Consensus of Two Graph Correspondences Through a Generalisation of the Bipartite Graph Matching

Carlos Francisco Moreno-García(✉), Francesc Serratosa, and Xavier Cortés

Universitat Rovira i Virgili, Tarragona, Spain
carlosfrancisco.moreno@estudiants.urv.cat,
{xavier.cortes,francesc.serratosa}@urv.cat

Abstract. One of the most important processes related to structural pattern recognition is to compare the involved objects through representing them as attributed graphs and using error-tolerant graph matching methods. To do so, it is needed a first step to extract the graphs given the original objects and deduct the whole attribute values of nodes and edges. Depending on the application, there are several methods to obtain these graphs and so, the object at hand can be represented by several graphs, not only with different nodes and edges but also with different attribute domains. In the case that we have several graphs to represent the same object, we can deduct several correspondences between graphs. In this work, we want to solve the problem of having these correspondences by exploding this diversity to announce a final correspondence in which the incongruences introduced in the graph extraction and also the graph matching could be reduced. We present a consensus method which, given two correspondences between two pairs of attributed graphs generated by separate entities and with different attribute domains, enounces a final correspondence consensus considering the existence of outliers. Our method is based on a generalisation of the Bipartite graph matching algorithm that minimises the Edit cost of the consensus correspondence while forcing (to the most) to be the mean correspondence of the two original correspondences.

Keywords: Bipartite graph matching · Graph correspondence · Consensus correspondence

1 Introduction

When two parties decide to solve the assignment problem between two images, differences in the mappings may occur. In [1], it is explained how consensus methodologies are used to combine two different mappings between images to obtain a final consensus mapping. That work was inspired on [2], where a weighted mean consensus of a pair of clustering was obtained. In this paper, we generalise the paper presented in [1] since the input of our method is composed of two correspondences between attributed graphs instead of two correspondences between two sets of points

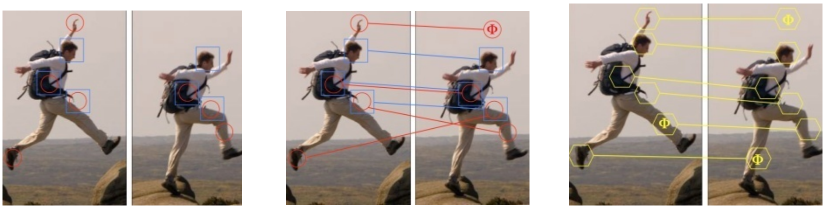
This research is supported by the Spanish CICYT project DPI2013-42458-P, by project TIN2013-47245-C2-2-R and by Consejo Nacional de Ciencia y Tecnologías (CONACyT Mexico)

C.-L. Liu et al. (Eds.): GbRPR 2015, LNCS 9069, pp. 87–97, 2015.
DOI: 10.1007/978-3-319-18224-7_9

that represent images. In this way, attributed graphs can represent any type of object, being images one of them. This is a useful property, since in the image registration domain, some techniques have appeared that represent images as an attributed graph and then a correspondence between graphs is deducted, such as in [3] and [4].

Suppose we have two images and we have used two different salient point extractors (for instance, SIFT and SURF) on these two images to represent them as a set of points. Figure 1.a shows the two images and the extracted salient points. Red circles and blue squares represent the two different extractors. We first realise there are some points that have been selected from both methods but other ones have been selected by only one of the methods. Then, from each image and from each set of points we generate a graph using any structural method such as Delaunay triangulation or K-Nearest neighbours. Therefore, we have graphs G^a and G^b that represent the left image and graphs G'^a and G'^b that represent the right image (graph edges are not drawn). Nodes in G^a and G'^a are drawn in red circles and nodes in G^b and G'^b are drawn in blue squares. If we want to compare these images, we need to apply an error-tolerant graph matching method to deduct the node correspondences and also a distance. There are some options, for instance, Graduated Assignment [5] or Bipartite Graph Matching [6], [7], [8], [9], [10]. Figure 1.b shows the obtained correspondences between both pairs of graphs. Note there are some discrepancies not only on the selected points, but also in the node mappings. Moreover, some new nodes (we call them null nodes) have been introduced to assure correspondences f^a (red lines) and f^b (blue lines) be bijective. Figure 1.c shows the final obtained consensus $f^{a,b}$. Notice $G^{a,b}$ has one null node and $G'^{a,b}$ has two null nodes.

In summary, the input of our method consists of two bijective correspondences $f^a: G^a \rightarrow G'^a$ and $f^b: G^b \rightarrow G'^b$, as well as two node mappings that mark which points in [G^a and G^b] and [G'^a and G'^b] have to be considered the same ones, forcing the node intersection between them. Contrarily of the method presented in [1] that has to be used on images, our method is independent of the domain of these graphs.



