

MORENO-GARCÍA, C.F., SERRATOSA, F. and CORTÉS, X. 2016. Generalised median of a set of correspondences based on the hamming distance. In: Robles-Kelly A., Loog M., Biggio B., Escolano F., Wilson R. (eds.) *Structural, syntatic and statistical pattern recognition: proceedings of the 2016 Joint International Association of Pattern Recognition (IAPR) structural, syntatic and statistical pattern recognition international workshop (S+SSPR 2016), 29 November - 2 December 2016, Mérida, Mexico*. Lecture Notes in Computer Science, vol 10029. Cham: Springer, pages 507-518. Available from: https://doi.org/10.1007/978-3-319-49055-7_45

Generalised median of a set of correspondences based on the hamming distance.

MORENO-GARCÍA, C.F., SERRATOSA, F. and CORTÉS, X.

2016



Generalised Median of a Set of Correspondences based on the Hamming Distance¹

Carlos Francisco Moreno-García, Francesc Serratosa & Xavier Cortés

Universitat Rovira I Virgili, Departament d'Enginyeria Informàtica i Matemàtiques, Spain
carlosfrancisco.moreno@estudiants.urv.cat
xavier.cortes@urv.cat
francesc.serratosa@urv.cat

Abstract: A correspondence is a set of mappings that establishes a relation between the elements of two data structures (i.e. sets of points, strings, trees or graphs). If we consider several correspondences between the same two structures, one option to define a representative of them is through the generalised median correspondence. In general, the computation of the generalised median is an NP-complete task. In this paper, we present two methods to calculate the generalised median correspondence of multiple correspondences. The first one obtains the optimal solution in cubic time, but it is restricted to the Hamming distance. The second one obtains a sub-optimal solution through an iterative approach, but does not have any restrictions with respect to the used distance. We compare both proposals in terms of the distance to the true generalised median and runtime.

Keywords: Correspondence, Mappings, Hamming Distance Generalised Median, Linear Assignment Problem.

1. Introduction

In several pattern recognition applications, there is a need to define an element-to-element relation between two objects. This process, commonly referred as “matching”, has been applied on data structures such as sets of points [1], strings [2], trees [3] and most notably, graphs [4], [5], [6], [7], [8]. While this previous work demonstrates that there has been a long standing effort to increase the quality of the methods that perform structural matching, it may also derive in scenarios where we encounter two or more parties which, having applied different matching algorithms, have produced several matching solutions. These solutions, onwards referred as “correspondences”, may be the result of the several existing methodologies or different parameterisations of these methodologies, which generate each time a different set of mappings between the elements of an output data structure and the elements of an input data structure.

Given a set of objects, their median is defined as the object that has the smallest sum of distances (SOD) [9], [10] to all objects in the set [11]. From this definition, we are able to identify the generalised median (GM) and the set median, which difference lies in the space where each median is searched for. In the first case there are no re-

¹This research is supported by projects DPI2013-42458-P, TIN2013-47245-C2-2-R, and by Consejo Nacional de Ciencia y Tecnología (CONACyT México).

restrictions for the search, while on the second case the exploration space is restricted to the elements in the set.

Due to its robustness, the concept of GM has been implemented to deduce the representative prototype of a set of data structures [12] and for clustering ensemble purposes [13] on data structures such as strings [14], graphs [15] and data clusters [16]. For these data structures (and for correspondences as well), the GM computation turns out to be an NP-complete problem. This drawback has led to a variety of methods solely developed for the GM approximation, such as a genetic search method [11] or approximations through the weighted mean of a pair of strings [17], graphs [18] or data clusters [19]. Most recently, a method known as the Evolutionary method [20] has been presented, offering a good trade-off between accuracy and runtime. This makes the Evolutionary method one of the most viable options for the GM approximation on most domains.

