus, a new distance called Correspondence Edit Distance (CED) was defined by Moreno-García et al. (2018).

Given the vast availability of matching algorithms such as the ones presented by Xiao et al. (2009a), Xiao et al. (2009b) and Carcassoni and Hancock (2003), numerous correspondences can be produced and thus, the aforementioned tasks scale in

**Corresponding author: Tel.: +44-1224-262790
e-mail: author@author.com (Carlos Francisco Moreno-García)

complexity and uncertainty. Therefore, the use of concepts to find a representative prototype correspondence of a set of objects comes into use. This is the case of the median, roughly defined as a sample that achieves the minimum sum of distances (SOD) to all members of such set. Authors such as Jiang and Bunke (2010) have considered this as a suitable representative prototype of a set of structures given its robustness and its applicability to several domains, such as representation of symbols by Jiang et al. (2000), digit recognition by Jiang et al. (2003), document retrieval by Chaieb et al. (2017), amongst others described by Jiang et al. (2001). Computing the median is an *NP*-complete problem, as shown by Bunke et al. (2002) for strings, Bunke and Günter (2001) for graphs, and Franek et al. (2014) for data clusters. Thus, some sub-optimal methods have been presented to calculate an approximation to the median. For instance, an embedding approach has been presented for computing the median string in Jiang et al. (2012), the median of graphs in Ferrer et al. (2010) and the median of data clusters in Franek and Jiang (2014). In addition, a strategy known as the *evolutionary method* was introduced by Franek and Jiang (2012) to compute the median string, obtaining a fair approximation in reasonable runtime.

In terms of the median graph, Ferrer et al. (2010) presented some strategies which need to compute a common correspondence (Solé-Ribalta and Serratosa (2013, 2011)) a consensus correspondence (Moreno-García and Serratosa (2017a)) or a median correspondence (Moreno-García et al. (2018)) between the graphs in the set. For this reason, a method to compute the exact median, also known as generalised median (GM), of these graph correspondences (not to be confused with the median graph) was presented in Moreno-García et al. (2016b) with respect to the HD and using optimisation functions.

The aim of this paper is to present a method that obtains the GM correspondence for a set of attributed-graph correspondences based on the CED instead of the HD, as done by Moreno-García et al. (2016b). The main difference between using the CED instead of the HD is that the attributes and structures of graphs are taken into consideration, thus achieving more useful correspondences for higher-level methods or applications. A preliminary version of this paper was presented by Moreno-García and Serratosa (2018). In the current version, we have added much more explanation of the methodology, we have tested the use of the local sub-structure within the CED (and thus within the median calculation), as well as an experimental validation using public databases.

The computation of the generalised median of a set of graph correspondences cannot be directly applied to solve a classification problem. The deduced median graph correspondence can be used to find better graph prototypes. Then, with these prototypes, higher level tasks could be performed, such as clustering or classification (in the structural pattern recognition field) or image segmentation (in the computer vision field). These other tasks are out of the scope of this paper due to space reduction and that they are application dependent.

The rest of the paper is structured as follows. Section 2 establishes the basic definitions. Afterwards, in Section 3 we present the method to calculate the GM based on the HD. Then, in Section 4 we present our method to compute the median based on the CED. Section 5 provides experimental validation of the framework. Finally, Section 6 is reserved for the conclusions and further work.

2. Basic Definitions

2.1. Distance between structures

Consider a structure $G = (\Sigma, \mu)$, where $v_i \in \Sigma$ denotes the elements (i.e. local information) and μ is a function that assigns a set of attributes to each element. This structure may contain null elements which have a set of attributes that differentiate them from the rest. Moreover, given $G = (\Sigma, \mu)$ and $G' = (\Sigma', \mu')$ of the same order N (naturally or due to the aforementioned null element presence), we define the set of all possible correspondences T, such that each correspondence in T maps all elements of G to elements of G', $f : \Sigma \to \Sigma'$ in a bijective manner.

One of the most widely used frameworks to calculate the distance between structures is the edit distance. It has been presented by Wagner and Fischer (1974) for strings, Bille (2005) for trees and Sanfeliu and Fu (1983), Gao et al. (2010), Solé-Ribalta et al. (2012) and Serratosa (2019) for graphs. The edit distance is defined as the minimum amount of required operations that transform one object into the other. To this end, several distortions or edit operations, consisting of insertion, deletion and substitution 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 considering the amount of distortion that it introduces in the transformation. Substitutions simply indicate element-to-element mappings. Deletions are transformed to assignments of a non-null element of the first structure to a null element of the second structure. Insertions are transformed to assignments of a non-null element of the second structure to a null element of the first structure. Given G and G' and a correspondence f between them, the edit distance is obtained as follows:

$$EditCost(G, G', f) = \sum_{v_i \in \Sigma \ v'_a \in \Sigma'} d(v_i, v'_a) \tag{1}$$