(a): Two feature extractors on two images (b): $f^a$and f^b correspondences (c): $f^{a,b}$correspondence

Fig. 1. The process of obtaining a consensus correspondence from the original extracted salient points

One of the most well-known and practical options to reduce the complexity of a combinatorial calculation of all the possible consensus options is combinatorial optimisation. The concept of optimisation is related to the selection of the "best" configuration or set of parameters to achieve a certain goal [9]. Functions involved in an optimisation problem can be either conformed by continuous values or discrete values, often called as combinatorial scenarios. These scenarios have been largely studied and applied for matching problems, particularly the case of the Hungarian algorithm [10] and the Jonker-Volgenant solver [11]. This method converts a combinatorial problem into a correspondence problem, which will eventually derive in an optimal configuration for a cost-based correspondence. Recently, some collaborative methods have been proposed that given a set of classifiers; return the most promising class [12]. These methods learn some weights that gauge the importance of each classifier and also of the sample through several techniques, such as voting [13] or hierarchical methods [14]. Nevertheless, these methods cannot be directly adapted to our problem, since their output is a class index and our output is a whole correspondence between two sets of salient points.

To generate a correspondence between graphs, several proposals can be found on literature. One of the most relevant in recent years is Bipartite Graph Matching (BP) [8], which has demonstrated to be the efficient and error-tolerant. Also, in [6] and [7] a version to calculate BP in a fast and efficient form has been presented, called Fast bipartite and Square Fast Bipartite (FBP and SFBP). In the Bipartite algorithms, it is important to consider which local sub-structure has been used. This situation, acknowledged in [15] and [16], is also a matter of discussion on this paper.

The paper is structured as follows. In section 2 we briefly define and explain attributed graphs and Bipartite graph matching. In section 3 and 4 we explain how we generalise the BP algorithm to obtain the consensus correspondence and we demonstrate our method. In section 5 we show the experimental evaluation. There is a first explanation of the used database, which we defined and made public [17]. Its main feature is that it is composed of pairs of graphs and some ground truth correspondences between them. Moreover, there is also some information of which nodes of different graphs have to be considered the same ones in the consensus process. Finally, section 6 concludes the paper and presents our future work.

2 Attributed Graphs and Bipartite Graph Matching

Let $G^a = (\Sigma_v^a, \Sigma_e^a, \gamma_v^a, \gamma_e^a)$ and $G'^a = (\Sigma'^a_v, \Sigma'^a_e, \gamma'^a_v, \gamma'^a_e)$ be two attributed graphs. To allow maximum flexibility in the matching process, these graphs have been extended with null nodes to be of order n^a. $\Sigma_v^a = \{v_i^a \mid i = 1, \ldots, n^a\}$ is the set of vertices and $\Sigma_e^a = \{e_{i,j}^a | i, j \in 1, \ldots, n^a\}$ is the set of edges. Functions $\gamma_v^a : \Sigma_v^a \rightarrow \Delta_v^a$ and $\gamma_e^a : \Sigma_e^a \rightarrow \Delta_e^a$ assign attribute values in any domain to vertices and edges. Coherent definitions hold for $G'^a = (\Sigma'^a_v, \Sigma'^a_e, \gamma'^a_v, \gamma'^a_e)$. One of the most widely used methods to evaluate an error-correcting graph isomorphism is the Graph edit distance [18], [19], [20]. The dissimilarity is defined as the minimum amount of required distortion to transform one graph into the other. To this end, a number of distortion or edit operations, consisting of insertion, deletion and substitution of nodes and edges 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 transformed to assignations of a non-null node of the first or second graph to a null node of the second or first graph. Substitutions simply indicate node-to-node assignations. Using this transformation, given two graphs G^a and G'^a, and a bijection between their nodes f^a, the graph edit cost $EditCost(G^a, G'^a, f^a)$ is computed [20]. It is based on the following constants and functions: C_{vs} is a function that represents the cost of substituting node v_i^a of G^a by node $f^a(v_i^a)$ of G'^a. C_{es} is a function that represents the cost of substituting edge $e_{i,k}^a$ of G^a by edge$f^a\left(e_{i,k}^a\right)$ of G'^a. Constant K_v is the cost of deleting node v_i^a of G^a(mapping it to a null node) or inserting node v'^a_j of G'^a (or being mapped from a null node). Likewise for the edges, K_e is the cost of assigning edge $e_{i,k}^a$ of G^a to a null edge of G'^aor assigning edge $e'^a_{j,p}$ of G'^a to a null edge of G^a. Note that we have not considered the cases in which two null nodes or null edges are mapped; this is because this cost is zero by definition.

The Graph edit distance $EditDist$ is defined as the minimum cost under any bijection in T:

$$EditDist(G^a, G'^a) = \min_{f^a \in T}\{EditCost(G^a, G'^a, f^a)\} \tag{1}$$

We say the optimal bijection, f^{a*}, is the one that obtains the minimum cost.