In this paper, we present two methodologies to obtain the GM of a set of correspondences. The first one is based on a voting and minimisation process, and the second one is based on the Evolutionary method adapted to the correspondence case. Notice that the calculation of a representative prototype of a set of correspondence has been studied before in what we called the ‘‘correspondence consensus frameworks’’ [21], [22], [23], [24], [25]. Nonetheless, there are some differences between approximating towards the GM and these consensus frameworks; the most important one being the function to be mini-mised. As commented before, the GM computation aims to minimise the SOD, whereas in the consensus framework, the function could also include the reduction of some other restrictions, such as the cost defined on the correspondences or the structural information of the data structures mapped.

The rest of the paper is structured as follows. In section 2, we introduce the basic definitions. In sections 3 and 4, we present both methods. In section 5, we compare them and evaluate the results in terms of the distance to the ground truth GM and the runtime. Finally, section 6 is reserved for conclusions and further work.

2. Basic Definitions

Let us represent any kind of data structure as $G = (\Sigma, \gamma)$, where $v_i \in \Sigma$ is an element inside the structure (elements can be, for instance, characters for strings, or nodes and edges for trees and graphs), and γ is a function that maps each element to a set of attributes. To allow maximum flexibility in the matching process, these structures may have been extended with null elements (represented as Φ), which have a set of attributes that differentiate them from the rest of the elements.

Given $G = (\Sigma, \gamma)$ and $G' = (\Sigma', \gamma')$ of the same order N (naturally or due to the aforementioned null element extension), we define the set of all possible correspondences $T_{G,G'}$ such that each correspondence in this set maps elements of G to elements of G' , $f: \Sigma \rightarrow \Sigma'$ in a bijective manner. Onwards, we refer to T instead of $T_{G,G'}$. Moreover, consider a subset of correspondences $S \in T$ between the same pair of structures that have been produced using different matching approaches.

Let f^1 and f^2 denote two different correspondences within S . We can deduct how dissimilar these two correspondences are through the Hamming distance (HD) function, which calculates the distance (number of different mappings) between f^1 and f^2 . More formally, the HD is defined as:

$$Dist_{HD}(f^1, f^2) = \sum_{i=1}^N (1 - \delta(v'_x, v'_y)) \quad (1)$$

being x and y such that $f^1(v_i) = v'_x$ and $f^2(v_i) = v'_y$, and δ being the well-known Kronecker Delta function.

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

Given a set of M input correspondences, the GM is the correspondence that has the smallest SOD to all objects in such set.

$$\hat{f} = \underset{\forall f \in T}{\operatorname{argmin}} \sum_{i=1}^M Dist_{HD}(f, f^i) \quad (3)$$

If the minimisation to find \hat{f} is restricted to be within the elements of S , then the solution is called the set median. Conversely, a search of all elements within T is known as the GM, which is considered a more attractive but more computationally demanding option. As noticed, the calculation of a median is closely related to the distance between the objects involved, and thus, the importance of defining the HD for the correspondence case.

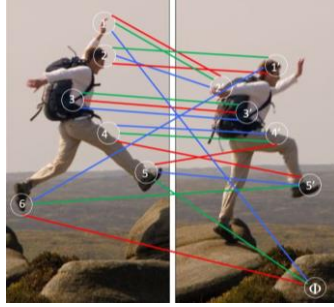


Figure 1. Example of three correspondences.

3. Minimisation Method

The first method presented in this paper is called Minimisation method. The name is related to the minimisation of the sum of the linear assignment problem (SLAP). To introduce it, consider the following example. Suppose that three separate entities have proposed correspondences as shown in figure 1, depicted as f^1 (red lines), f^2 (blue lines) and f^3 (green lines). Notice that as commented in section 2, the input set has been extended with a null element (Φ) to make correspondences mutually bijective.

We are able to represent these correspondences as correspondence matrices F^1 , F^2 and F^3 as shown in figure 2. These matrices are defined as follows. $F^k[x, y] = 1$ if $f^k(v_x) = v'_y$ and $F^k[x, y] = 0$ otherwise.