where $f(v_i) = v'_a$, function d is a distance between the mapped elements, which is application dependent and also depends on whether the elements are non-null or null. Thus, the edit distance ED is defined as the minimum cost under any bijection in T:

$$ED(G, G') = \min_{f \in T} EditCost(G, G', f) \tag{2}$$

2.2. Mean, weighted mean and median

In its most general form, the mean of two structures G and G' is defined as a structure $\bar{G}$ such that:

$$\begin{aligned} Dist(G, \bar{G}) &= Dist(\bar{G}, G') \\ Dist(G, G') &= Dist(G, \bar{G}) + Dist(\bar{G}, G') \end{aligned} \tag{3}$$

where *Dist* is any distance metric defined on the domain of these structures. Moreover, the concept of weighted mean is used to gauge the importance or the contribution of the involved structures in the mean calculation. The weighted mean between two structures is defined as:

$$Dist(G, \bar{G}) = \lambda \quad \text{and} \quad Dist(G, G') = \lambda + Dist(\bar{G}, G') \tag{4}$$

being λ is a constant that controls the contribution of the structures and holds $0 \leq \lambda \leq Dist(G, G')$. G and G' satisfy this condition, and therefore are also weighted means of themselves.

From the definition of the median, two different approaches are identified: the set median (SM) and the GM. The first one is defined as the structure within the set which has the minimum SOD. Conversely, the GM is the structure out of any element in the set which obtains the minimum SOD.

2.3. Distance between correspondences

Given two attributed structures G and G' such as graphs, and two correspondences f^1 and f^2 between them, we proceed to define the HD and the CED.

2.3.1. Hamming distance

The HD is defined as:

$$HD(f^1, f^2) = \sum_{i=1}^{N}(1 - \delta(v'_a, v'_b)) \tag{5}$$

where a and b such that $f^1(v_i) = v'_a$ and $f^2(v_i) = v'_b$, and δ being the Kronecker Delta function:

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

2.3.2. Correspondence edit distance

The CED is defined in a similar way to Equations 1 and 2, but the elements used by CED are the node-to-node mappings within f^1 and f^2. More formally, correspondences f^1 and f^2 are defined as sets of mappings $f^1 = m^1_1, ..., m^1_i, ..., m^1_N$ and $f^2 = m^2_1, ..., m^2_j, ..., m^2_N$, where m^1_i and m^2_j are the two-element vectors $m^1_i = (v_i, f^1(v_i))$ and $m^2_j = (v_j, f^2(v_j))$.

Then, we define:

$$CED(f^1, f^2) = \min_{h \in H} Corr\_EditCost(f^1, f^2, h) \tag{7}$$

where

$$Corr\_EditCost(f^1, f^2, h) = \sum_{m^1_i \in M^1 \ m^2_j \in M^2} d(m^1_i, m^2_j) \tag{8}$$

being M^1 and M^2 the sets of all possible node-to-node mappings. Moreover, bijection h maps the mapping m^1_i in f^1 into the mapping m^2_j in f^2.

Moreno-García et al. (2018) define the distance between mappings $d(m^1_i, m^2_j)$ as,

$$d(m^1_i, m^2_j) = dn(v_i, v_j) + dn(f^1(v_i), f^2(v_j)) \tag{9}$$

where dn is a distance between the local parts of the structures, which is application dependent. In this paper, we have used the star structure, which is composed of a central node and their adjacent edges and neighbouring nodes, in contrast to Moreno-García and Serratosa (2018), where only the points were considered without edges.

3. Generalised Median Correspondence based on the Hamming Distance

This section summarises the method presented in Moreno-García et al. (2016b) to calculate the GM, $\hat{f}$, of a set of correspondences, $f^1, ..., f^p, ..., f^n$, based on the HD for the case of correspondences, f^p, between sets of points.

The first step of this method is to convert the set of correspondences $f^1, ..., f^p, ..., f^n$ into correspondence matrices (double stochastic) $F^1, ..., F^p, ..., F^n$ as follows,

$$F^p[i, b] = \begin{cases} 1 & \text{if } f^p(v_i) = v'_b \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Afterwards, a linear solver such as the Hungarian method presented by Kuhn (1955), the Munkres algorithm by Munkres (1957), or the Jonker-Volgenant solver by Jonker and Volgenant (1987), is applied to the sum of these matrices as follows:

$$\hat{f} = argmin \sum_{p=1}^{n}(C \circ F^p[i, b]) \tag{11}$$

where $[i, b]$ is a specific cell and C is the following matrix:

$$C = \sum_{p=1}^{n}(\mathbf{1} - F^p[i, b]) \tag{12}$$

The idea is that by introducing a value of either 0 or a 1 in the correspondence matrix, the HD is not only being considered by the method, but also minimised.