BP algorithm [8] is composed of three main steps. The first step defines a cost matrix, the second step applies a linear solver such as the Hungarian method or the Jonker-Volgenant method to this matrix and obtains the correspondence f^{a*}. The third step computes the Edit distance cost given this correspondence between both graphs, $EditDist(G^a, G'^a) = EditCost(G^a, G'^a, f^{a*})$. Figure 2 shows the cost matrix of the BP algorithm.

$$C^{BP} = n+m\left\{\overbrace{\left[\begin{array}{cccccc|cccccc}
C_{1,1} & \cdots & \cdots & \cdots & \cdots & C_{1,m} & C_{1,\varepsilon} & \infty & \cdots & & & \infty \\
\vdots & \ddots & & & & & \infty & \ddots & & & \infty & \\
\vdots & & C_{a,\varepsilon} & & & & & & C_{a,\varepsilon} & & & \\
\vdots & & & & & \vdots & & & & & & \vdots \\
\vdots & & & & \ddots & \vdots & & \infty & & & \ddots & \infty \\
C_{n,1} & \cdots & \cdots & \cdots & \cdots & C_{n,m} & \infty & & & \cdots & \infty & C_{n,\varepsilon} \\
\hline
C_{\varepsilon,1} & \infty & \cdots & & & \infty & & & & & & \\
\infty & \ddots & & & \infty & & & & & & & \\
\vdots & & C_{\varepsilon,a} & & & & & & 0 & & & \\
 & \infty & & & \ddots & \vdots & & & & & & \\
\infty & & & & & \infty & & & & & & \\
 & & & \cdots & \infty & C_{\varepsilon,m} & & & & & &
\end{array}\right]}^{n+m}\right.$$

Fig. 2. Cost matrix of the BP algorithm

Quadrant Q1 denotes the combination of substituting costs $C_{i,j}$ and their local sub-structures. The diagonal of quadrant Q2 denotes the whole costs $C_{i,\varepsilon}$ of deleting nodes v_i^a and its local sub-structures. Similarly, the diagonal of quadrant Q3 denotes the whole costs $C_{\varepsilon,j}$ of inserting nodes v'^a_j and its local sub-structures. Q4 quadrant is filled with zero values since the substitution between null nodes has a zero cost. In this paper, we propose a method to perform a consensus of two initial correspondences using the most used local sub-structures, viz. node, degree and clique. In the node case, edges are not considered. The degree is composed of the set of neighbouring edges and the clique is composed of the set of neighbouring edges and also the neighbouring nodes. Other structures have been presented in [15], [16].

3 Consensus of a Pair of Correspondences Between Graphs

Assume $f^a: \Sigma_v^a \rightarrow \Sigma'^a_v$ and $f^b: \Sigma_v^b \rightarrow \Sigma'^b_v$ are two correspondence functions between nodes of two attributed graphs $G^a = (\Sigma_v^a, \Sigma_e^a, \gamma_v^a, \gamma_e^a)$ and $G^b = (\Sigma_v^b, \Sigma_e^b, \gamma_v^b, \gamma_e^b)$ and two attributed graphs $G'^a = (\Sigma'^a_v, \Sigma'^a_e, \gamma'^a_v, \gamma'^a_e)$ and $G'^b = (\Sigma'^b_v, \Sigma'^b_e, \gamma'^b_v, \gamma'^b_e)$. The order of G^a and G'^a is n^a and the order of G^b and G'^b is n^b since the correspondences f^a and f^b are defined to be bijective (some null nodes in these graphs may have been added to consider insertion and deletion operations). We can only assure $\Delta_v^a = \Delta'^a_v$ and $\Delta_v^b = \Delta'^b_v$ yet Δ_v^a may be deferent of Δ_v^b and Δ'^a_v may be deferent of Δ'^b_v (and similarly for the edges). Moreover, we assume there is some level of intersection between both input node sets and also both output node sets, although it is not strictly necessary, and also it may happen that $n^a \neq n^b$. Note this intersection is imposed through mappings $\zeta: \Sigma_v^a \, x \, \Sigma_v^b \rightarrow \{0,1\}$ and $\zeta': \Sigma'^a_v \, x \, \Sigma'^b_v \rightarrow \{0,1\}$. Mapping $\zeta(v_i^a, v_j^a) = 1$ means v_i^a has to be considered the same node as v_j^b and 0 means they are not the same node. In the same way, mapping $\zeta'(v'^a_i, v'^a_j) = 1$ means v'^a_i has to be considered the same node as v'^b_j and 0 means they are not the same node. If this function is expressed in a matrix form, there is only one cell with a value of 1 in each row and column. Moreover, the number of 1's in the matrix is the number of nodes that are considered the same in both graphs.

