MORENO-GARCÍA, C.F. and SERRATOSA, F. 2017. Correspondence consensus of two sets of correspondences through optimisation functions. *Pattern analysis and applications* [online], 20(1), pages 201-213. Available from: https://doi.org/10.1007/s10044-015-0486-y

Correspondence consensus of two sets of correspondences through optimisation functions.

MORENO-GARCÍA, C.F. and SERRATOSA, F.

2017

Correspondence Consensus of Two Sets of Correspondences through Optimisation Functions[1]

Carlos Francisco Moreno-García & Francesc Serratosa

carlosfrancisco.moreno@estudiants.urv.cat
francesc.serratosa@urv.cat
Universitat Rovira i Virgili, Spain

Abstract: We present a consensus method which, given two correspondences between sets of elements generated by separate entities, enounces a final correspondence consensus considering the existence of outliers. Our method is based on an optimisation technique that minimises the cost of the correspondence while forcing (to the most) to be the mean correspondence of the two original ones. The method decides the mapping of the elements for which the original correspondences disagree and returns the same element mapping when both correspondences agree. We first show the validity of the method through an experiment in ideal conditions based on palmprint identification and subsequently, present two practical experiments based on image retrieval.

Keywords: Assignment Problem, Consensus strategy, Weighted Mean, Hamming Distance, Optimisation.

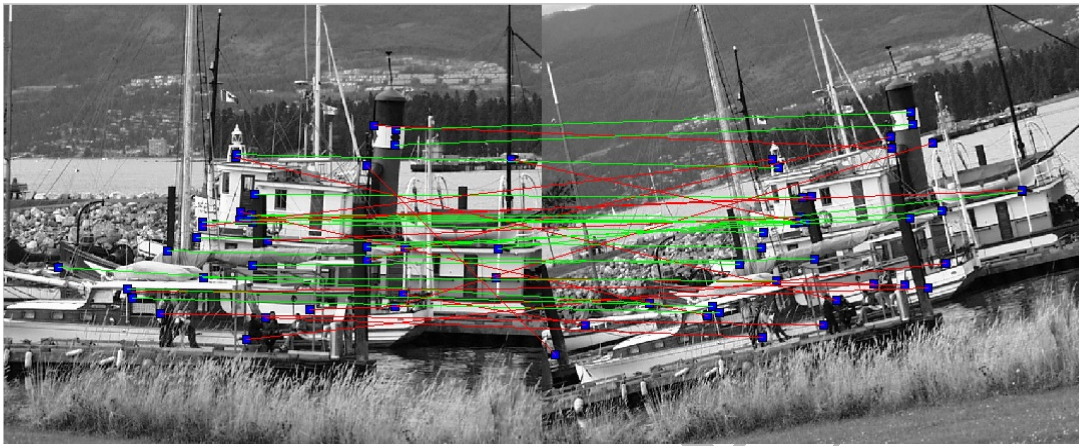
1. Introduction

When two different strategies decide to solve the assignment problem (finding the correspondence or mapping between two sets of elements), differences on the elements' mapping may occur. These differences appear due to several factors. Between them, we could cite the following. a) One of the strategies gives more importance to some of the element's attributes, and the other strategy believes other ones are more important. If our scenario is based on an automatic method, these differences are gauged by the features or the weights on these features. Contrarily, if the scenario is based on a human-machine interaction (for instance, semi-automatic medical image recognition), the strategy is based on the experience of a human specialist. For instance, if such elements represent regions of segmented images, one subject may think the area is more important than the colour, and the other one may think the opposite. b) Another factor that influences the elements' mapping could be that the elements' assignment problem is computed with a suboptimal algorithm, and different non-exact assignments can appear.
Some examples of state of the art of correspondence methods are the classical ones such as [1], the ones that discard outlier elements [2], [3] or the ones that characterise the set of elements into an attributed graph [4], [5]. Although some manual methods [6] have been presented to improve the correspondences made by a single matching algorithm, for these three scenarios, a consensus system could intervene as a third party to decide the final elements' assignment when discrepancies appear, especially as the number of involved elements increase.

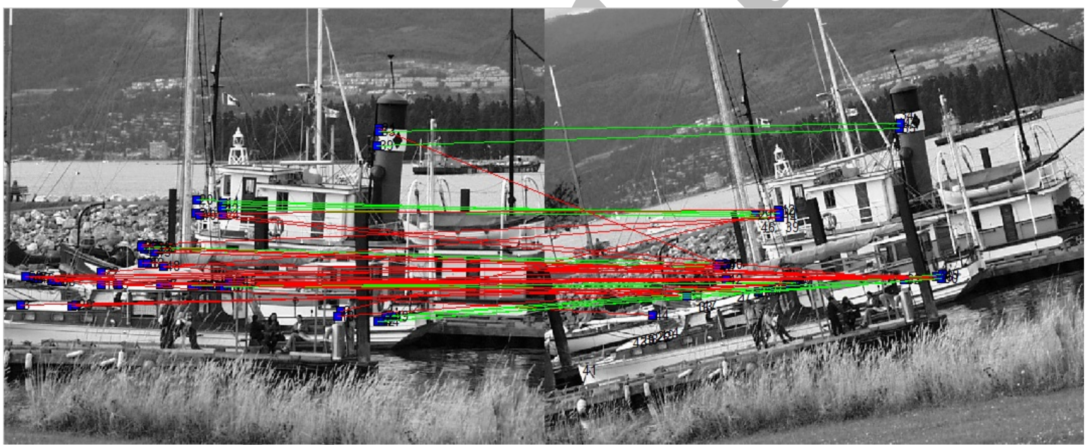
The most practical form to understand the advantages of our approach is through an image recognition scenario. We define salient points as image locations that can be robustly detected among different instances of the same scene, with varying imaging conditions. Salient points play the role of parts of the image that result in the elements to be matched. These points can be corners (intersection of two edges) [7] maximum curvature points [8] or isolated points of maximum or minimum local intensity [9]. There is an evaluation of the most competent approaches in [10]. Figure 1 shows an image with the salient points detected by SIFT [11] and SURF [12] methods, and the correspondences deducted by the *FastBipartite* algorithm [13] [14] and the matching function *MatchFeatures* contained in Matlab. We only show the mapped points (usually called inliers), excluding the non-matched points (outliers). Based on a previously established ground truth, we depict green lines representing correctly detected correspondences, and red lines representing correspondences that are different from the ground truth. Since the salient point extractor

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

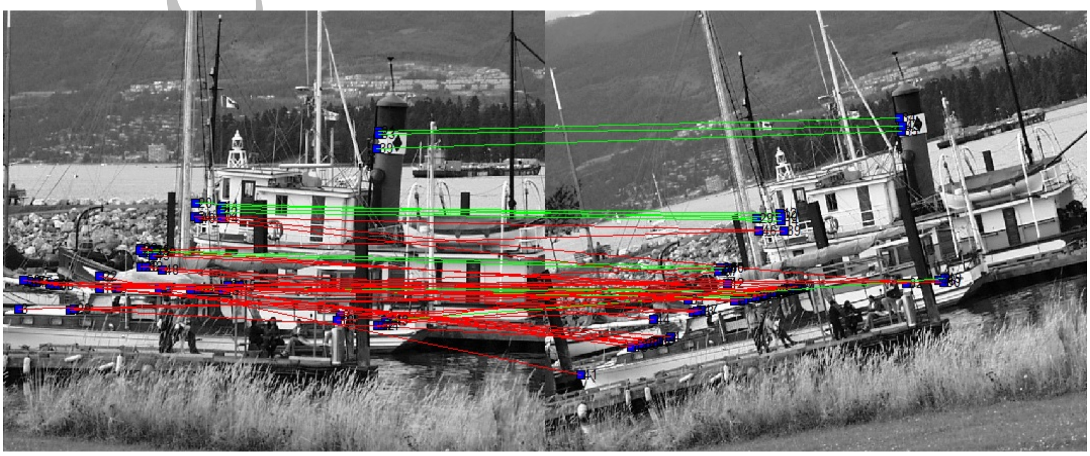
is different, between figures 1(a) and 1(b) not only salient points are located in different positions but also different features are extracted. For this reason, the obtained correspondences are clearly different. If we compare figures 1(b) and 1(c), even if the initially extracted salient points have the same location and feature values, different correspondences are obtained since different matching algorithms were used. Therefore, some of the mapped points are different.