	F^1_{ϕ}						F^2_{ϕ}						F^3_{ϕ}					
	1'	2'	3'	4'	5'	6'	1'	2'	3'	4'	5'	6'	1'	2'	3'	4'	5'	6'
1	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
2	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
3	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0
5	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1
6	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0

Figure 2. Correspondences matrices F^1 , F^2 and F^3 .

Our method minimises the following expression:

$$\hat{f} = \underset{\forall f \in T}{\operatorname{argmin}} \left\{ \sum_{x,y=1}^N [H \circ F]\{x, y\} \right\} \quad (4)$$

where $\{x, y\}$ is a specific cell and H is the following matrix:

$$H = \sum_{k=1}^M \mathbf{1} - F^k \quad (5)$$

with $\mathbf{1}$ being a matrix of all ones, F being the correspondence matrix of $f \in T$ (if $f(v_x) = v'_y$ then $F\{x, y\} = 1$, otherwise $F\{x, y\} = 0$) and \circ being the Hadamard product.

We deduct \hat{f} through equation 4 but we wish to minimise the SOD of all correspondences to obtain the true \hat{f} (equation 3). Therefore, we have to demonstrate that the obtained correspondence \hat{f} in equation 3 is the same than the one in equation 4. For this reason, we have to demonstrate that equation 6 holds:

$$\sum_{k=1}^M \operatorname{Dist}_{HD}(f, f^k) = \sum_{x,y=1}^N \left[\left[\sum_{k=1}^M \mathbf{1} - F^k \right] \circ F \right] \{x, y\} \quad (6)$$

Applying the associative property of Hadamard product, the following expression is obtained:

$$\sum_{k=1}^M \operatorname{Dist}_{HD}(f, f^k) = \sum_{k=1}^M \left(\sum_{x,y=1}^N [[\mathbf{1} - F^k] \circ F]\{x, y\} \right) \quad (7)$$

Then, if we demonstrate that each individual term holds the equality $\operatorname{Dist}_{HD}(f, f^k) = \sum_{x,y=1}^N [[\mathbf{1} - F^k] \circ F]\{x, y\}$, then for sure equation 7 holds. As shown in its definition, the HD counts the number of mappings that are different between the two correspondences and similarly, expression $\sum_{x,y=1}^N [[\mathbf{1} - F^k] \circ F]\{x, y\}$

does, since this last expression counts the number of times $F\{x, y\} = 1$ and that simultaneously $F^k\{x, y\} = 0$.

Notice that by adding all correspondence matrices in equation 5, we create a structure similar to a voting matrix [25]. This method is based on minimising the linear assignment problem applied to this voting matrix using any solver such as the Hungarian method [27], the Munkres algorithm [28] or the Jonker-Volgenant solver [29] as shown in Algorithm 1:

Algorithm 1: Minimisation

Input: A set of correspondences
Output: GM correspondence \hat{f}
 Compute matrix H (equation 5)
 $\hat{f} = \text{linear solver}(H)$ ([27], [28] or [29])

End Algorithm

Since the three solvers have all been demonstrated to obtain an optimal value, it is possible to guarantee that this method obtains the exact GM, given that there is only first order information involved and no second order information is considered. That is, we do not take into account relations between the mapped elements inside their set. Figure 3 shows the GM correspondence obtained for this particular practical example.



Figure 3. GM correspondence of the three correspondences in figure 1.

4. Evolutionary Method

The second option explored in this paper for the GM correspondence computation is the use of the meta algorithm presented in [20] called Evolutionary method. This proposal relies on the concept of the weighted mean of a pair of correspondences, which is defined as follows. Given f^1, f^2 and a distance $Dist$ between them (for instance $Dist_{HD}$), the mean correspondence of f^1 and f^2 is a correspondence $\bar{f} \in T$ such that:

$$\begin{aligned} Dist(f^1, \bar{f}) &= Dist(\bar{f}, f^2) \\ Dist(f^1, f^2) &= Dist(f^1, \bar{f}) + Dist(\bar{f}, f^2) \end{aligned} \tag{8}$$