4. Generalised Median Correspondence based on the Correspondence Edit Distance

This section presents the main contribution of this paper, which is a method to calculate the GM $\hat{f}$ of a set of correspondences, $f^1, ..., f^p, ..., f^n$, this time based on the CED for the case of correspondences which map attributed graphs. As commented previously, the median calculation had only been modelled through the HD by (Moreno-García et al. (2016b)) and partially using CED by Moreno-García et al. (2018), but without considering the local structure of the nodes being mapped. Through the CED and the consideration of the local substructures of both graphs, it is clear that much more interesting and useful medians could be generated from an application point of view. As a result, this new method becomes the generalisation of the one presented by Moreno-García et al. (2018), in which the structural and semantic information of the graphs is considered.

First, we define the new equation based on the one used to compute the GM correspondence using the HD. In this case, we need to redefine matrix C in Equation 12 as follows,

$$C = \sum_{p=1}^{n} B^p[i, b] \tag{13}$$

Notice that instead of minimising $(\mathbf{1} - F^p)$, the goal is to minimise B^p, which is a matrix where the distance between local substructures is added to the corresponding cell, in contrast to F^p, which is a matrix where a value of 0 or 1 is inserted if a mapping is similar or different, respectively. The next step is to define the function used to fill matrix B^p, which is,

$$B^p[i, b] = d(m, m_i^p) + d(m, m_j^p) \tag{14}$$

being m the mapping $m = \{v_i, v'_b\}$. Moreover, $m_i^p = (v_i, f^p(v_i))$, $m_j^p = (v_j, f^p(v_j))$ and $f^p(v_j)) = v'_b$. It is important to note that the first part of the expression is similar to how the bijective function h is calculated in Equation 7, in the sense that it only computes the distances between mappings that have the same elements on the output structure G. According to the distance between node-to-node mappings used, null elements (and thus null mappings) are considered accordingly.

Moreover, $B^p[i, b]$ is defined as the sum of the distances between any supposed mapping from v_i to v'_b and the mappings imposed by correspondence f^p that relates elements v_i and v_j to elements v'_a and v'_b, respectively. As the distances between these two mappings increase, so it does the value of $B^p[i, b]$. Figure 1 graphically shows the computation of $B^p[i, b]$.

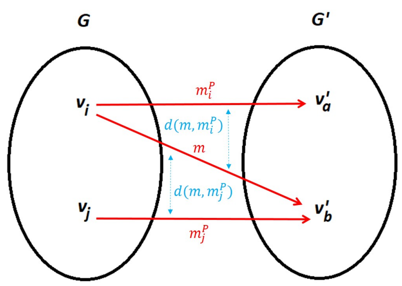


Fig. 1. Computing $B^p[i, b]$ as a sum of the distances $d(m, m_i^p)$ and $d(m, m_j^p)$

In this case, the calculation of a value in $B^p[i, b]$ also yields two cases as in Equation 10, but this time meeting the following two criteria,

$$B^p[i, b] = \begin{cases} 0 & \text{if } m = m_i^p \vee m = m_j^p \\ > 0 & \text{otherwise} \end{cases} \tag{15}$$

As a result, matrix C in Equation 13 is a generalisation of matrix C in Equation 12. Finally, matrix C is minimised in the same way as in Equation 11.

5. Experimental Validation

We have split the validation into two subsections. The first one shows an evaluation on artificial data, while the second one has been designed using a real case of image matching and graph correspondences. All experiments were implemented using Matlab 2017a on a Windows 10 machine with 16 GB RAM and a 2.7 GHz processor.

5.1. Test with artificial data

The experimental validation was carried out as follows. We have generated two repositories S^5 (with sets of points with 5 elements) and S^{30} (with sets of points of 30 elements), with the attributes of the nodes being real numbers. Each repository is integrated by 3 datasets consisting of 60 8-tuples $s_1 = \{G_1, G'_1, f_1^1, ..., f_1^6\}, .., s_i = \{G_i, G'_i, f_i^1, ..., f_i^6\}, ..., s_{60} = \{G_{60}, G'_{60}, f_{60}^1, ..., f_{60}^6\}$. All correspondences for each dataset are obtained using the following three correspondence generation scenarios:

- Completely at random: Six bijective correspondences are randomly generated per each tuple.
- Evenly distributed: From a "seed" bijective correspondence generated using the *Fast Bipartite* method presented by Serratosa (2014), two mappings are swapped randomly and a new correspondence is created. This process is repeated six times per each tuple. The seed correspondence is not included in the tuple.
- Unevenly distributed: From a "seed" bijective correspondence generated using the *Fast Bipartite* method presented by Serratosa (2014), pairs of mappings are swapped a random number of times and a new correspondence is created. This process is repeated six times per each tuple. Due to the randomness of the swaps, the seed correspondence may be included in the tuple.