Note these two mappings relate nodes but not edges since they may cause some edge inconsistencies. The problem at hand is to define a consensus correspondence $f^{a,b}: \Sigma_v^{a,b} \rightarrow \Sigma'^{a,b}_v$ given the four graphs G^a, G^b, G'^a and G'^b , bijections f^a and f^b and mappings ζ and ζ'. The set $\Sigma_v^{a,b}$ is composed of the union of sets Σ_v^a and Σ_v^b but the ones mapped by ζ are considered only once. In this case, some null nodes are added to have the possibility of deleting the whole graph. $n^{a,b}$is the cardinality of $\Sigma_v^{a,b}$ and $\Sigma'^{a,b}_v$. The set $\Sigma'^{a,b}_v$ is composed of the union of sets Σ'^a_v and Σ'^b_v where the elements mapped by ζ' are considered only once. For this set, some null nodes are added to have the possibility of inserting the whole graph. In this work, we do not want to find the attributed graphs $G^{a,b}$ and $G'^{a,b}$ that could represent the union of graphs G^a and G^b and also of graphs G'^a and G'^b. There are some methods, such as median graphs [21], dedicated to it. This is because we suppose the nature of graphs G^a and G^b and also of graphs G'^a and G'^b is different and, as commented, they have different attribute domains. The only operation that we do on these four graphs is to extend them with null nodes to have the same order $n^{a,b}$. The extended graphs are called $\hat{G}^a$, $\hat{G}^b$, $\hat{G}'^a$ and $\hat{G}'^b$. Accordingly to these graph extensions, the correspondences f^a and f^b are also extended to $\hat{f}^a$ and $\hat{f}^b$ such that the new null nodes are mapped to each other. Thus, $EditCost(G^a, G'^a, f^a) = EditCost(\hat{G}^a, \hat{G}'^a, \hat{f}^a)$ and $EditCost(G^b, G'^b, f^b) = EditCost(\hat{G}^b, \hat{G}'^b, \hat{f}^b)$.

We seek for the correspondence $f^{a,b}$ such that $EditCost(\hat{G}^a, \hat{G}'^a, f^{a,b})$ and $EditCost(\hat{G}^b, \hat{G}'^b, f^{a,b})$ are minimised, and also that it is restricted to be a mean of bijections f^a and f^b. The degree of restriction depends on weight λ, which is a real positive number. Moreover d_H is the Hamming distance.

$$f^{a,b^*}_{\lambda} = \operatorname{argmin}_{\forall f^{a,b}: \Sigma_v^{a,b} \rightarrow \Sigma'^{a,b}_v} \left\{ \begin{matrix} EditCost(\hat{G}^a, \hat{G}'^a, f^{a,b}) + \mathrm{EditC}ost(\hat{G}^b, \hat{G}'^b, f^{a,b}) + \\ \lambda \cdot [\, d_H(\hat{f}^a, f^{a,b}) + d_H(\hat{f}^b, f^{a,b})] \end{matrix} \right\} \quad (2)$$