(a) Points detected by SURF method and matched by *FastBipartite* algorithm



(b) Points detected by SIFT method and matched by *FastBipartite* algorithm



(c) Points detected by SIFT method and matched by *MatchFeatures* Matlab function

Figure 1. Correspondences deducted by different feature extractors and matching algorithms. Green lines are correct correspondences and red ones are incorrect ones.

Our method is intended to be applied on two scenarios. The first one, when either two human specialists or two suboptimal algorithms seek for salient points or parts of an object and also their correspondences (or in an interactive way [15], []), (figure 1a and 1b). The second one, when the salient points have been previously computed, and both specialists receive the same input (figure 1b and 1c). The main difference resides in the fact that some points can only be considered by one of the specialists and so, cannot be mapped by the other specialist. Consider the example presented above. The aim of our method is to deduct a set of points on both images, and a correspondence between them such that the number of correct correspondences is greater than any of both methods. Figure 2 shows the result of applying our consensus method to examples of Figures 1(b) and 1(c) (second scenario). Our method must be able to deal with points that appear in both original correspondences and points that only appear in one of the correspondences. The number of correct correspondences (green lines) has increased with respect to the ones in figures 1(b) and 1(c) and the number of incorrect correspondences (red lines) has decreased.

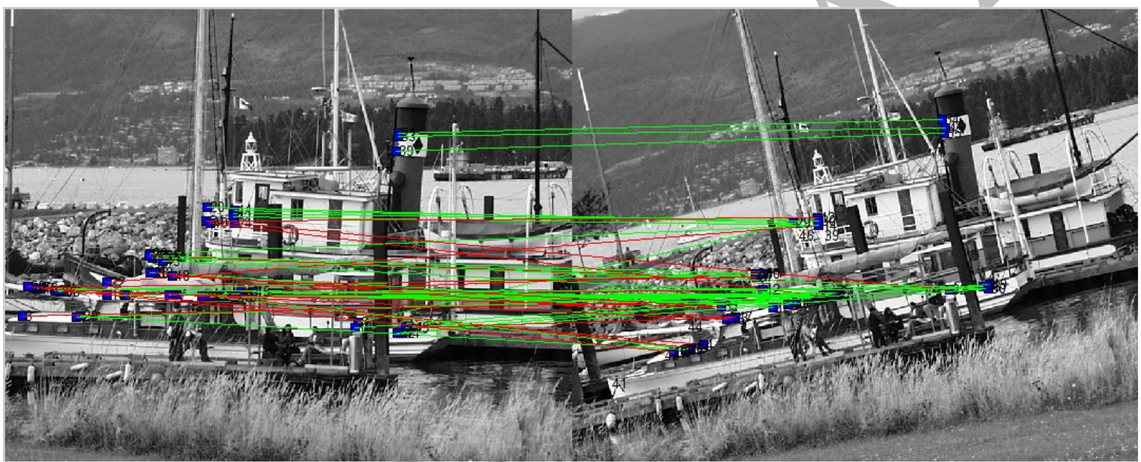


Figure 2. Consensus of examples on Figures 1(b) and 1(c). Green lines: correct correspondences. Red lines: incorrect ones.

A key factor of our method resides in modelling the consensus correspondence as a weighted mean of the two initial ones. That is because the aim of consensus methodologies is to find an assignment that is as close as possible to both initial ones, but that minimises the final assignment's cost as well. Thus, our methodology is inspired in the one presented in [16], which obtains a weighted mean clustering consensus given a pair of clusterings. Our method, and also [16], does not restrict the correspondence consensus to be a strict mean but a weighted mean. Other methods to perform clustering consensus are [17], where a final cluster is obtained based on a similarity graph and [18], where the least square algorithm are used. Moreover, in [19] consensus clustering is used to classify genetic diseases. All of the aforementioned methods are closely related to unsupervised machine learning methods [20].

The main drawback of intending to find a weighted mean assignment resides in the large number of possible solutions. One of the most well-known and practical options to reduce the complexity of a combinatorial calculation 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 [21]. Functions involved in an optimisation problem can be either conformed by continuous values or discrete values, often called as combinatorial scenarios. These second scenarios have been largely studied and applied for matching problems, particularly the case of the Hungarian Algorithm [1]. 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 [22]. These methods learn some weights that gauge the importance of each classifier and also the sample through several techniques such as voting [23]. Nevertheless, they cannot be directly adapted to our problem, since their output is a class index and our output must be a whole correspondence between two sets of points. The method we propose could be applied to obtain a common correspondence between sets [22], [23] or to define a median that represents different sets in only one structure [24], [25]. In the application domain, this method cold be used to latent palmprint matching [30] or hand-written electronic schemes [35].

The rest of the paper is organised as follows. In section 2, we present the basic definitions and we propose a practical example. In section 3, we define our method to obtain a correspondence consensus given two different correspondences calculated from two sets of points. In section 4, we present the experimental validation, which is split in three main experiments. We conclude the paper and propose the future work in section 5. An initial study of these problems has been submitted [28]. In this paper, we have added much more experiments and the possibility of having outliers in both sets of elements; therefore, we have modelled more real sets of applications.

2. Basic Definitions

Given two sets of elements and a correspondence between them, we define inlier elements as the elements that are mapped by the correspondence and outlier elements as the ones that are not. In the case we have two correspondences between the same set of elements, the number of inliers and outliers that generated both correspondences in these sets can be different. To formalise this situation, it is usual to consider some extra elements in both sets (usually called null elements) to discern between inliers and outliers. Thus, the elements in the domain set have to be considered outliers if mapped to null elements in the codomain set. In the same way, the elements in the codomain set have to be considered outliers if their argument value is a null element. More formally, given a set of elements $G^1 = \{g_1^1, g_2^1, \dots, g_{\mu^1}^1\}$ where the elements $g_i^1 = (m_i^1, a_i^1)$, being $m_i^1 \in \Sigma^1$ (where Σ^1 denotes a set of indices) and $a_i^1 \in T$ (where T is the domain of the attribute of the elements), a correspondence f can be established between G^1 and another set of elements G^2, with similar domain T. It is usual to consider that G^1 and G^2 have been previously enlarged with an enough number of null elements such that all combinations of mappings between inliers and outliers are possible. Therefore, if the initial cardinalities were μ^1 and μ^2, after the enlargement both sets have $\mu = \mu^1 + \mu^2$ elements. This correspondence f is understood as a mapping that proposes a match $f: \Sigma^1 \to \Sigma^2$ from elements of G^1 to elements of G^2. f can be defined through a so called correspondence matrix F, where $F[i,j] = 1$ when $f(m_i^1) = m_j^2$ and $F[i,j] = 0$ otherwise.

We define the cost of a correspondence $Cost(G^1, G^2, f)$ as the addition of individual element costs in a similar way as in the Graph Edit Distance [29],

$$Cost(G^1, G^2, f) = \sum_{i=1}^{\mu} c\left(m_i^1, f(m_i^1)\right) \tag{1}$$

where c is defined as a distance function over the domain of attributes T. Distance c is application dependent [29] and it has to consider both the case where two original elements are mapped, and the one where one of them is a null element ($c = 0$ if the mapped elements are null elements). We suppose that $c \in [0,1]$ since it makes easier the setting of some weights of our method.

The distance between sets $d_S(\cdot,\cdot)$, which also delivers the minimum cost of all the correspondences, is a function defined as

$$d_S(G^1, G^2) = min\{Cost(G^1, G^2, f)\} \; \forall\, f: \Sigma^1 \to \Sigma^2 \tag{2}$$

The correspondence that obtains this distance is known as the optimal correspondence f^*, and is defined as

$$f^* = argmin_{\forall f: \Sigma^1 \to \Sigma^2} \{Cost(G^1, G^2, f)\} \tag{3}$$

We convert this linear minimisation problem into an assignment problem [1] in which any correspondence f is related with a combination. With the calculation of a cost matrix $\boldsymbol{C}[i,j] = c(m_i^1, m_j^2)$, we can convert equation 3 into

$$f^* = argmin_{\forall f: \Sigma^1 \to \Sigma^2} \{\mathbf{C}_f\} \tag{4}$$

where $\mathbf{C}_f$ is the cost of the combination f (or correspondence in the set domain) applied to matrix $\boldsymbol{C}$. That is

$$\mathbf{C}_f = \sum_{i=1}^{\mu} \boldsymbol{C}[i,k] \; where \; f(m_i^1) = m_k^2 \\ s.t. \forall i \neq j, f(m_i^1) \neq f(m_j^1) \tag{5}$$