The median was calculated for HD and CED-P (CED only using the attributes of the points as the local structure to have a fairer comparative) through the following methods:

1. SM. It is the correspondence in the set with the lowest sum of distances using HD and CED-P.
2. Evolutionary method presented by Moreno-García et al. (2016b) using HD and CED-P (EVOL1).
3. Evolutionary method presented by Moreno-García et al. (2016b) based on a modified weighted mean search strategy introduced in Moreno-García (2016) using HD and CED-P (EVOL2).
4. Minimisation method presented by Moreno-García et al. (2016b) using HD (M-HD) and the proposal of this paper using CED-P (M-CED-P).

Tables 1 to 3 show the following three metrics: a) SOD_{AVG}: the average and standard deviation of the SOD between the correspondences in the set and the mean correspondence. b) RED: the average percentage of reduction of the SOD with respect to the SM. c) RUN: the average runtime of each run (in seconds). Notice that since the HD and the CED are distances which exist in different spaces, a comparison of SOD_{AVG} results between HD and CED methods is not applicable. Moreover, RED scores are mostly meant to illustrate the improvement of each method

Table 1. Average SOD: SOD_{AVG} with its standard deviation. Reduction percentage of average SOD with respect to SM: RED. Runtime: RUN using the "Completely at random" scenario.

		Completely at random					
		S^5			S^{30}		
		SOD_{AVG}	RED	RUN	SOD_{AVG}	RED	RUN
HD	SM	19.16 ± 0.37	-	0.0009	140.6 ± 1.31	-	0.01
	M-HD	18 ± 0.26	5.26	0.002	137 ± 0.79	2.83	0.008
	EVOL1	19.16 ± 0.37	0	0.004	140.6 ± 1.31	0	0.1
	EVOL2	19.16 ± 0.37	0	0.009	138.91 ± 1.5	1.41	0.2
CED	SM	$62k \pm 1.33k$	-	0.01	$642k \pm 48k$	-	4.4
	M-CED-P	$60k \pm 0.87k$	3.22	0.02	$580k \pm 13k$	9.65	9.3
	EVOL1	$62k \pm 1.33k$	0	0.014	$642k \pm 48k$	0	4.7
	EVOL2	$62k \pm 1.33k$	0	0.007	$628k \pm 27k$	2.18	4.8

Table 2. Average SOD: SOD_{AVG} with its standard deviation. Reduction percentage of average SOD with respect to SM: RED. Runtime: RUN using the "Evenly distributed" scenario.

		Evenly distributed					
		S^5			S^{30}		
		SOD_{AVG}	RED	RUN	SOD_{AVG}	RED	RUN
HD	SM	13.42 ± 0.49	-	0.006	19 ± 6.3	-	0.01
	M-HD	12 ± 0	7.69	0.002	12 ± 0	36.84	0.003
	EVOL1	13.42 ± 0.49	0	0.003	15 ± 2.11	21.05	0.004
	EVOL2	13.42 ± 0.49	0	0.007	14 ± 1.76	26.32	0.02
CED	SM	18.4 ± 0.31	-	0.02	$63.1k \pm 9.58$	-	4.1
	M-CED-P	18.1 ± 0.26	1.63	0.03	$49.3k \pm 6.98$	21.87	9
	EVOL1	18.4 ± 0.31	0	0.003	$63.1k \pm 9.58$	0	3.5
	EVOL2	18.4 ± 0.31	0	0.007	$59k \pm 7.23$	6.5	3.5

Table 3. Average SOD: SOD_{AVG} with its standard deviation. Reduction percentage of average SOD with respect to SM: RED. Runtime: RUN using the "Unevenly distributed" scenario.

		Unevenly distributed					
		S^5			S^{30}		
		SOD_{AVG}	RED	RUN	SOD_{AVG}	RED	RUN
HD	SM	17.12 ± 0.27	-	0.006	66 ± 5.21	-	0.001
	M-HD	16 ± 0.13	5.88	0.002	53 ± 4.73	16.69	0.003
	EVOL1	17 ± 0.27	0	0.003	65 ± 5.02	1.51	0.006
	EVOL2	17 ± 0.27	0	0.007	64 ± 4.97	3.03	0.02
CED	SM	$76.5k \pm 3.8k$	-	0.005	$839k \pm 121k$	-	4.9
	M-CED-P	$69.1k \pm 2.1k$	9.67	0.002	$669k \pm 41k$	20.26	9.9
	EVOL1	$76.5k \pm 3.8k$	0	0.006	$839k \pm 121k$	0	5.3
	EVOL2	$76.5k \pm 3.8k$	0	0.01	$779k \pm 56k$	7.15	5.3

with respect to the SM in its own distance space, since the increment of HD is linear while CED depends on the attributes of the points.