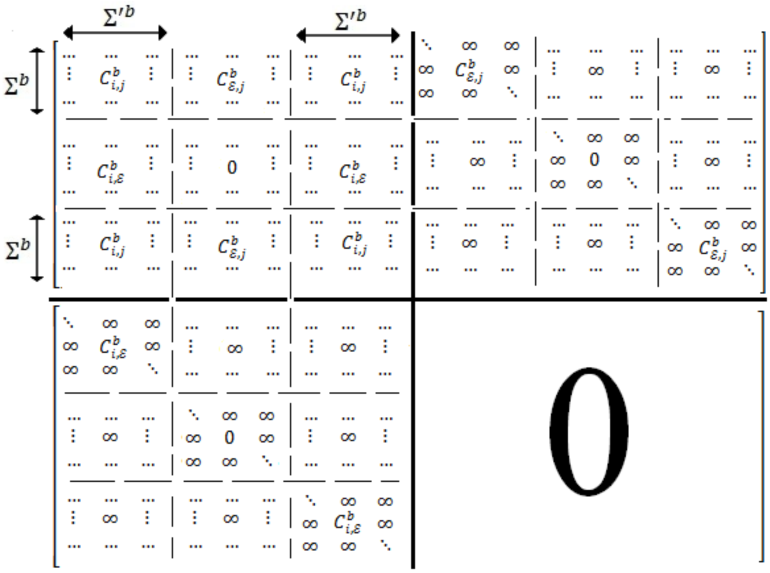


Fig. 3. Generalised cost matrix C^bgiven graphs G^b and G'^b

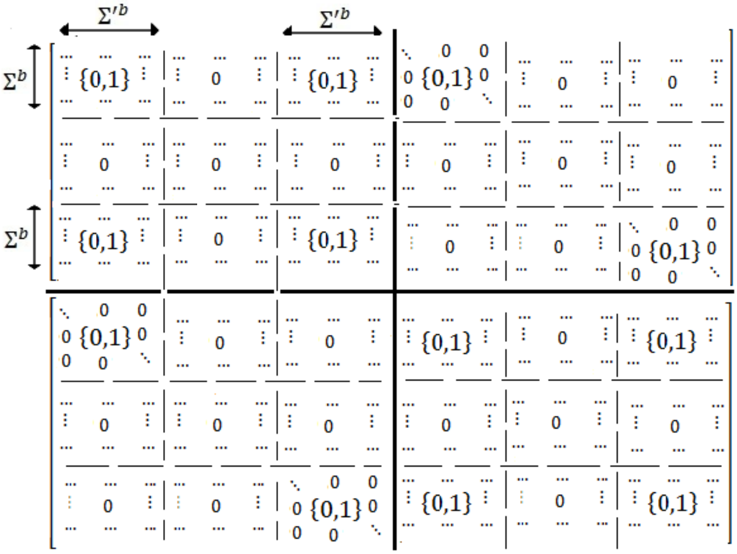


Fig. 4. Generalised correspondence F^b

In the next section, we demonstrate that minimising the above functional is the same than minimising the following one expressed through matrices instead of functions.

$$f^{a,b}{}^{*}_{\lambda} = \mathrm{argmin}_{\forall f^{a,b}:\, \Sigma_v^{a,b} \rightarrow \Sigma'^{a,b}_v} \left\{ C^a{}_{f^{a,b}} + C^b{}_{f^{a,b}} + \lambda \cdot \left[\left(\mathbf{1}_{f^{a,b}} - F^a{}_{f^{a,b}} \right) + \left(\mathbf{1}_{f^{a,b}} - F^b{}_{f^{a,b}} \right) \right] \right\} \quad (3)$$

where C^a and C^b are extended cost matrices and F^a and F^b are extended correspondence matrices. **1** is a matrix with all ones. The four matrices have been extended to have $n^{a,b} x\, n^{a,b}$ cells. That is, to assure the whole combinations of substituting, deleting and inserting nodes is possible for the whole nodes. Nevertheless, these matrix extensions have to consider mappings ζ and ζ' between nodes. Figure 3 shows the extended cost matrix C^b. Rows have been split depending on nodes belong to $\Sigma_\mathrm{v}^\mathrm{a}$, $\Sigma_\mathrm{v}^\mathrm{b}$, both or none of them. And similarly for columns and nodes in $\Sigma'^\mathrm{a}_\mathrm{v}$ and $\Sigma'^\mathrm{b}_\mathrm{v}$. Correspondingly, figure 4 shows the extended matrix F^b.

The consensus method we propose is based on applying a linear solver such as the Hungarian method or the Jonker-Volgenant method to obtain ${f^{a,b}}^*_\lambda$ as it is defined in equation 3, but using the following matrix and the BP algorithm [8].

$$H = C^a + C^b + \lambda \cdot [\mathbf{2} - (F^a + F^b)] \qquad (4)$$

where **2** is a matrix with all two.

4 Minimising the Functional Through Matrices

We have to demonstrate that $EditCost(\hat{G}^a, \hat{G}'^a, f^{a,b}) + \mathrm{Edit}Cost(\hat{G}^b, \hat{G}'^b, f^{a,b}) + \lambda \cdot [\,\mathrm{d_H}(\hat{f}^a, f^{a,b}) + \mathrm{d_H}(\hat{f}^b, f^{a,b})]$ equals to ${C^a}_{f^{a,b}} + {C^b}_{f^{a,b}} + \lambda \cdot [(\mathbf{1}_{f^{a,b}} - {F^a}_{f^{a,b}}) + (\mathbf{1}_{f^{a,b}} - {F^b}_{f^{a,b}})]$. By construction, $EditCost(\hat{G}^a, \hat{G}'^a, f^{a,b}) + \mathrm{Edit}Cost(\hat{G}^b, \hat{G}'^b, f^{a,b}) = \mathrm{C}^\mathrm{a}_{f^{a,b}} + \mathrm{C}^\mathrm{b}_{f^{a,b}}$ and also $\mathbf{1}_{f^{a,b}} = n^{a,b}$, for this reason we only need that,

$$\mathrm{d_H}(\hat{f}^a, f^{a,b}) + \mathrm{d_H}(\hat{f}^b, f^{a,b}) = 2 \cdot n^{a,b} - {F^a}_{f^{a,b}} - {F^b}_{f^{a,b}} \qquad (5)$$

Table 1. Four cases of nodes on $\hat{G}^a$ respectcorrespondences $\hat{f}^a$ and $\hat{f}^b$