Assume f^a and f^b are two correspondence functions between sets $G^1 = \{g_1^1, g_2^1, \dots, g_\mu^1\}$ and $G^2 = \{g_1^2, g_2^2, \dots, g_\mu^2\}$. We define the Hamming Distance $d_H(\cdot)$ between the correspondences f^a and f^b as

$$d_H(f^a, f^b) = \sum_{i=1}^{\mu} \left(1 - \delta\left(f^a(m_i^1), f^b(m_i^1)\right)\right) \tag{6}$$

where function δ is the well-known Kronecker Delta.

$$\delta(a,b) = \begin{cases} 0 \; if \; a \neq b \\ 1 \; if \; a = b \end{cases} \tag{7}$$

In its more general form, the mean of two elements e^a and e^b has been defined as an element $\bar{e}$ such that $d(e^a, \bar{e}) = d(e^b, \bar{e})$ and $d(e^a, e^b) = d(e^a, \bar{e}) + d(\bar{e}, e^b)$, being d any distance measure defined on the domain of these elements. Moreover, the concept of weighted mean is used to gauge the importance or the contribution of the involved elements. In this case, the most general definition is $d(e^a, \bar{e}) = \alpha$ and $d(e^a, e^b) = \alpha + d(\bar{e}, e^b)$, where α is a constant that controls the contribution of the elements and it holds $0 \leq \alpha \leq d(e^a, e^b)$. Finally, if we do not need to introduce the weighting factor α in our model, any element $\bar{e}$ is a weighted mean of two elements e^a and e^b if it holds $d(e^a, e^b) = d(e^a, \bar{e}) + d(\bar{e}, e^b)$. Note that elements e^a and e^b hold this condition so, they also are weighted means.
One of the most appropriate form to model a consensus scenario given two different proposals is the one done by [16], which defined the consensus as the weighted mean of these two proposals. The aim of [14] is to find the consensus clustering of a set of elements given two different clustering proposals applied to this set of elements. If we want to translate this model to our problem, we should find the weighted mean correspondences $\bar{f}$ given two different correspondences f^a and f^b. As commented in the previous paragraph, if we want $\bar{f}$ to be defined as a weighted mean correspondence of f^a and f^b the following restriction has to hold,

$$d_H(f^a, f^b) = d_H(f^a, \bar{f}) + d_H(\bar{f}, f^b) \tag{8}$$

As stated before, several correspondences $\bar{f}$ hold this condition, including f^a and f^b. To select one of the options, one choice would be a brute force method that obtains all possible combinations and selects the best one from the application point of view. A more optimal choice is a standard minimisation approach, to reduce the computational time. Standard minimisation approaches aim to find an optimal element e^* that globally minimises a specific function. Usually, this function is composed of an empirical risk $\nabla(e)$ plus a regularization term $\Omega(e)$ weighted by a parameter λ [27]. The empirical risk is the function to be minimised *per se*, and the regularisation term is a mathematical mechanism to impose some restrictions. Parameter λ weights how much these restrictions have to be imposed.

$$e^{*} = argmin_{\forall e} \{\nabla(e) + \lambda \cdot \Omega(e)\} \tag{9}$$

The aim of this paper is to present a method to find an approximation of the weighted mean correspondence given two correspondences. Therefore, we want to find $\bar{f}^{*}$ such that the following equation holds,

$$\bar{f}^{*} = argmin_{\forall f: \Sigma^1 \rightarrow \Sigma^2} \{ \lambda_C \cdot \nabla(f) + \lambda_H \cdot \Omega(f)\} \tag{10}$$

Although not strictly necessary, we present this equation with parameters λ_C and λ_H instead of only one parameter λ as in equation 9 to simplify some explanations and examples. In the next sub-section, we present a practical example. Afterwards, we explain functions $\nabla(f)$ and $\Omega(f)$.

2.1 Practical Example

To facilitate the understanding of our method, we propose the following example which we will comment throughout the paper. Suppose two sets G^1 and G^2 of 7 elements have attributes represented by Natural numbers (Table 1). Additionally, two correspondences f^a and f^b have been defined between these two sets (Figure 3). Note that only the first mapping is common to both correspondences with $f^a(m_1^1) = f^b(m_1^1)$. Moreover, elements m_3^1, m_6^1 and m_7^1 (m_3^2, m_4^2 and m_5^2) are outliers of G^1 (G^2) with respect to f^a. Elements m_4^1 and m_5^1 (m_6^2 and m_7^2) are outliers of G^1 (G^2) with respect to f^b. The Hamming distance is $d_H(f^a, f^b) = 7 - 1 = 6$. If we define the cost between elements g_i^1 and g_j^2 of both sets as $c(a_i^1, a_j^2) = |a_i^1 - a_j^2| / max(a_i^1, a_j^2)$, then the costs (equation 1) of these correspondences are $Cost(f^a) = 1.4$ and $Cost(f^b) = 2.7$.

Table 1. Attributes of sets $G^1$and G^2.

s = {1,2}	a_1^s	a_2^s	a_3^s	a_4^s	a_5^s	a_6^s	a_7^s
G^1	1	4	7	5	3	2	2
G^2	2	5	6	8	4	3	2

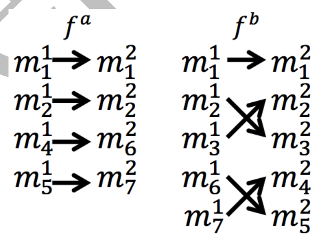


Figure 3. Example correspondences f^a and f^b between sets $G^1$and G^2.

3. Method

Our method defines the optimal correspondence $\bar{f}^{*}$ through equation 10 in which the Loss function and the Regularisation term are

$$\nabla(f) = Cost(G^1, G^2, f) \; and \; \Omega(f) = d_H(f^a, f) + d_H(f, f^b) - d_H(f^a, f^b) \tag{11}$$

With this equation we intend to minimise the obtained correspondence's cost (equation 1) while restricting it to be a weighted mean (equation 8). The degree of restriction depends on weights λ_C and λ_H. Note that by definition of a distance, $d_H(f^a, f) + d_H(f, f^b) - d_H(f^a, f^b) \geq 0$.

Since the aim of our method is to decide the correspondence closer to both human's correspondences (or automatically obtained correspondences), our strategy only seeks for the partial correspondence where both of the specialists disagree. The other partial correspondence in which both specialist decided the same point mapping is directly assigned as the final result. Understanding f^a and f^b as sets of element's mappings, we split them in two disjoint subsets such that $f^a = f'^a \cup f''^a$ and $f^b = f'^b \cup f''^b$. Subsets f'^a and f'^b are related with the partial correspondences where $f^a(m_i^1) = f^b(m_i^1)$ and subsets f''^a and f''^b are related with the other partial ones where $f^a(m_i^1) \neq f^b(m_i^1)$. This also means that the cost of both correspondences is $Cost(G^1, G^2, f^a) = Cost(G^1, G^2, f'^a) + Cost(G^1, G^2, f''^a)$ and $Cost(G^1, G^2, f^b) = Cost(G^1, G^2, f'^b) + Cost(G^1, G^2, f''^b)$. Additionally, we define $\Sigma = \Sigma' \cup \Sigma''$. The set of elements Σ' in G^1 is composed of the elements such that $f^a(m_i^1) = f^b(m_i^1)$ and the set of elements Σ'' in G^1 is composed of the elements such that $f^a(m_i^1) \neq f^b(m_i^1)$.
Thus, the obtained consensus correspondence $\bar{f}^*$ is a union of two partial correspondences, $\bar{f}^* = \bar{f}'^* \cup \bar{f}''^*$ where $\bar{f}'^* = f'^a = f'^b$ and $\bar{f}''^*$ that is the one defined by the following equation:

$$\bar{f}''^{*}_{\lambda_C,\lambda_H} = argmin_{\forall f'': \Sigma''^1 \to \Sigma''^2} \left\{ \begin{array}{c} \lambda_C \cdot Cost(G^1, G^2, f'') + \\ \lambda_H \cdot \left[d_H(f''^a, f'') + d_H(f'', f''^b) - d_H(f''^a, f''^b) \right] \end{array} \right\} \tag{12}$$