For the ”Completely at random” datasets, Table 1 shows that both, for the average and the standard deviation, lower values have been achieved by the minimisation-based methods (M-HD and M-CED-P) compared to the alternatives in their respective spaces (S^5 and S^{30}). Moreover, it can be observed that in S^5, M-HD obtained better RED than M-CED-P (5.26% vs 3.22%), while the opposite occurs on S^{30} (9.65% vs 2.83%). Furthermore, it can also be seen that M-CED-P obtained the largest runtime, in particular on S^{30}, where the runtime with respect to SM is more than the double. Finally, it can be noticed that EVOL1 never outperforms the SM (in fact, it always obtains the SM), while EVOL2 only does on S^{30}. Both evolutionary methods have similar runtimes.

For the ”Evenly distributed” datasets shown in Table 2, the best SOD_{AVG} and RED results are obtained by M-HD on both repositories. In fact, in this experiment we confirm that M-HD always obtains the exact GM, given that the median correspondence for S^5 and S^{30} will always yield a SOD of 12 with respect to rest of correspondences in the set, with the standard deviation equal to 0 confirming that this occurs on every experiment. Note that this value of 12 results from multiplying the number of correspondences (i.e. 6) times the mappings swapped from the seed correspondence (i.e. 2), which is known in advance to be the GM. Given the attribute dependant nature of the CED, this rule is not visible for the SOD_{AVG}. For M-CED-P, the RED scores appear to be lower compared to M-HD. With regards to the runtime, it can be seen that once again M-HD achieves the lowest runtime on both repositories, while M-CED-P obtains the lowest for S^5, but the largest for S^{30}.

Finally, Table 3 shows the results for the ”Unevenly distributed” datasets. Notice that larger SOD_{AVG} values (both average and standard deviation) are obtained for all methods compared to the previous two scenarios. That is because the correspondences in each tuple are ”farther away” from each other and thus, the median correspondence obtained has a larger SOD with respect to the set. In this case, the RED is larger in M-CED-P compared to M-HD. Nonetheless, the computation of M-CED-P for the S^{30} dataset conveys the largest runtime. Meanwhile, EVOL1 and EVOL2 keep a similar trend to the previous two scenarios.

The following conclusions can be drawn from these experiments. If the correspondences have a low number of mappings or high precision is required, then M-CED-P is the best option. In contrast, M-HD has a better accuracy to runtime trade-off for huge graphs. It is also interesting to notice that the evolutionary methods, regardless of the weighted mean strategy, only outperformed the SM approach on the S^{30} repository, since the low amount of mappings in S^5 did not allow an effective weighted mean computation.

5.2. Test with real data

To show that our method, M-CED, deduces a more representative correspondence than the M-HD in a real scenario, we have used the Tarragona Exteriors Dataset presented by Moreno-García and Serratosa (2015), which is an image repository containing pictures of several objects obtained from different angles and perspectives. The dataset is comprised of 12 sequences: The first five sequences (i.e. ”BOAT”, EASTPARK”, ”EASTSOUTH”, ”ENSIMAG”, ”RESIDENCE”) contain 10 images, while the last seven (i.e. ”BARK”. ”BIKES”, ”GRAF”, ”LEUVEN”, ”TREES”, ”UBC”, ”WALL”) contain 6 images. We have used three graph repositories: $s = 5$, $s = 10$ and $s = 50$, consisting of graphs with 5, 10 and 50 nodes, respectively. Graphs have been created by applying the SURF point extractor presented by Bay et al. (2008) on all images, and the Delaunay triangulation to conform the edges. The image points used as nodes for the graphs are the s more reliable features in each case.

The experimental validation was set as follows. Using each graph dataset, for all distinct pairs of graphs in each sequence, we generated $3 \leq n \leq 50$ correspondences using three different methods: 1) the *Fast Bipartite* method presented by Serratosa (2014) with a random insertion/deletion cost, 2) Matlab's *MatchFeatures* function with a random *MaxRatio* value and 3) a random bijective correspondence. Notice that in the case that the first two algorithms do not obtain a bijective correspondence, null mappings were inserted to the nodes which were not mapped in the input graph. Afterwards, we have calculated the set median considering three distances: HD, CED considering only the points as the local structure and CED considering the star as the local structure. Thus, we have deduced the GM using these three distances, which yields on M-HD, M-CED-P (only using attributes on nodes) and M-CED-C (using the star as the local structure). The fact of comparing the only-points option is to analyse how important is the information of the edges and the neighbouring nodes.

Table 4 shows the average reduction of SOD (RED) obtained by the minimisation methods with respect to their corresponding SM for $n = 3$, $n = 10$ and $n = 30$. A value of 0 means that the SM is equal to the GM, and thus the minimisation-based approach has found the same correspondence as the SM approach. It can be seen that the RED value is much higher for M-CED-C and M-CED-P methods in comparison to the M-HD method. This can be attributed to the fact that CED is a distance that considers much more information than the HD. Moreover, notice that M-CED-C obtains a higher RED compared to M-CED-P, since the local substructure used considers the edit cost of the edges and the neighbouring nodes. Finally, it is worth to point out that the order of the graphs and the number of correspondences influence on the RED. When the number of nodes in the graphs increases, all methods are more capable of obtaining higher REDs. In terms of the number of correspondences used n, notice that all methods achieve higher performances at $n = 10$ in comparison to $n = 5$ and $n = 30$. This is due to the fact that it is more likely the SM to become the GM as the n increases, and thus the RED value gradually tends to be 0.