Respect $\hat{f}^a$	**Respect $\hat{f}^b$**	**Nodes in $\Sigma_\mathrm{v}^\mathrm{a,b}$**	**$\mathrm{d_H}(\hat{f}^a, f^{a,b})$**	**$\mathrm{d_H}(\hat{f}^b, f^{a,b})$**	**${F^a}_{f^{a,b}}$**	**${F^b}_{f^{a,b}}$**
$\hat{f}^a(\mathrm{v}_\mathrm{i}^\mathrm{a}) \neq f^{a,b}(\mathrm{v}_\mathrm{i}^\mathrm{a})$	$\hat{f}^b(\mathrm{v}_\mathrm{i}^\mathrm{b}) \neq f^{a,b}(\mathrm{v}_\mathrm{i}^\mathrm{b})$	A	A	A	0	0
$\hat{f}^a(\mathrm{v}_\mathrm{i}^\mathrm{a}) \neq f^{a,b}(\mathrm{v}_\mathrm{i}^\mathrm{a})$	$\hat{f}^b(\mathrm{v}_\mathrm{i}^\mathrm{b}) = f^{a,b}(\mathrm{v}_\mathrm{i}^\mathrm{b})$	B	B	0	0	B
$\hat{f}^a(\mathrm{v}_\mathrm{i}^\mathrm{a}) = f^{a,b}(\mathrm{v}_\mathrm{i}^\mathrm{a})$	$\hat{f}^b(\mathrm{v}_\mathrm{i}^\mathrm{b}) \neq f^{a,b}(\mathrm{v}_\mathrm{i}^\mathrm{b})$	C	0	C	C	0
$\hat{f}^a(\mathrm{v}_\mathrm{i}^\mathrm{a}) = f^{a,b}(\mathrm{v}_\mathrm{i}^\mathrm{a})$	$\hat{f}^b(\mathrm{v}_\mathrm{i}^\mathrm{b}) = f^{a,b}(\mathrm{v}_\mathrm{i}^\mathrm{b})$	D	0	0	D	D
TOTAL:		**$n^{a,b}$**	**A+B**	**A+C**	**C+D**	**B+D**

holds. Then, we can confirm that it is valid to use equation 4 to solve our problem. Considering the relation between correspondences $f^{a,b}$, $\hat{f}^a$ and $\hat{f}^b$, we have split the node set $\Sigma_\mathrm{v}^\mathrm{a,b}$ in four cases. The first two rows of table 1 show these four combinations. The third row shows the supposed number of nodes that hold this case (we use A, B, C and D to represent these number of nodes). Clearly, the addition of these four values is the number of nodes in $\Sigma_\mathrm{v}^\mathrm{a,b}$. In the next two columns, we show the Hamming distance between the subsets of nodes that hold each specific case. And in the last two columns, we show the obtained values of the correspondence matrices applied only to the specific nodes.

Considering the third and fourth columns, we deduct $\mathrm{d_H}(\hat{f}^a, f^{a,b}) + \mathrm{d_H}(\hat{f}^b, f^{a,b}) = 2\mathrm{A} + \mathrm{B} + \mathrm{C}$. Besides, considering the last two columns, we obtain ${F^a}_{f^{a,b}} + {F^b}_{f^{a,b}} = 2\mathrm{D} + \mathrm{B} + \mathrm{C}$. Therefore, $2 \cdot n^{a,b} - {F^a}_{f^{a,b}} - {F^b}_{f^{a,b}} = 2\mathrm{A} + 2\mathrm{B} + 2\mathrm{C} + 2\mathrm{D} - 2\mathrm{D} - \mathrm{B} - \mathrm{C} = 2\mathrm{A} + \mathrm{B} + \mathrm{C}$ which is exactly the same number than the one obtained for $\mathrm{d_H}(\hat{f}^a, f^{a,b}) + \mathrm{d_H}(\hat{f}^b, f^{a,b})$■

5 Experimental Validation

5.1 Database Used

To validate our method, we decided to use the "Tarragona Exteriors" dataset [17], defined through five public image databases called "BOAT", "EAST_PARK", "EAST_SOUTH", "RESIDENCE" and "ENSIMAG [22]. These databases are composed of a sequence of images taken from the same object, but from different positions and using a different zoom. Together with the images, the homography estimations h^i that convert the first image (img00) of the set into the other ones (img01 through img10) are provided. From each of the images, the 50 most reliable salient points were extracted using 5 methodologies: FAST, HARRIS, MINEIGEN, SURF (native Matlab 2013b libraries) and SIFT (own library). From these five sets of salient points, we were able to build five representative graph of each image, where the nodes represented the position of the salient points, and the edges were conformed using the Delaunay triangulation method. Notice that the key difference between the salient point data and the graph data is solely the addition of the edges, since the location and features of both the salient points and the graph's nodes is the same.

Between the first image (img00) of the sequence and the other ten images (img01 through img10), we computed correspondences using the five different extracted structures and four different matching functions which are: a) the Matlab's *MatchFeatures* function for salient points' matching (native Matlab 2013b libraries) and the *FastBipartite* function for graph matching using b) node, c) degree and d) clique local sub-structures (own library). Note that for the *MatchFeatures* function, the *MaxRatio* parameter was set to 1 to find as many mappings as possible, although we removed the non-bijective labellings, since this function often maps a salient point more than twice. Thus, the database has a total of 5 sequences × 10 pairs of images × 5 extractor methods × 4 matching options = 1000 quartets Q_i composed of two structures S_i^1 and S_i^2 (each representing the salient point's location and features plus the graph's edges) for two given images, one correspondence f^i and one homography h^i, resulting in $Q_i = \{S_i^1, S_i^2, f^i, h^i\}$, where i ∈ $[1 \ldots 1000]$.