Weights λ_C and λ_H gauge how much the method trusts either the cost or the initial correspondences. If $\lambda_C = 0$ and $\lambda_H > 0$, we force the correspondence to be a union of both correspondences (incongruences are solved by selecting one of the mappings arbitrarily). Conversely, if $\lambda_C > 0$ and $\lambda_H = 0$, the method tends to minimize the cost without taking into account the initial proposals. In fact, in the experimental section, we show the that best combinations are not these extreme cases. To solve equation 12, we translate the linear minimisation problem to an assignment problem [29] as shown in equation 4 but, instead of the cost matrix $\boldsymbol{C}$, our method minimises expression H_{λ_C,λ_H} defined as

$$H_{\lambda_C,\lambda_H} = \lambda_C \cdot \boldsymbol{C}''_{f''} + \lambda_H \cdot [1 - F''^{a,b}]_{f''} \tag{13}$$

where expression $[\cdot]_{f''}$ is the cost of the correspondence f'' applied on matrix $[\cdot]$ (equation 5). $\boldsymbol{C}''[i,j] = c(m_i^1, m_j^2)$ with $m_i^1 \in \Sigma''^1$ and $m_j^2 \in \Sigma''^2$. Besides, $F''^{a,b} = F''^a + F''^b$, where F''^a and F''^b are the correspondence matrices (as defined at the beginning of section 2) corresponding to f''^a and f''^b respectively. 1 represents a matrix of all ones. Note that the number of rows and columns of matrices $\boldsymbol{C}''$, F''^a and F''^b is lower or equal than $\boldsymbol{C}$. As node mappings of f^a and f^b are more similar, the smaller the number nodes in Σ''^1 and Σ''^2 is, and so, the dimensions of $\boldsymbol{C}''$, F''^a and F''^b. This fact affects directly on the computational cost. Notice that we could define $\lambda_C = 1 - \lambda_H$ or viceversa, however, we decide to present both parameters separately to standardise some aspects of the experimental explanations. Considering equation 13, we obtain the following expression,

$$\bar{f}''^{*}_{\lambda_C,\lambda_H} = argmin_{\forall f''} \left\{ \left(H_{\lambda_C,\lambda_H} \right)_{f''} \right\} \tag{14}$$

Figure 4 shows values of matrices F^a, F^b, $\boldsymbol{C}''$, $F''^{a,b}$ and H_{λ_C,λ_H} (using $\lambda_C = 1$ and $\lambda_H = 1$) on our practical example. All matrices are squared given the need of the consensus correspondence to be bijective, however they have different dimensions. This is because the union of elements in the domain and codomain is 7, but elements m_1^1 and m_1^2 have not been taken into consideration since $f^a(m_1^1) = f^b(m_1^2)$. Note that outliers in G^1 (and G^2) generate null rows (and null columns) in F^a and F^b.

F^a

	1	2	3	4	5	6	7
1	1	0	0	0	0	0	0
2	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	1	0
5	0	0	0	0	0	0	1
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0

F^b

	1	2	3	4	5	6	7
1	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0
3	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	1	0	0	0
7	0	0	0	0	1	0	0

$F''^{(a,b)}$

	2	3	4	5	6	7
2	1	1	0	0	0	0
3	1	0	0	0	0	0
4	0	0	0	0	1	0
5	0	0	0	0	0	1
6	0	0	1	0	0	0
7	0	0	0	1	0	0

C''

	2	3	4	5	6	7
2	0,2	0,3	0,5	0	0,3	0,5
3	0,3	0,1	0,1	0,4	0,6	0,7
4	0	0,2	0,4	0,2	0,4	0,6
5	0,4	0,5	0,6	0,3	0	0,3
6	0,6	0,7	0,8	0,5	0,3	0
7	0,6	0,7	0,8	0,5	0,3	0

H

	2	3	4	5	6	7
2	0,2	0,3	1,5	1	1,3	1,5
3	0,3	1,1	1,1	1,4	1,6	1,7
4	1	1,2	1,4	1,2	0,4	1,6
5	1,4	1,5	1,6	1,3	1	0,3
6	1,6	1,7	0,8	1,5	1,3	1
7	1,6	1,7	1,8	0,5	1,3	1

Figure 4. Matrices F^a and F^b obtained from correspondences f^a and f^b between sets G^1 and G^2 and the obtained $\boldsymbol{C}''$,$F''^{a,b}$ and $H_{\lambda_C,\lambda_H}(\lambda_C = 1 \text{ and } \lambda_H = 1)$.

Several algorithms can be used to minimise equation 14, for instance the Hungarian algorithm [1]. As a result, the cost of the obtained consensus becomes,

$$\boldsymbol{C}_{\bar{f}^*_{\lambda_C,\lambda_H}} = \boldsymbol{C}''_{\bar{f}''^*_{\lambda_C,\lambda_H}} + Cost\left(G^1, G^2, f'^a\right) \tag{15}$$

In the next sub-section we demonstrate equations 12 and 14 minimise the same approximation of the consensus correspondence $\bar{f}''^*_{\lambda_C,\lambda_H}$ for all weights λ_C and λ_H and pairs of graphs G^1 and G^2. Additionally, we specify the cases in which the obtained correspondence is a weighted mean correspondence and not an approximated weighted mean correspondence.

3.1 Reasoning about Optimality and Accuracy

If we want to use equation 14 to solve our problem instead of equation 12, we must now demonstrate that function $\{\lambda_C \cdot Cost(G^1, G^2, f'') + \lambda_H \cdot [d_H(f''^a, f'') + d_H(f'', f''^b) - d_H(f''^a, f''^b)]\}$ extracted from equation 12 minimises the same partial correspondence than $\left\{\left[\lambda_C \cdot \boldsymbol{C}''_{f''} + \lambda_H \cdot [1 - F''^{a,b}]\right]_{f''}\right\}$ extracted from equation 14. Notice that by definition, $Cost(G^1, G^2, f'') = \boldsymbol{C}''_{f''}$ therefore, we have to demonstrate the following equation,

$$[1 - F''^{a,b}]_{f''} = d_H\left(f''^a, f''\right) + d_H(f'', f''^b) - d_H\left(f''^a, f''^b\right) \tag{16}$$

If equation 16 holds, then we can confirm that it is valid to use equation 14 to solve our problem. Suppose the cardinality of Σ''^1 and Σ''^2 is n. Since we have defined these sets such that they do not have any element in common, then, by definition of these sets, $d_H(f''^a, f''^b) = n$. Given the involved correspondences f'', f''^a and f''^b, we can define the three following natural numbers: n_p, n_q and n_t.

1) n_p: number of nodes in Σ''^1 that hold $f''(m_i^1) \neq f''^a(m_i^1)$ and $f''(m_i^1) \neq f''^b(m_i^1)$.
2) n_q: number of nodes in Σ''^1 that hold $f''(m_i^1) = f''^a(m_i^1)$ and $f''(m_i^1) \neq f''^b(m_i^1)$.
3) n_t: number of nodes in Σ''^1 that hold $f''(m_i^1) \neq f''^a(m_i^1)$ and $f''(m_i^1) = f''^b(m_i^1)$.

Likewise, by definition of these sets there is no m_i^1 such that $f''(m_i^1) = f''^a(m_i^1)$ and $f''(m_i^1) = f''^b(m_i^1)$. Therefore, $n = n_p + n_q + n_t$. For simplicity, we order the nodes in Σ''^1 such that m_1^1 to $m_{n_p}^1$ hold the first condition, $m_{n_p+1}^1$ to $m_{n_p+n_q}^1$ hold the second condition and $m_{n_p+n_q+1}^1$ to m_n^1 hold the third condition.
For equation 16 to hold, we first demonstrate that $[1 - F''^{a,b}]_{f''} = n_p$ and afterwards that $d_H\left(f''^a, f''\right) + d_H(f'', f''^b) - d_H\left(f''^a, f''^b\right) = n_p$.