Finally, Table 5 shows the average runtime for all datasets across different configurations of correspondences used n and graph cardinalities s. Notice that as both values are increased, the runtime remains almost constant from the minimisation-based methods (i.e. M-HD, M-CED-P and M-CED-C) with respect to the SM-based methods (i.e. SM.HD, SM-CED-P and

Table 4. Average RED (reduction percentage between the SOD of the SM and the SOD of the GM) across the different sequences of the *Tarragona Exteriors* image dataset.

		n=3			n=10			n=30		
Dataset	s	M-HD	M-CED-P	M-CED	M-HD	M-CED-P	M-CED	M-HD	M-CED-P	M-CED
BOAT	s=50	0.00	0.81	3.90	0.00	2.55	**36.72**	0.00	3.80	29.91
	s=10	0.00	0.06	2.92	0.00	2.29	**33.99**	0.00	3.18	26.25
	s=5	0.01	0.01	0.84	0.00	1.03	**33.26**	0.00	2.37	29.99
EASTPARK	s=50	0.00	1.97	16.88	0.00	2.27	**31.23**	0.00	2.65	26.85
	s=10	0.00	0.11	7.88	0.00	1.84	**31.63**	0.00	2.40	24.61
	s=5	0.00	0.06	3.37	0.00	0.90	**29.76**	0.00	2.13	25.24
EASTSOUTH	s=50	0.00	0.18	2.48	0.00	2.55	**24.57**	0.03	2.99	20.16
	s=10	0.00	0.30	3.95	0.09	1.17	**26.08**	0.00	1.45	21.66
	s=5	0.00	0.05	1.75	0.05	0.74	**23.43**	0.01	1.70	21.46
ENSIMAG	s=50	0.00	3.50	21.50	0.00	2.96	**38.78**	0.00	4.03	34.63
	s=10	0.00	0.57	10.43	0.00	2.64	**38.45**	0.00	3.02	30.90
	s=5	0.01	0.33	4.57	0.04	1.54	**31.92**	0.04	3.24	24.05
RESIDENCE	s=50	0.00	4.22	11.26	0.00	7.62	**45.87**	0.00	6.90	35.00
	s=10	0.00	1.34	8.47	0.00	3.06	**39.76**	0.00	4.05	30.11
	s=5	0.01	0.09	2.11	0.02	1.38	**29.86**	0.03	2.69	26.00
BARK	s=50	0.00	0.84	3.31	0.00	7.23	**39.99**	0.00	6.13	32.82
	s=10	0.00	0.00	3.14	0.00	3.95	**31.87**	0.00	3.69	25.55
	s=5	0.00	0.02	1.06	0.00	0.64	**31.07**	0.00	2.11	26.08
BIKES	s=50	0.00	7.31	**27.26**	0.00	2.73	26.46	0.00	0.26	23.87
	s=10	0.00	3.33	24.38	0.14	1.13	**27.11**	0.05	1.31	18.04
	s=5	0.00	0.40	8.40	0.04	1.05	**26.44**	0.10	2.27	18.24
GRAF	s=50	0.00	1.50	5.07	0.00	2.90	**28.73**	0.00	4.90	27.56
	s=10	0.00	0.30	1.83	0.00	1.12	23.90	0.00	2.86	**24.17**
	s=5	0.00	0.01	0.66	0.03	0.90	**29.70**	0.00	2.18	28.08
LEUVEN	s=50	0.42	4.58	15.43	0.00	5.77	**37.81**	0.00	4.97	35.11
	s=10	0.00	1.02	17.43	0.00	2.12	**35.78**	0.00	3.34	25.68
	s=5	0.00	0.07	4.77	0.08	1.21	**29.04**	0.00	2.53	23.30
TREES	s=50	0.00	0.00	3.02	0.00	0.96	19.22	0.00	1.40	**19.99**
	s=10	0.00	0.19	2.13	0.00	0.80	**17.27**	0.00	1.38	15.90
	s=5	0.00	0.04	0.46	0.07	0.36	**17.73**	0.04	0.88	17.55
UBC	s=50	0.00	7.25	31.17	0.15	1.66	48.36	0.00	1.87	**49.23**
	s=10	0.00	4.20	21.03	0.00	2.77	**37.48**	0.00	2.45	28.02
	s=5	0.00	0.23	6.67	0.12	1.41	**22.56**	0.09	2.29	17.89
WALL	s=50	0.00	2.84	6.56	0.00	6.09	**42.45**	0.00	9.38	28.30
	s=10	0.00	0.27	3.97	0.00	2.55	**33.00**	0.00	3.97	22.63
	s=5	0.00	0.69	1.98	0.00	1.20	**28.79**	0.00	2.09	24.14
Average	s=50	0.03	2.92	12.32	0.01	3.77	**35.02**	0.00	4.10	30.29
	s=10	0.00	0.97	8.96	0.02	2.12	**31.36**	0.00	2.76	24.46
	s=5	0.00	1.01	1.98	0.04	1.31	**28.79**	0.03	2.23	24.14

SM-CED-C). It is clear that this shows a trend of scalability of the method towards larger graphs.