Additionally, the weighted mean correspondence $\bar{f}_\alpha \in T$ is defined as a correspondence in T that holds:

$$\begin{aligned}
Dist(f^1, \bar{f}_\alpha) &= \alpha \\
Dist(f^1, f^2) &= \alpha + Dist(\bar{f}_\alpha, f^2)
\end{aligned} \tag{9}$$

where α is a constant: $0 \leq \alpha \leq Dist(f^1, f^2)$

Clearly, $\bar{f}_0 = f^1$ and $\bar{f}_{Dist(f^1, f^2)} = f^2$. Nevertheless, both the mean correspondence and the weighted mean correspondence given a specific α are usually not unique. The concept of the weighted mean has been previously defined for strings [17], graphs [18] and data clusters [19].

As proven in [11], the GM of some elements in any space can be estimated through an optimal partition of the pairs of these elements. This is because they demonstrated that by computing the weighted mean of such optimal pairs of elements, all of those weighted means tend to match in one element that can be considered a good estimation of the GM of the set. Since in some cases the GM can be far away from the deducted element, an iterative algorithm is proposed in [20] which tends to achieve the true GM. This algorithm, applied to the correspondence domain, consists on the steps shown in Algorithm 2:

Algorithm 2: Evolutionary

Input: A set of correspondences

Output: GM correspondence \hat{f}

While convergence

1. Deduct the optimal pairs of correspondences.
2. Estimate some weighted means per each pair.
3. Add the weighted means to the current set of correspondences.
4. Select the optimal correspondences in the current set.

End Algorithm

We proceed to detail steps 1, 2 and 4 in the correspondence domain. Notice the third step is simply adding the obtained weighted mean correspondences to the current set of correspondences.

4.1 Optimal Pairs of Correspondences

We generate the distance matrix given the whole correspondences, where any distance between these correspondences can be used. Then, the optimal pairs of elements are considered the ones that generate the minimum SOD between them [11]. Thus, we simply obtain the pairs of correspondences by applying a SLAP solver such as the Hungarian method [27], the Munkres algorithm [28] or the Jonker-Volgenant solver [29]. Note that we do not want one correspondence to be assigned as the optimal pair of itself and for this reason, instead of filling the diagonal of the distance matrix with zeros, we impose a high value. Nevertheless, if there is an odd number of correspondences, for sure the solver returns a correspondence mapped to itself. In this case, this correspondence is stored until the third step.

4.2 Weighted Means of Pairs of Correspondences

The aim of the second step is to estimate Ω equidistant weighted means per each pair of correspondences. Thus, we generate $\bar{f}_{\alpha_1}, \dots, \bar{f}_{\alpha_\Omega}$ such that $\alpha_i = \frac{i}{\Omega+1}$ (equation 9). The order of Ω is usually set from 1 to 3. This is because, through the practical

validation, we have seen that restricting the process to calculate only the mean correspondence (that is $\Omega = 1$) makes the process converge slower than when having for instance three equidistant weighted means, even though these are obtained in a sub-optimal form. Moreover, experimentation has shown that if $\Omega > 3$, the computational cost is also increased without gaining in accuracy.

The weighted mean search strategy we used is inspired by the ‘‘Moving Elements Uniformly’’ strategy presented in [19] for the domain of data clusters. In that case, they were able to generate multiple weighted mean data clusters from two initial ones. To do so, authors defined an initial weighted mean as one of the data clusters, and then they systematically swap elements that belong to two different clusters in the weighted data cluster in such a way that the weighted mean data clusters formed tends to move from one of the initial data clusters into the other one.