1) Demonstration of $[1 - F''^{a,b}]_{f''} = n_p$:
Suppose that $f''(m_i^1) = m_k^2$ then

$$[1 - F''^{a,b}]_{f''} = \sum_{i=1}^{n} (1 - F''^{a,b})[i,k] = \sum_{i=1}^{n_p} 1 + \sum_{i=n_p+1}^{n} 0 = n_p$$

2) Demonstration of $d_H(f''^a, f'') + d_H(f'', f''^b) - d_H(f''^a, f''^b) = n_p$:

$$\begin{aligned} d_H(f''^a, f'') + d_H(f'', f''^b) - n \\ &= \sum_{i=1}^{n} \left(2 - \delta\left(f''^a(m_i^1), f''(m_i^1)\right) - \delta\left(f''(m_i^1), f''^b(m_i^1)\right)\right) - n \\ &= \left(\sum_{i=1}^{n_p} 2 + \sum_{i=n_p+1}^{n_p+n_q} 1 + \sum_{i=n_p+n_q+1}^{n} 1\right) - n = 2n_p + n_q + n_t - n = n_p \end{aligned}$$

■

In some cases, it is interesting to know if we have obtained a weighted mean correspondence or an approximated one. First, we have to realise that if both f' and f'' are partial weighted mean correspondences, then the union $f = f' \cup f''$ is a weighted mean correspondence, since by definition $f' \cap f'' = 0$. It is clear that if equation 8 holds for both partial correspondences, then it holds for the complete one. Moreover, by definition of our partial correspondence f', it is always defined as weighted mean correspondence. Therefore, we conclude that the obtained correspondence $\bar{f}^*$ is a weighted mean correspondence when $\bar{f}''^*$ is also a weighted mean correspondence. These cases are the ones where $\bar{f}''^*$ holds equation 8. Since we have demonstrated that $d_H(f''^a, f'') + d_H(f'', f''^b) - d_H(f''^a, f''^b) = n_p$, then n_p must be 0. By definition of n_p, these correspondence are the ones where $\bar{f}''^*(m_i^1) = f''^a(m_i^1)$ or $\bar{f}''^*(m_i^1) = f''^b(m_i^1)$. Therefore, we conclude that

$$\begin{gathered} \bar{f}^*_{\lambda_C,\lambda_H} \text{ is a weighted mean correspondence if:} \\ \bar{f}''^*_{\lambda_C,\lambda_H}(m_i^1) = \begin{cases} f''^a(m_i^1) & or \\ f''^b(m_i^1) & \forall m_i^1 \in \Sigma''^1 \end{cases} \end{gathered} \tag{17}$$

The cost of testing if the correspondence obtained is a weighted mean is linear with respect to the number of discrepancies between correspondences f^a and f^b.
Note that if $\lambda_C = 0$ and $\lambda_H > 0$ and if we use the Hungarian method [1] to solve equation 14, our method always obtains a weighted mean correspondence. Indeed, in this case, solving equation 14 reduces to minimise $[1 - F''^{a,b}]_{f''}$ (equation 16), which is equal to zero if and only if the associated correspondence is a weighted one (equation 17, and above discussion). The Hungarian algorithm, being optimal, reaches this minimal value and the associated correspondence is a weighted mean.

4. Experimentation

We have split the experimental section into four sub-sections. First, we introduce the databases used for our experimentation. Then, we describe the first experiment, which aims confirm the theoretical background and validating parameters λ_C and λ_H. This way, we want to check how close the obtained consensus correspondence is to the mean correspondence ("correspondence" means correspondence). Also, we want to know how low the obtained cost is.
The aim of the following two experiments described on the third and fourth subsections respectively, is to show our method in two practical scenarios (the ones commented in the introduction). In the first one, there are no outliers, that is, the two proposed correspondences relate the whole points of both sets and so, the correspondences are bijective functions and both sets have the same cardinality. We call it a correspondence-oriented scenario. In the second one, the sets of points that relate both correspondences

are not exactly the same and so, some points are considered inliers in one of the correspondences but outliers in the other one. We call it a feature-oriented scenario.

4.1 Point Set Databases with an oracle's correspondence

In this section we introduce the two datasets used for the experimentation process. Given two images sets, our purpose is to generate datasets containing correspondences to consider as oracle. An oracle is a correspondence between elements of two sets that we consider as ideal.

"Tarragona Palmprint" database

This database is composed of 640 triplets of two minutiae sets G_i^1 and G_i^2, and a correspondence f_i between them. The database was created as follows. First, we used images contained in the Tsinghua 500 PPI Palmprint Database [30]. This is a public high-resolution palmprint database composed of 500 palmprint images of a resolution of 2040 x 2040 pixels. Eight different palmprint inks are enrolled from each person. We used only the first 10 subjects of the database, therefore making our initial dataset composed of 80 palmprints from which we extracted 80 minutiae sets using the algorithm presented in [31], [32] and [33]. The average number of extracted minutiae is 1000 per palmprint. Given each subject, we computed the minutiae correspondences between its 8 sets of minutiae through the Hungarian method [1], plus a greedy algorithm to select the matches from the resulting matrix of the previous step. Thus, we generate a total of 64 oracles per subject and so, a total of 640 triplets composed of two sets of minutiae from the same subject and the corresponding correspondence, $\{G_i^1, G_i^2, f_i\}$, where $i \in [1..640]$. The information of each minutia is the 2D position, angle and type of minutia (Terminal or Bifurcation)

"Tarragona Exteriors" database

This database is composed of 5 sequences $\times 5$ feature extractors $\times 10$ pairs of images $\times 2$ matching algorithms = 500 quartets of two point sets and two correspondences between them. The first correspondence has been obtained by one of two experimented matching algorithms, and the second one is the correct correspondence. To compose this database, we used the 11 image samples from 5 public image databases called "BOAT", "EAST_PARK", "EAST_SOUTH", "RESIDENCE" and "ENSIMAG" [35], [36]. These databases are composed of a sequence of images taken of the same object, but with different points of views and scales. Together with the images, the homography estimations that convert the first image (img00) of the set into the other ones (img01 through img10) is provided. From each of the images, we extracted the 50 most reliable salient points using 5 methodologies: FAST, HARRIS, MINEIGEN, SURF (native Matlab 2013b libraries) and SIFT (own library). Moreover, we computed the 10 correspondences between the first image of the sequence and the other ten ones using 2 algorithms: the Matlab function *MatchFeatures* and the *FastBipartite* method [13] [14] (note that *FastBipartite* is usually applied to match graphs [31]). Finally, we defined the oracle using the homography provided by the original image databases. Thus, the database has a total of 500 quartets composed of two sets of features and two correspondences $\{G_i^1, G_i^2, f_i, h_i\}$, where $i \in [1..500]$. The features in the feature set are the ones extracted using one of the 5 methodologies.

4.2 Evaluation in Ideal Conditions and Parameter Validation

The aim of our method is to deduct a consensus correspondence that has the minimum possible cost and that is a weighted mean of both correspondences proposed by human experts or automatic systems. We have performed tests in the first database for this purpose. We evaluated three different aspects of our method. First, we analysed how minimum the cost of the obtained consensus is. To that aim, we used the cost function (equation 1), where c considers the distance between the angle and position of the minutiae. Second, we analysed if our method is minimising the Regularization term defined in equation 12. We propose the following metric for this purpose,

$$Regularisation\left(f^{a}, f^{b}, \bar{f}^{*}_{\lambda_C,\lambda_H}\right) = \frac{d_H\left(f^{a}, \bar{f}^{*}_{\lambda_C,\lambda_H}\right) + d_H\left(f^{b}, \bar{f}^{*}_{\lambda_C,\lambda_H}\right)}{d_H\left(f^{a}, f^{b}\right)} \quad (18)$$

As the value of $Regularisation$ approaches to 1, the closer a consensus is to be a weighted mean. Third, we checked if the obtained consensus $\bar{f}^{*}_{\lambda_C,\lambda_H}$ is indeed a weighted mean correspondence. In this case, we simply validated that equation 8 holds.

Given each correspondence f_i in the 640 triplets that compose the "Tarragona Palmprint" database [37], we randomly generated 212 pairs of correspondences f^{a} and f^{b}. Therefore, we tested a total of $640\text{x}212 = 135680$ pairs of correspondences: $\{f_i^{\alpha a}, f_i^{\alpha b}\}$, where i ∈ $[1..640]$ and $\alpha \in [1..212]$. Index i represents the original triplet, and index α represents the level of noise added to the original correspondence in the i[th] triplet. The noise is set such that $d_H\left(f_i^{\alpha a}, f_i^{\alpha b}\right) = \alpha$. Thus, $f_i^{\alpha a}$ is first randomly generated. Then we randomly generate $f_i^{\alpha b}$ such that $d_H\left(f_i^{\alpha b}, f_i\right) = (\alpha - \beta)$, where $0 \leq \beta \leq \alpha$ and $\beta = d_H\left(f_i^{\alpha a}, f_i\right)$. We have applied this method to avoid $d_H\left(f_i^{\alpha a}, f_i\right) = d_H\left(f_i, f_i^{\alpha b}\right)$ for every case.

Figures 5 to 8 show the obtained results in this section. We chose three different configurations for λ_C and λ_H: a) The first one: $\lambda_C = 1$ and $\lambda_H = 0$. This configuration reproduces a classical minimum-cost method (red lines) since the specialist correspondences are not considered. b) The second one: $\lambda_C = 1$ and $\lambda_H = 1$. There is a contribution on the consensus cost and the specialist correspondences, keeping in mind that the features' cost has been normalised to 1(green lines). c) The third one: $\lambda_C = 0$ and $\lambda_H = 1$. The cost of the consensus correspondence is not minimised. Therefore, this approach would be considered a pure consensus of the correspondences done by $f_i^{\alpha a}$ and $f_i^{\alpha b}$ (violet lines). The horizontal axis in figures 5 to 7 represents the noise α and the vertical axis represents the average of the quality measures computed for each of the 640 triplets.