6. Conclusions

As graph matching becomes a more popular and widely available technology, the amount of graph correspondences that can be produced increases, thus delivering a vast collection of different functions between pairs of graphs. To find the representative prototype of a set of graph correspondences, thus reducing the computational demand of systems, we propose the use of the median given its robustness and applicability for different domains. In spite of the complexity of finding the GM for other data structures, we have presented a method based on optimisation functions which considers the edit distance between correspondences, called CED. In contrast to the classically used HD, this distance yields results which are more interesting from the application point of view.

Experimental validation on artificial and real datasets has shown that the minimisation-based method produces better results than using the SM or other state-of-the-art strategies in comparable runtime. Likewise, we have shown that for its application on a public dataset of images, the medians obtained are far better as more factors, such as the attributes and the local structures of the graphs, are considered.

In future work, we plan to implement this methodology for more application cases and considering more local structures for the CED.

Acknowledgments

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

References

Bay, H., Tuytelaars, T., Van Gool, L., 2008. Speeded-up robust features (SURF). Computer Vision and Image Understanding 110, 346–359.

Bille, P., 2005. A survey on tree edit distance and related problems. Theoretical Computer Science 337, 217–239.

Bunke, H., Günter, S., 2001. Weighted mean of a pair of graphs. Computing 67, 209–224.

Bunke, H., Jiang, X., Abegglen, K., Kandel, A., 2002. On the weighted mean of a pair of strings. Pattern Analysis and Applications 5, 23–30.

Caetano, T.S., McAuley, J.J., Cheng, L., Le, Q.V., Smola, A.J., 2009. Learning graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 1048–1058.

Carcassoni, M., Hancock, E.R., 2003. Correspondence matching with modal clusters. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 1609–1615.

Chaieb, R., Kalti, K., Luqman, M.M., Coustaty, M., Ogier, J.M., Amara, N.E.B., 2017. Fuzzy generalized median graphs computation: Application to content-based document retrieval. Pattern Recognition 72, 266–284.

Cortés, X., Moreno-García, C.F., Serratosa, F., 2013. Improving the correspondence establishment based on interactive homography estimation, in: International Conference on Computer Analysis of Images and Patterns, pp. 457–465.

Cortés, X., Serratosa, F., 2016. Learning graph matching substitution weights based on the ground truth node correspondence. International Journal of Pattern Recognition and Artificial Intelligence 30, 1650005.

Cortés, X., Serratosa, F., Moreno-García, C.F., 2016. Semi-automatic pose estimation of a fleet of robots with embedded stereoscopic cameras, in: Emerging Technologies and Factory Automation, pp. 1–6.

Ferrer, M., Valveny, E., Serratosa, F., Riesen, K., Bunke, H., 2010. Generalized median graph computation by means of graph embedding in vector spaces. Pattern Recognition 43, 1642–1655.

Franek, L., Jiang, X., 2012. Evolutionary weighted mean based framework for generalized median computation with application to strings, in: Structural, Syntactic, and Statistical Pattern Recognition, pp. 70–78.

Franek, L., Jiang, X., 2014. Ensemble clustering by means of clustering embedding in vector spaces. Pattern Recognition 47, 833–842.

Franek, L., Jiang, X., He, C., 2014. Weighted mean of a pair of clusterings. Pattern Analysis and Applications 17, 153–166.

Gao, X., Xiao, B., Tao, D., Li, X., 2010. A survey of graph edit distance. Pattern Analysis and Applications 13, 113–129.

Jiang, X., Abegglen, K., Bunke, H., Csirik, J., 2003. Dynamic computation of generalised median strings. Pattern Analysis and Applications 6, 185–193.

Jiang, X., Bunke, H., 2010. Learning by generalized median concept, in: Pattern Recognition and Machine Vision. chapter 15, pp. 231–246.

Jiang, X., Munger, A., Bunke, H., 2000. Synthesis of representative graphical symbols by computing generalized median graph, in: Graphics Recognition Methods and Applications, pp. 183–192.

Jiang, X., Münger, A., Bunke, H., 2001. On median graphs: properties, algorithms, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 1144–1151.

Jiang, X., Wentker, J., Ferrer, M., 2012. Generalized median string computation by means of string embedding in vector spaces. Pattern Recognition Letters 33, 842–852.

Jonker, R., Volgenant, A., 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. Computing 38, 325–340.

Kuhn, H.W., 1955. The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 2, 83–97.