5.2 Results and Interpretation

The purpose of this section is to show that performing a consensus using FBP ([6] and [7]) and considering the local sub-structure (node, degree or clique) is better than performing a consensus on the set of points as done in [1]. Notice that for the consensus of the graph matching, we only show the configuration that delivered the best results. C_{vs} is the normalised Euclidean distance and $C_{es} = 0$ due to edges do not have attributes. Node: $K_v = 50$, Degree: $K_v = K_e = 50$ and Clique: $K_v = K_e = 250$.

Table 2. Average number of correct inlier mappings obtained by our consensus strategy with sub-structures Node, Degree and Clique combining two feature extractors ($E_{n,m}$). We have added the number of inliers published in [1] in which the consensus method is based on points.

		$E_{1,2}$	$E_{1,3}$	$E_{1,4}$	$E_{1,5}$	$E_{2,3}$	$E_{2,4}$	$E_{2,5}$	$E_{3,4}$	$E_{3,5}$	$E_{4,5}$	Average
	CLIQUE	**84**	**78**	**104**	**70**	**96**	**116**	**89**	**128**	**57**	**160**	**88.2**
	DEGREE	74	72	82	60	68	109	61	111	52	42	73.1
	NODE	70	65	84	63	56	89	70	62	42	37	63,8
BOAT	POINTS	10	10	48	9	9	50	9	46	9	66	26,6
	CLIQUE	**35**	**27**	**84**	**36**	22	**94**	**43**	**83**	**41**	**85**	**55**
	DEGREE	29	**27**	31	27	**33**	42	42	35	40	32	33.8
EAST	NODE	11	23	43	35	18	38	32	46	37	46	32.9
PARK	POINTS	2	1	38	4	2	47	1	44	2	67	20.8
	CLIQUE	8	**19**	20	12	**16**	27	11	**27**	10	30	18
	DEGREE	**11**	7	**34**	**18**	3	**29**	**14**	25	**12**	30	**18.3**
EAST	NODE	9	2	18	10	4	17	10	19	**12**	18	11.9
SOUTH	POINTS	1	1	22	1	1	20	1	22	1	**32**	10.2
	CLIQUE	**32**	**24**	**124**	**27**	**25**	**129**	**26**	**122**	**23**	**117**	**64.9**
	DEGREE	16	19	36	26	12	26	20	29	21	21	22.5
	NODE	10	15	24	15	12	24	16	24	19	17	17.6
RESID	POINTS	1	1	59	1	2	51	1	39	1	106	26.2
	CLIQUE	**4**	3	**47**	4	**5**	**44**	**6**	43	**5**	52	**21.3**
	DEGREE	**4**	**5**	34	3	1	37	5	**35**	3	35	16.2
ENSI	NODE	2	3	30	**5**	3	37	**6**	31	**5**	36	15.8
MAG	POINTS	1	1	42	1	1	38	1	36	1	**53**	17.5
	CLIQUE	**32.6**	**30.2**	75.8	**29.8**	**32.8**	**82**	**35**	**80.6**	**27.2**	68.8	
	DEGREE	25.8	23.8	**147.6**	27.6	23.8	123.6	28.4	120	25	**170.8**	
Av	NODE	20.4	21.6	39.8	25.6	18.6	41	26.8	36.4	23	30.8	
	POINTS	3	2.8	41.8	3.2	3	41.2	2.6	37.4	2.8	64.8	

Table 2 shows the number of correct mappings found after applying the consensus for each possible combination of the five feature extractors. Consider 1=FAST, 2=HARRIS, 3=MINEIGEN, 4=SURF, 5=SIFT.

In this experimental validation, the results show that when performing a consensus between two graph matching correspondences and using the new model, we obtain a bigger improvement than when performing salient point's correspondence consensus using the previous method. For example, in the case of the "BOAT" dataset using $E_{3,4}$, the clique consensus obtained 128 correct inliers (31 more than the sum of the individual correspondences) whereas the previous consensus obtained 46 correct inliers, even though the sum of the individual methods is of 73 correct inliers.

6 Conclusions and Further Work

We have presented a consensus method that obtains a correspondence between two attributed graphs given two correspondences between two pairs of attributed graphs generated by separate entities. The method is based on a generalisation of the BP algorithm. We have shown in the experimental section the validity of our method. In this paper, we are able to show that graph representation and matching helps not only to increase the number of correct mappings in the initial proposals, but also to increase the improvement of a consensus correspondence respect representing the objects as a set of points. As a future work, we propose to extend this method such that the consensus can be applied to several correspondences and not only on a pair of them as done in [23] for the salient point case and in [24] for the clustering case. To do so, we are investigating on weighting and voting consensus methods. Nevertheless, the method we present is the first step for the several-correspondences method. Since we have defined the basic mechanism of the method, a several-correspondences method could be applied simply by using the 2-correspondence method iteratively. As we have seen, our method achieves a good accuracy when there are discrepancies between both labellings. Due to in the several-correspondences case the number of discrepancies would increase, our first intuition is that our method would obtain a good consensus correspondence.