Figure 5 shows the average cost of $\bar{f}^{*}_{\lambda_C,\lambda_{H_i}}$ with respect to α. It is clearly noticeable that, the larger the distance between the specialists ($d_H\left(f_i^{\alpha a}, f_i^{\alpha b}\right) = \alpha$), the higher the cost of the resulting consensus correspondence. This relation appears because when the specialists' correspondences were generated from the original oracle in the "Tarragona Palmprint" database, we introduced some random modifications that made the correspondence "worse" than the original one, that is, their cost tends to increase. We also notice that the correspondence's cost is lower when the optimisation method considers the cost variable ($\lambda_C = 1$) and so, we deduct that this term must be always considered in the optimisation function. Moreover, the cost of configuration $\lambda_C = 1$ and $\lambda_H = 1$ (green line) is slightly higher than configuration $\lambda_C = 1$ and $\lambda_H = 0$ (red line). This happens because forcing the consensus to be a weighted mean increases slightly the correspondence's cost.

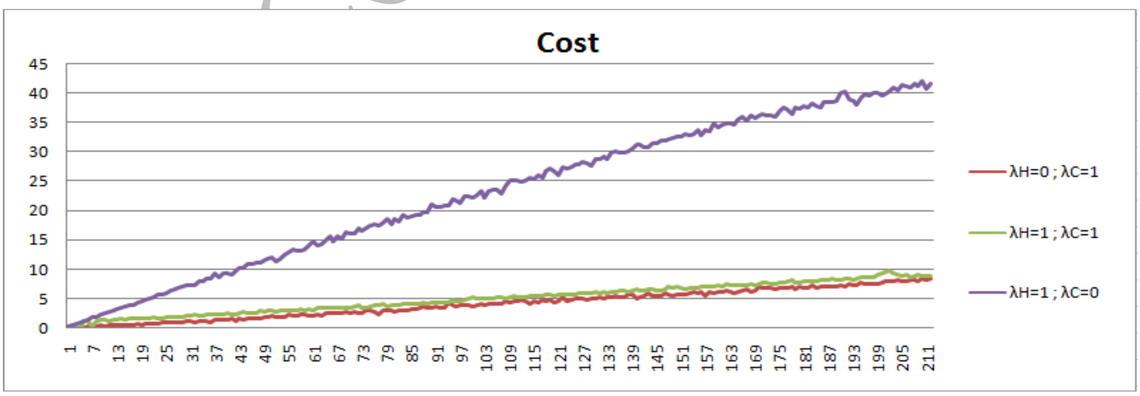


Figure 5. Cost of the consensus correspondence w.r.t. the Hamming distance between the original correspondences

Figure 6 shows the behaviour of the Regularisation term (equation 18) as α is increased. We can appreciate that, since the $\lambda_H = 1$ and $\lambda_C = 0$ configuration (violet line) always selected a weighted mean consensus, the value of $Regularisation$ remains 1. Nevertheless, the $\lambda_H = 0$ and $\lambda_C = 1$ configuration (red line), which resembles a classical minimisation solution, is less likely to obtain a weighted mean correspondence than the $\lambda_H = 1$ and $\lambda_C = 1$ (green line) which represents our proposal.

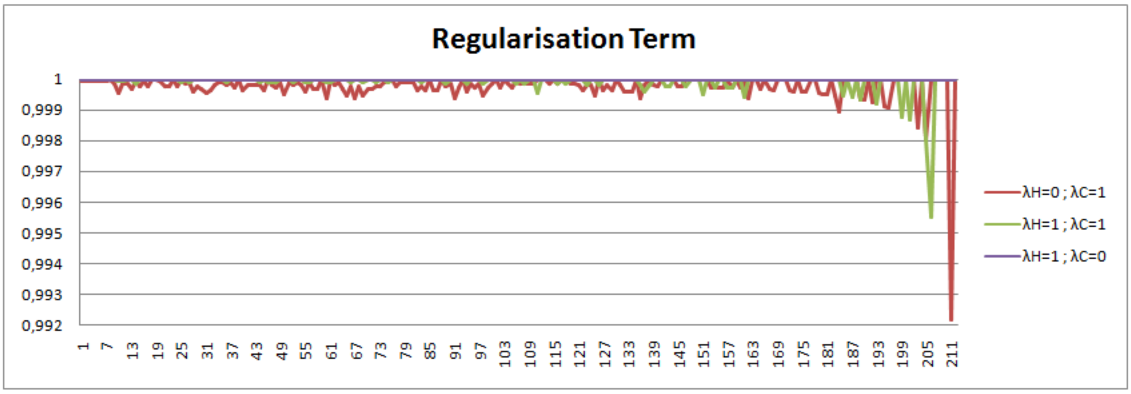


Figure 6. Regularisation term minimisation for the correspondence w.r.t. the hamming distance between the original correspondences

This aspect is further supported by Figure 7, where we show the percentage of experiments in which $\bar{f}^*_{\lambda_C,\lambda_{H_i}}$ is really a weighted mean of $f_i^{\alpha a}$ and $f_i^{\alpha b}$. First of all, as we deducted from equation 17, the configuration $\lambda_C = 0$ and $\lambda_H = 1$ (violet line) always results in weighted mean consensus. For configuration $\lambda_H = 0$ and $\lambda_C = 1$ (red line), we realise that not every consensus correspondence is a weighted mean since $f_i^{\alpha a}$ and $f_i^{\alpha b}$ are not considered. Finally, only in some cases that α is sufficiently large, configuration $\lambda_H = 1$ and $\lambda_C = 1$ (green line) fails to obtain a weighted mean.

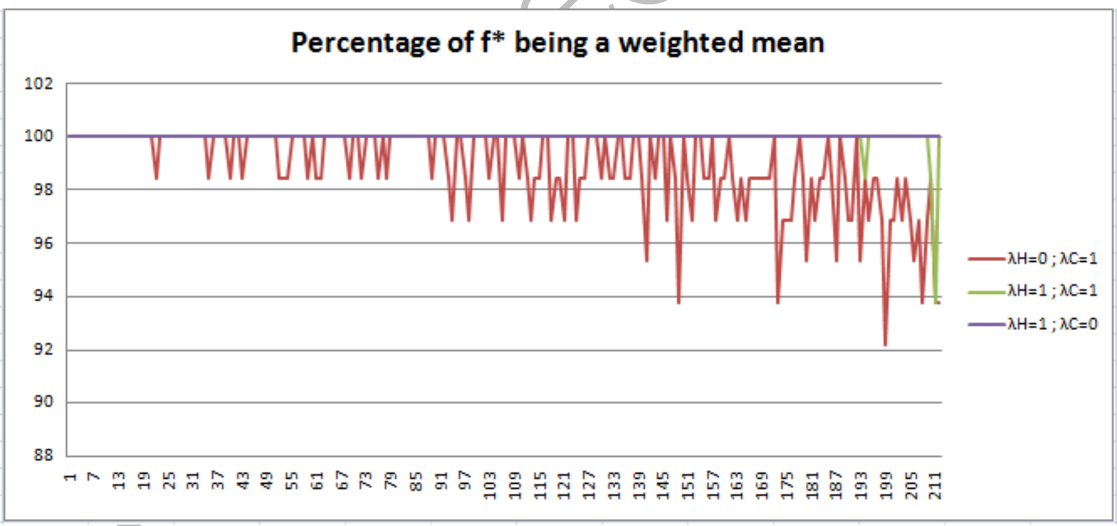


Figure 7. Percentage of tests that the consensus correspondence becomes a weighted mean w.r.t. the hamming distance between the original correspondences

Figure 8 shows the cost of the obtained consensus $\bar{f}^*_{\lambda_C,\lambda_H}$ with respect to the mean cost of the original correspondences f^a and f^b in all configurations of α. When the cost is considered using $\lambda_C = 1$ (figures 8a and b), the consensus always obtains a lower cost than the average of the original correspondences but this is not the case when $\lambda_C = 0$ (figure 8c). When comparing figures 8a and b, we realise results are almost similar. From these first experiments, we conclude that when both parameters are taken into consideration, the method obtains consensus correspondences that are closer to the mean and whose costs are close to the minima.