Moreno-García, C.F., 2016. Learning the Consensus of Multiple Correspondences Between Data Structures. Ph.D. thesis. Universitat Rovira i Virgili.

Moreno-García, C.F., Cortés, X., Serratosa, F., 2016a. A graph repository for learning error-tolerant graph matching, in: Structural, Syntactic, and Statistical Pattern Recognition, pp. 519–529.

Moreno-García, C.F., Serratosa, F., 2015. Online learning the consensus of multiple correspondences between sets. Knowledge-Based Systems 90, 49–57.

Moreno-García, C.F., Serratosa, F., 2016. Consensus of multiple correspondences between sets of elements. Computer Vision and Image Understanding 142, 50–64.

Moreno-García, C.F., Serratosa, F., 2017a. Correspondence consensus of two sets of correspondences through optimisation functions. Pattern Analysis and Applications 20, 201–213.

Moreno-García, C.F., Serratosa, F., 2017b. Obtaining the consensus of multiple correspondences between graphs through online learning. Pattern Recognition Letters 87, 79–86.

Moreno-García, C.F., Serratosa, F., 2018. Modelling the generalised median correspondence through an edit distance, in: Structural, Syntactic, and Statistical Pattern Recognition, pp. 271–281.

Moreno-García, C.F., Serratosa, F., Cortés, X., 2016b. Generalised median of a set of correspondences based on the hamming distance, in: Structural, Syntactic, and Statistical Pattern Recognition, pp. 507–518.

Moreno-García, C.F., Serratosa, F., Jiang, X., 2017. An edit distance between graph correspondences, in: Graph-Based Representations in Pattern Recognition, pp. 232–241.

Moreno-García, C.F., Serratosa, F., Jiang, X., 2018. Correspondence edit distance to obtain a set of weighted means of graph correspondences. Pattern Recognition Letters Available online 30 August (in press).

Munkres, J., 1957. Algorithms for the assignment and transportation problems.

Navarro, G., 2001. A guided tour to approximate string matching. ACM Computing Surveys 33, 31–88.

Sanfeliu, A., Fu, K.S., 1983. A distance measure between attributed relational graphs for pattern recognition. IEEE Transactions on Systems, Man and Cybernetics SMC-13, 353–362.

Serratosa, F., 2014. Fast computation of bipartite graph matching. Pattern Recognition Letters 45, 244–250.

Serratosa, F., 2019. Graph edit distance: Restrictions to be a metric. Pattern Recognition 90, 250–256.

Solé-Ribalta, A., Serratosa, F., 2011. Models and algorithms for computing the

Table 5. Average runtime (in seconds) across the different sequences of the *Tarragona Exteriors* image dataset.

		SM-HD	SM-CED-P	SM-CED-C	M-HD	M-CED-P	M-CED-C
n=3	s=50	0.00003	0.00004	0.00003	0.00251	0.05249	0.49511
	s=10	0.00003	0.00003	0.00001	0.00026	0.00077	0.01699
	s=5	0.00003	0.00002	0.00001	0.00017	0.00030	0.00339
n=10	s=50	0.00012	0.00016	0.00015	0.00096	0.02183	0.44177
	s=10	0.00014	0.00011	0.00010	0.00020	0.00053	0.01633
	s=5	0.00015	0.00011	0.00010	0.00016	0.00026	0.00337
n=50	s=50	0.00240	0.00352	0.00350	0.00128	0.00618	0.42458
	s=10	0.00302	0.00232	0.00233	0.00029	0.00050	0.01643
	s=5	0.00345	0.00232	0.00228	0.00021	0.00031	0.00332

common labelling of a set of attributed graphs. Computer Vision and Image Understanding 115, 929–945.

Solé-Ribalta, A., Serratosa, F., 2013. Graduated assignment algorithm for multiple graph matching based on a common labeling. International Journal of Pattern Recognition and Artificial Intelligence 27, 1350001.

Solé-Ribalta, A., Serratosa, F., Sanfeliu, A., 2012. On the graph edit distance cost: properties and applications. International Journal of Pattern Recognition and Artificial Intelligence 26, 1260004.

Vento, M., 2015. A long trip in the charming world of graphs for pattern recognition. Pattern Recognition 48, 291–301.

Wagner, R.A., Fischer, M.J., 1974. The string-to-string correction problem. Journal of the ACM 21, 168–173.

Xiao, B., Hancock, E.R., Wilson, R.C., 2009a. A generative model for graph matching and embedding. Computer Vision and Image Understanding 113, 777–789.

Xiao, B., Hancock, E.R., Wilson, R.C., 2009b. Graph characteristics from the heat kernel trace. Pattern Recognition 42, 2589–2606.

Zhou, F., De La Torre, F., 2016. Factorized graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 38, 1774–1789.

Zitová, B., Flusser, J., 2003. Image registration methods: A survey. Image and Vision Computing 21, 977–1000.