Our proposal initially defines the weighted mean correspondence as one of the correspondences. Then, it simply swaps pairs of element-to-element mappings in the proposed weighted mean \bar{f}_α . Note, every time a swap is performed, the value $Dist(f^1, \bar{f}_\alpha)$ is increased by two, but we cannot guarantee that $Dist(f^2, \bar{f}_\alpha)$ is also decreased by two. For this reason, the strategy checks if the current correspondence is a true weighted mean (equation 9 holds). If it is the case, a weighted mean has been formed and the swapping process continues until finding all required weighted means. If it is not the case, the process is reset and repeated until finding weighted means. This method has its base on a theorem presented in [21], where it was shown that a weighted mean correspondence has to hold that $\bar{f}_\alpha(v_x) = f^1(v_x)$ or $\bar{f}_\alpha(v_x) = f^2(v_x)$ for all elements in the weighted mean correspondence.

4.3 Selecting the Optimal Correspondences

Once the current correspondences are put together with the new weighted mean correspondences to enlarge the set (step 3), the method could return to the first step of Algorithm 2 with this newly enlarged set without running the fourth step. Nevertheless, the computational cost and memory space needed in each iteration would exponentially increase. For this reason, the aim of this fourth step is to discard the correspondences that are believed not to be a good choice for the GM. To that aim, a distance matrix is computed between the whole correspondences. Then, the ones that have a larger SOD from themselves to the rest are discarded. Note that this methodology is in line of the GM (equation 3).

When the fourth step finishes, Algorithm 2 iterates again until one of three options happens: 1) The sum of the minimum SOD of the whole correspondence in the set is lower than a threshold. 2) A maximum number of iterations is achieved. 3) A minimum difference on the total SOD between the previous iteration and the current one is achieved. Independently of the terminating option, Algorithm 2 returns the correspondence in the set that has at the moment the minimum SOD to the set as the GM correspondence. Convergence is assured since the SOD, in each iteration, is equal or lower than the previous iteration. Moreover, in case the SOD is kept equal, Algorithm 2 stops.

5. Experimental Validation

Two methods have been presented. The first one obtains the exact GM correspondence, but is restricted to the use of the HD. The second one deducts an approximation of the GM correspondence, but is not restricted to any distance between corre-

spondences. In this section, we show how close the suboptimal method is with respect to the optimal one, as well as the runtime of both methods. To have a fair comparison, we have used the HD in both cases.

We performed three tests, all of them executed the same way but using correspondences with $N = 5$, $N = 10$ and $N = 30$ mapped elements in each case. Each test was prepared as follows. We randomly generated 100 sets of $M = 2, 3 \dots, 50$ correspondences. For each set, both methods to find the GM correspondence are executed. In the Minimisation method, the Hungarian method [27] was used to solve the SLAP.

Figure 4 shows the normalised difference on the SOD that the GM correspondences generated by the Evolutionary method obtained with respect to the ones from the Minimisation method (x-axis) in the first test ($N = 5$), second test ($N = 10$) and third test ($N = 30$) respectively. Each dot in the plot represents the average of the 100 executions. For the Evolutionary method, we show results using the number of iterations $I_{max} = 1$ and $I_{max} = 2$. Results for larger values of I_{max} are not shown since they deliver exactly the same values that the ones of $I_{max} = 2$.