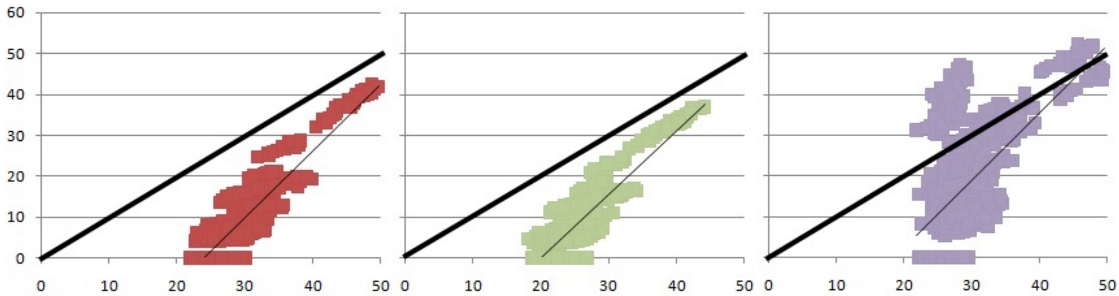


Figure 8. Cost of the obtained correspondence w.r.t. the mean cost of the original correspondences. (a):$\lambda_H = 0$ and $\lambda_C = 1$. (b): $\lambda_H = 1$ and $\lambda_C = 1$. (c): $\lambda_H = 1$ and $\lambda_C = 0$.

4.3 Practical Application in Correspondence-oriented Scenario

From the “Tarragona Exteriors” database [36] (figure 1 shows some examples), we selected the cases such that correspondences were obtained by the *MatchFeatures* function (250 quartets). We applied our consensus methodology using the configurations of parameters $\lambda_C = 1$ and $\lambda_H = 1$, and we checked the correct element mappings through the validation correspondence in the database. The cost function between elements was the Euclidean distance between the two sets of features. Since the features in this database are normalised, we assume it is fair to compare data from different extractors. We compared our method with a simpler one that we called “No Discrepancies”. This method selects the mappings where no discrepancies are found. That is, the union of mappings that appear in both methods and the correspondences that appear in one of the methods, but not in the other. The “No Discrepancies” method can be seen as a simplification of voting methods [28], when only two voting entities are involved. Note that this is a correspondence-oriented application, therefore the salient points that are not mapped by the method are not used. That is, we select only the salient points that have been mapped and, for this reason, the number of point mappings is not always 50.

Figure 9 shows a graphical representation of the correct correspondences of an item in the “Tarragona Exteriors” database obtained by the feature extraction methods FAST and HARRIS using the *MatchFeatures* function. Although the matching algorithm is the same, the extracted points are different, and so, the obtained correspondences are different. In red we show the salient points and correspondences made exclusively by FAST, and in yellow we show the salient points extracted and the correspondences made by HARRIS. Finally, in green, we depict the extracted points that appear in both methods (two points are considered the same if they are located within a radius of 6 pixels or less) and also the common correspondences. We realise that FAST has achieved 6 correct correspondences (3 red lines plus 3 green ones) and HARRIS has achieved 7 correct correspondences (4 yellow lines plus 3 green ones). In this case, the “No Discrepancies” method obtains 10 correct correspondences (the addition of red, green and yellow) since there are not any discrepancies between red and yellow lines.

Assuming we are in a correspondence-oriented scenario, the non-mapped points by any correspondence are not considered anymore. Therefore, correspondences f''^a_i and f''^b_i are composed of red and yellow lines and correspondences f'^a_i and f'^b_i are represented by the green lines. Note that in each experiment, the cardinality of the sets does not depend on the original sets, but rather on the matching processes. Moreover, given the correspondences f''^a_i and f''^b_i, there are two types of points in both sets. The ones that contribute in both correspondences and the ones that only appear in one of them.

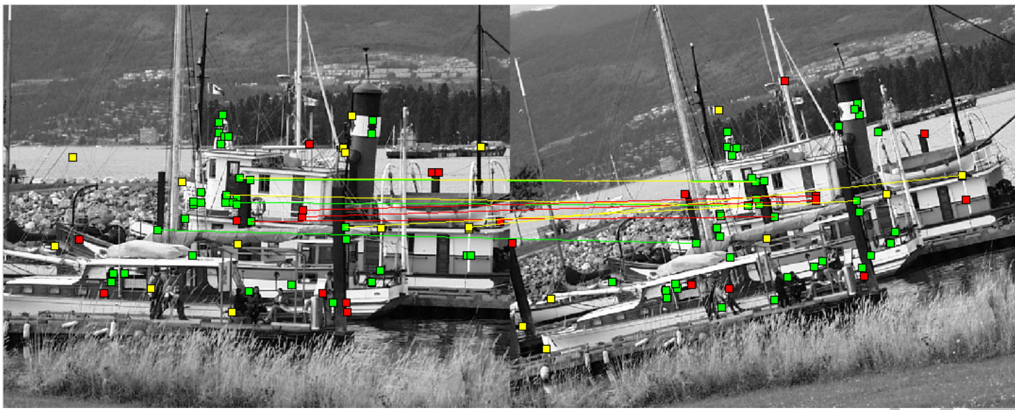


Figure 9. Red (yellow) points extracted by FAST (HARRIS). Green points extracted by both methods. Lines are mapped points deducted by *MatchFeatures* Matlab function.

Figure 10 shows the consensus correspondence obtained by our method with parameters $\lambda_C = 1$ and $\lambda_H = 1$. Only the correct mappings with respect to the oracle in the database are depicted. For this pair of images, we also achieve 10 correct point mappings.

Figure 10. Consensus correspondence obtained by our method given the initial correspondences in figure 9.

Table 2 shows the average number of correct correspondences obtained by our consensus method compared to the "No Discrepancies" method, given the 5 sequences. $\boldsymbol{C_{a,b}}$ represents the combination of extractor methods M_a and M_b, where M_1: FAST, M_2: HARRIS, M_3: MINEIGEN, M_4: SURF and M_5: SIFT. We realise that the consensus method obtains a greater number of correct correspondences with respect to the "No Discrepancies" method for all combinations and datasets.

Table 2. Average number of correct correspondences obtained by our consensus method and the "No Discrepancies" method.

	Correct correspondences	$C_{1,2}$	$C_{1,3}$	$C_{1,4}$	$C_{1,5}$	$C_{2,3}$	$C_{2,4}$	$C_{2,5}$	$C_{3,4}$	$C_{3,5}$	$C_{4,5}$	Average
Boat	**Consensus**	**14**	**16**	**70**	**53**	**14**	**69**	**54**	**73**	**59**	**95**	**51.7**
	No Discrepancies	12	14	65	46	13	64	45	67	49	79	45.4
East Park	**Consensus**	**4**	**4**	**150**	**79**	**2**	**147**	**76**	**144**	**91**	**270**	**96.7**
	No Discrepancies	3	3	67	74	1	68	61	70	63	213	62.3
East South	**Consensus**	**1**	**1**	**48**	**6**	**1**	**48**	**5**	**48**	**5**	**50**	**21.3**
	No Discrepancies	1	1	43	3	1	43	2	43	2	43	18.2

Resid	**Consensus**	**2**	**3**	**106**	**4**	**3**	**108**	**5**	**104**	**4**	**110**	**44.9**
	No Discrepancies	1	1	68	4	2	63	6	65	4	73	28.7
Ensimag	**Consensus**	**1**	**2**	**112**	**54**	**1**	**111**	**53**	**111**	**75**	**153**	**67.3**
	No Discrepancies	1	2	75	34	1	73	39	73	67	113	47.8
Average	**Consensus**	**4.4**	**5.2**	**97.2**	**39.2**	**4.2**	**96.6**	**38.6**	**96**	**46.8**	**135.6**	
	No Discrepancies	3.6	4.2	63.6	32.2	3.6	62.2	30.6	63.6	37	104.2	

4.4 Practical Application in Feature-oriented Scenario

For this experiment, we also used the "Tarragona Exterior" database. We selected the two most frequently used feature extractors in the consulted literature: SIFT and SURF. Moreover, we used both matching algorithms *MatchFeatures* and *FastBipartite*. We have considered the whole 50 points in the point sets, independently if they were outliers or not. Recall that the outlier points are the ones that are not mapped by the correspondence function. Thus, when we computed the consensus correspondence given two oracles, it could happen that a point mapped by the consensus correspondence would not have been mapped by any of the two oracles. Table 3 shows the average number of discrepancies and the percentage that these discrepancies represent (the number of points in the set is 50) between both algorithms. A discrepancy appears when the point mapping of both matching algorithms is different. We show this table since our algorithm can only increase the accuracy when the number of discrepancies is considerable. If there are no discrepancies, the algorithm imposes the mapping decided by both oracles.