References

1. Moreno-García, C.F., Serratosa, F.: Weighted Mean Assignment of a Pair of Correspondences Using Optimisation Functions. In: Fränti, P., Brown, G., Loog, M., Escolano, F., Pelillo, M. (eds.) S+SSPR 2014. LNCS, vol. 8621, pp. 301–311. Springer, Heidelberg (2014)
2. Franek, F., Jiang, X., He, C.: Weighted Mean of a Pair of Clusterings. Pattern Analysis Applications (2012)
3. Sanromà, G., Alquézar, R., Serratosa, F., Herrera, B.: Smooth Point-set Registration using Neighbouring Constraints. Pattern Recognition Letters 33, 2029–2037 (2012)
4. Sanromà, G., Alquézar, R., Serratosa, F.: A New Graph Matching Method for Point-Set Correspondence using the EM Algorithm and Softassign. Computer Vision and Image Understanding 116(2), 292–304 (2012)
5. Gold, S., Rangarajan, A.: A Graduated Assignment Algorithm for Graph Matching. Pattern Analysis and Machine Intelligence 18(4), 377–388 (1996)
6. Serratosa, F.: Speeding up Fast Bipartite Graph Matching through a New Cost Matrix. International Journal of Pattern Recognition & Artificial Intelligence 29(2) (2015)
7. Serratosa, F.: Fast Computation of Bipartite Graph Matching. Pattern Recognition Letters 45, 244–250 (2014)
8. Riesen, K., Bunke, H.: Approximate Graph Edit Distance Computation by Means of Bipartite Graph Matching. Image Vision Comput. 27(7), 950–959 (2009)

9. Serratosa, F., Cortés, X.: Interactive Graph-Matching using Active Query Strategies. Pattern Recognition 48, 1360–1369 (2015)
10. Cortés, X., Serratosa, F.: An Interactive Method for the Image Alignment problem based on Partially Supervised Correspondence. Expert Systems With Applications 42(1), 179–192 (2015)
11. Papadimitriou, C., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Dover Publications (1998)
12. Kuhn, H.W.: The Hungarian Method for the Assignment Problem Export. Naval Research Logistics Quarterly 2(1-2), 83–97 (1955)
13. Jonker, R., Volgenant, T.: Improving the Hungarian Assignment Algorithm. Operations Research Letters 5(4), 171–175 (1986)
14. Timofte, R., Van Gool, L.: Adaptive and Weighted Collaborative Representations for Image Classification. Pattern Recognition Letters (Available Online 2014)
15. Saha, S., Ekbal, A.: Combining Multiple Classifiers using Vote Based Classifier Ensemble Technique for Named Entity Recognition. Data & Knowledge Engineering 85, 15–39 (2013)
16. Serratosa, F., Cortés, X., Solé, A.: Component Retrieval Based on a Database of Graphs for Hand-Written Electronic-Scheme Digitalisation. Expert Systems with Applications 40, 2493–2502 (2013)
17. Riesen, K., Bunke, H., Fischer, A.: Improving Graph Edit Distance Approximation by Centrality Measures. In: International Conference on Pattern Recognition (2014)
18. Cortés, X., Serratosa, F.: Learning Graph-Matching Edit-Costs based on the Optimality of the Oracle's Node Correspondences. Pattern Recognition Letters (2015)
19. `http://deim.urv.cat/~francesc.serratosa/databases/`
20. Sanfeliu, A., Fu, K.S.: A Distance Measure between Attributed Relational Graphs for Pattern Recognition. IEEE Transactions on Systems, Man, and Cybernetics 13(3), 353–362 (1983)
21. Gao, X., Xiao, B., Tao, D., Li, X.: A Survey of Graph Edit Distance. Pattern Analysis and Applications 13(1), 113–129 (2010)
22. Solé, A., Serratosa, F., Sanfeliu, A.: On the Graph Edit Distance Cost: Properties and Applications. Intern. Journal of Pattern Recognition and Artificial Intell. 26(5) (2012)
23. Ferrer, M., Valveny, E., Serratosa, F.: Median Graph: A New Exact Algorithm using a Distance Based on the Maximum Common Subgraph. Pattern Recognition Letters 30(5), 579–588 (2009)
24. `http://www.featurespace.org`
25. Moreno-García, C.F., Serratosa, F., Cortés, X.: Iterative Versus Voting Method to Reach Consensus Given Multiple Correspondences of Two Sets. In: IbPRIA 2015 (2015)
26. Franek, L., Jiang, X.: Ensemble Clustering by means of Clustering Embedding in Vector Space. Pattern Recognition 47(2), 833–842 (2014)