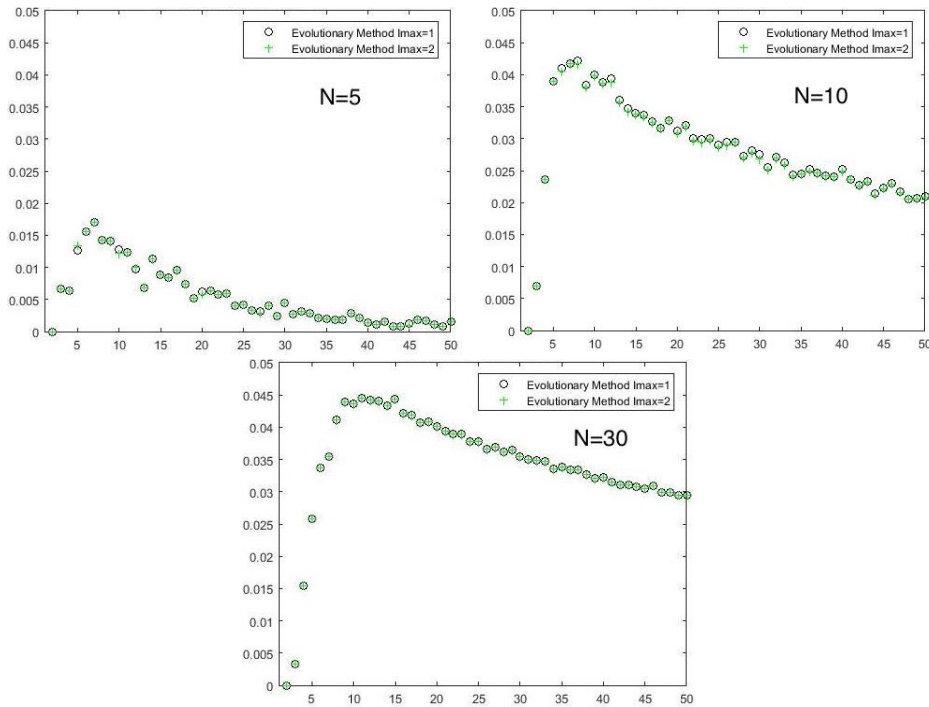


Figure 4. Average difference of SOD (HD) between the Evolutionary method and Minimisation method (x -axis, optimal method) for $N = 5$, $N = 10$ and $N = 30$.

In the three cases for a set of two correspondences, $M = 2$, the Evolutionary method obtains optimal GM correspondences since the method only has to deal with the mean calculation. Nonetheless as the number of correspondences M increases, this overestimation has a peak maximum value, and then it decreases until lowering down again towards the optimal value of the GM correspondence. This leads us to think that the Evolutionary method has an optimal number of correspondences to be used, since

certain values of M lead more overestimation than others. Finally, from these plots we conclude that the Evolutionary method, regardless of the I_{max} value used, obtains values that are really close to the optimal ones. In fact, the worst case overestimates the SOD in 4.5% with respect to the optimal SOD.

Figure 5 shows the runtime difference between the Evolutionary method and the Minimisation method (x -axis) in seconds. In the case of the Evolutionary method, it is clear that the time spent in each iteration is constant. Comparing both methods, the minimisation one is clearly faster than the evolutionary one, although both have a polynomial computational cost with respect to the number of correspondences used to deduct the GM. Finally, comparing the three plots, we realise the number of elements N in the sets seems to have almost no influence on the runtime.

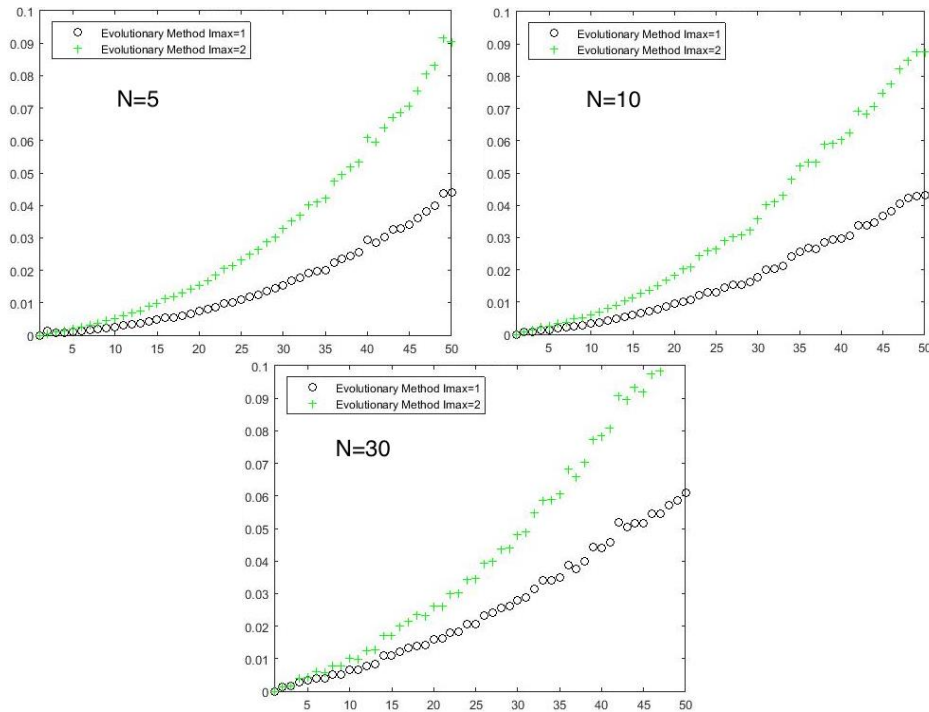


Figure 5. Average difference of runtime (seconds) between the Evolutionary method and Minimisation method (x -axis) for $N = 5$, $N = 10$ and $N = 30$.

6. Conclusions and Future Work

We have presented two methods to deduct the GM correspondence. The first one, called Minimisation method, computes the exact GM in a reduced runtime, but it is bounded to the use of the HD. Since it is based on the solution of the SLAP, it is limited in scalability for the cases where correspondences have a large size. The second one, called Evolutionary method, obtains a fair approximation of the GM, and may be used with any type of distance between correspondences. This method has better scalability, given that although there is a SLAP solution required (step 1), it only takes into consideration the distance between correspondences, and not the correspondences themselves as in the Minimisation method. Notice that the Evolutionary

method has been a viable solution for the approximation of the GM of strings and graphs [20], since these structures imply second order relations and therefore finding their exact GM has an exponential cost.

In the concrete case that the aim is to find the GM of correspondences and the HD can be used, we have shown that this problem can be solved in cubic time (the computational cost of the SLAP) using the Minimisation method. Nevertheless, we consider this paper as a first step towards future research in which other distances between correspondences will be explored, and thus the Evolutionary method should not be discarded for future uses.

We believe that other distances between correspondences which take into consideration not only the element-to-element mapping, but also the structure and attributes of the related elements, could produce more interesting GM correspondences from the application point of view. For instance, in the situation that the correspondences relate attributed graphs, the mapping is defined as node-to-node. In this case, we could consider the local structure of the nodes (its adjacent edges and their terminal nodes) to penalise the cost of the mapping. Then, the Minimisation method would not produce an exact GM and therefore, we would need to compare both algorithms, not only from the runtime point of view, but also in terms of accuracy to deduct the best approximation to the GM correspondence.

References

1. Zitová, B., Flusser, J., "Image registration methods: a survey". *Image Vision Computing* 21 (11), pp. 977-1000, 2003.
2. Navarro, G., "A guided tour to approximate string matching". *ACM Computing Surveys* 33 (1), pp. 31-88, 2001.
3. Bille, P., "A survey on tree edit distance and related problems". *Theoretical Computer Science* 337 (9), pp. 217-239, 2005.
4. Conte, D., Foggia, P., Sansone, C., Vento, M., "Thirty years of graph matching in pattern recognition". *International Journal of Pattern Recognition and Artificial Intelligence* 18 (3), pp. 265-298, 2004.
5. Foggia, P., Percannella, G., Vento, M., "Graph matching and learning in pattern recognition in the last ten years". *International Journal of Pattern Recognition and Artificial Intelligence* 28 (1), pp. 1450001, 2014.
6. Serratos, F., "Fast computation of bipartite graph matching". *Pattern Recognition Letters* 45, pp. 244 - 250, 2014.
7. Serratos, F., "Computation of graph edit distance: reasoning about optimality and speed-up". *Image and Vision Computing* 40, pp. 38-48, 2015.
8. Serratos, F., "Speeding up fast bipartite graph matching through a new cost matrix". *International Journal Pattern Recognition and Artificial Intelligence*, 29 (2), pp. 1550010, 2015.
9. Solé, A., Serratos, F., Sanfeliu, A., "On the graph edit distance cost: properties and applications". *International Journal of Pattern Recognition and Artificial Intelligence* 26 (5), pp. 1260004, 2012.
10. Serratos, F., Alquézar, R., Sanfeliu, A., "Efficient algorithms for matching attributed graphs and function-described graphs". *Proceedings of the 15th International Conference on Pattern Recognition, (ICPR) 2*, pp: 871-876, 2000.
11. Jiang, X., Munger, A., Bunke, H., "On median graphs: properties, algorithms, and applications". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (10), pp. 1144-1151, 2001.

12. Jiang, X., Bunke H., “Learning by generalized median concept”. Wang, P. (Ed). River Publisher. Pattern Recognition and Machine Vision, chapter 15, pp. 231–246, 2010.
13. Vega-Pons, S., Ruiz-Shulcloper, J., “A survey of clustering ensemble algorithms”. International Journal of Pattern Recognition and Artificial Intelligence 25 (3), pp. 337-372, 2011.
14. Jiang, X., Wentker, J., Ferrer, M., “Generalized median string computation by means of string embedding in vector spaces”. Pattern Recognition Letters 33, pp. 842-852, 2012.
15. Ferrer, M., Valveny, E., Serratos, F., Riesen, K., Bunke, H., “Generalized median graph computation by means of graph embedding in vector spaces”. Pattern Recognition 43, pp. 1642-1655, 2010.
16. Franek, L., Jiang, X., “Ensemble clustering by means of clustering embedding in vector spaces”. Pattern Recognition, 47 (2), pp. 833–842, 2014.
17. Bunke, H., Jiang, X., Abegglen, K., Kandel, A., “On the weighted mean of a pair of strings”. Pattern Analysis and Applications 5, pp. 23-30, 2002.
18. Bunke, H., Gunter, S., “Weighted mean of a pair of graphs”, Computing 67, pp. 209-224, 2001.
19. Franek, L., Jiang, X., He, C., “Weighted mean of a pair of clusterings”. Pattern Analysis and Applications 17, pp. 153-166, 2014.
20. Franek, L., Jiang, X., “Evolutionary weighted mean based framework for generalized median computation with application to strings”. Gimel, G. et al. (Eds.). Syntactic and Structural Pattern Recognition (SSPR), LNCS 7626, pp. 70-78, 2012.
21. Moreno-García, C. F., Serratos, F., “Consensus of two sets of correspondences through optimisation functions”. Pattern Analysis and Applications, pp. 1-13, 2015.
22. Moreno-García, C. F., Serratos, F., “Consensus of multiple correspondences between sets of elements”. Computer Vision and Image Understanding 142, pp. 50-64, 2015.
23. Moreno-García, C. F., Serratos, F., Cortés, X., “Consensus of two graph correspondences through a generalization of the bipartite graph matching”. Liu, C.L. et al. (Eds.). Graph Based Representations and Pattern Recognition (GbR), LNCS 9069, pp. 87–97, 2015.
24. Moreno-García, C. F., Serratos, F., “Online learning the consensus of multiple correspondences between sets”. Knowledge-Based Systems 90, pp. 49-57, 2015.
25. Moreno-García, C. F., Serratos, F., “Obtaining the consensus of multiple correspondences between graphs through online learning”. Pattern Recognition Letters. Available Online, <http://dx.doi.org/10.1016/j.patrec.2016.09.003>, 2016.
26. Saha, S., Ekbil, A., “Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition”. Data & Knowledge Engineering 85, pp. 15-39, 2013.
27. Kuhn, H.W., “The Hungarian method for the assignment problem export”. Naval Research Logistics Quarterly 2 (1-2), pp. 83–97, 1955.
28. Munkres, J., “Algorithms for the assignment and transportation problems”. Journal of the Society of Industrial and Applied Mathematics 5(1), pp. 32–38, 1957.
29. Jonker, R., Volgenant, T., “Improving the hungarian assignment algorithm”. Operations Research Letters 5 (4), pp. 171-175, 1986.