Table 3. Average number of discrepancies between *MatchFeatures* and *FastBipartite*

	SIFT	SURF	SIFT (%)	SURF (%)
Boat	17.1	39.7	34.2	79.4
East Park	16	44.8	32	89.6
East South	17	40.8	34	81.6
Resid	19.7	44.3	39.4	88.6
Ensimag	18.5	43.3	37	86.6
Average:	**17.6**	**42.5**	**35.3**	**85.1**

Table 4 shows the average accuracy of the mappings calculated by our consensus method. This value is computed as the average accuracy of the 10 correspondences and the 5 image sequences. The overall accuracy (column) of a correspondence is the number of correct node mappings (validated using the oracle included in the dataset) divided by the 50 possible node mappings. When the correspondences are considered in the optimisation function, $\lambda_H = 1$, the consensus method obtains slightly better accuracy than both of the initial matching methods. These accuracy is increased when the cost is also considered, $\lambda_C = 1$. As commented above, our algorithm can only increase the accuracy in the discrepancy cases. For this reason, we added the accuracy on the discrepancies (*Discrepancies Accuracy %* column). This value represents the percentage of mappings that are properly correctly found given that a discrepancy between the original methods existed. We realise that the consensus method obtains an important increase of the accuracy with respect to the initial methods. We observe that our consensus model obtains an accuracy of 85% (SURF) and 47% (SIFT) for the mappings with discrepancies between the original methods.

Table 4. Accuracy comparison between methods *MatchFeatures* and *FastBipartite* and features SURF and SIFT.

	Method	Parameters	Accuracy (%)	Discrepancies Accuracy (%)
	MatchFeatures		29.56	
	FastBipartite		24.32	

SURF	Consensus	$\lambda_C = 1 \& \lambda_H = 1$	30.28	85.73
		$\lambda_C = 0 \& \lambda_H = 1$	29.84	84.48
		$\lambda_C = 1 \& \lambda_H = 0$	28.04	79.38
SIFT	*MatchFeatures*		39.56	
	FastBipartite		25.6	
	Consensus	$\lambda_C = 1 \& \lambda_H = 1$	40.12	47.11
		$\lambda_C = 0 \& \lambda_H = 1$	39.72	46.64
		$\lambda_C = 1 \& \lambda_H = 0$	32.68	38.37

5. Conclusions and Further Work

We have presented a consensus method that obtains a correspondence between two sets of elements given two initial correspondences between these sets of elements generated by separate entities. The method is based on a classical optimisation scheme. The Loss function is defined as the cost of the correspondence and the Regularisation term gauges how near the correspondence is to the mean of the correspondences. We have shown in the experimental section the validity of our method. 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. 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 case. 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 correspondences. Moreover, we believe that in the several correspondences case, the number of discrepancies would increase. Therefore, our first intuition is that our methodology would obtain a good consensus correspondence in this scenario.

6. References

1. Kuhn, H. W., "The Hungarian method for the assignment problem Export", Naval Research Logistics Quarterly 2(1-2), 83–97, 1955.
2. Fischler, M.A., Bolles, R.C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", Commun. ACM 24 (6), pp: 381–395, 1981
3. Riesen, K., Bunke, H., "Approximate graph edit distance computation by means of bipartite graph matching", Image Vision Comput. 27(7): 950-959, (2009)
4. Sanromà, G., Alquézar, R., Serratosa, F., Herrera, B., "Smooth Point-set Registration using Neighbouring Constraints", Pattern Recognition Letters 33, pp: 2029-2037, 2012.
5. 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), pp: 292-304, 2012.
6. Cortés, X., Serratosa, F., "An Interactive Method for the Image Alignment problem based on Partially Supervised Correspondence", Expert Systems With Applications 42 (1), pp: 179 - 192, 2015.
7. Tomasi, C., Kanade, T., "Detection and tracking of point features", Tech. Rep. CMUCS-91-132, Carnegie Mellon University, 1991.
8. Han, W., Brady, M., "Real-time corner detection algorithm for motion estimation", Image and Vision Computing, pp: 695–703, 2005.
9. Rosten, E., Drummond, T., "Machine learning for high-speed corner detection", European Conference on Computer Vision, vol. 1, pp. 430–443, 2006.
10. Mikolajczyk, K., Schmid, C., "A performance evaluation of local descriptors", IEEE Trans. Pattern Anal. Mach. Intell. 27 (10), pp: 1615–1630, 2005.

11. Lowe, D.G., “Object recognition from local scale-invariant features”, Proceedings of the International Conference on Computer Vision 2. pp. 1150–1157, 1999.
12. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V., “SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008.
13. Serratosa, F., “Fast Computation of Bipartite Graph Matching”, Pattern Recognition Letters, 45, pp: 244 - 250, 2014.
14. F. Serratosa, Speeding up Fast Bipartite Graph Matching trough a new cost matrix, International Journal of Pattern Recognition and Artificial Intelligence, 29 (2), 2015.
15. F. Serratosa & X. Cortés, Interactive Graph-Matching using Active Query Strategies, Pattern Recognition 48 (4), 2015.
16. Franek, F. Jiang, X., He, C., “Weighted Mean of a Pair of Clusterings”, Pattern Analysis Applications. Springer-Verlag. 2012.
17. Mimaroglu, S., Erdil, E., “Combining multiple clusterings using similarity graphs”, Pattern Recognition 44 pp. 694-703. Elsevier. 2010.
18. Murino, L., Angelini, C., De Feis, I., Raiconi, G., Tagliaferri, R., “Beyond classical consensus clustering: The least squares approach to multiple solutions”, Pattern Recognition Letters 32 pp. 1604-1612. Elsevier. 2011.
19. Monti, S., Tamayo, P., Mesirov, J.P., Golub, T.R., “Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data”, Machine Learning 52(1-2): 91-118 (2003)
20. Ghahramani, Z., “Unsupervised Learning. Advanced Lectures on Machine Learning”, pp. 72-112. Springer-Verlag. 2004.
21. Papadimitriou, C., Steiglitz, K., “Combinatorial Optimization: Algorithms and Complexity”, Dover Publications, July 1998.
22. A. Solé & F. Serratosa, Graduated Assignment Algorithm for Multiple Graph Matching based on a Common Labelling, International Journal of Pattern Recognition and Artificial Intelligence 27 (1), pp: 1 - 27 2013.
23. A. Solé & F. Serratosa, Models and Algorithms for computing the Common Labelling of a set of Attributed Graphs, Computer Vision and Image Understanding 115 (7), pp: 929-945, 2011.
24. M. Ferrer, E. Valveny & F. Serratosa, Median graphs: A genetic approach based on new theoretical properties, Pattern Recognition 42 (9), pp: 2003-2012, 2009.
25. M. Ferrer, E. Valveny & F. Serratosa: Median graph: A new exact algorithm using a distance based on the maximum common subgraph, Pattern Recognition Letters 30 (5), pp: 579-588, 2009.
26. Timoftea, R., Gool, L.V., “Adaptive and Weighted Collaborative Representations for Image Classification”, Pattern Recognition Letters, available online, 2014.
27. Saha, S., Ekbal, A., “Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition”, Data & Knowledge Engineering, 85, 2013, pp: 15-39,
28. Moreno-García, C.F., Serratosa, F., “Weighted Mean Assignment of a Pair of Correspondences using Optimisation functions”, Syntactic and Structural Pattern Recognition, S+SSPR2014, Finland, 2014.
29. Solé, A., Serratosa, F., Sanfeliu, A., “On the Graph Edit Distance cost: Properties and Applications”, International Journal of Pattern Recognition and Artificial Intelligence 26, (5). 2012.
30. Jain, A.K., Feng, J., “Latent Palmprint Matching”. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2009.
31. J. Funada, *et. al.*, “Feature Extraction Method for Palmprint Considering Elimination of Creases”, Proc. 14th Int. Conf. Pattern Recognition, pp. 1849-1854, 1998.
32. Dai, J., Zhou, J., “Multifeature-Based High-Resolution Palmprint Recognition”, IEEE Trans. Pattern Analysis and Machine Intelligence, 33 (5), pp: 945-957, 2011.
33. Dai, J., Feng, J., Zhou, J., “Robust and Efficient Ridge Based Palmprint Matching”, IEEE Transactions on Pattern Analysis and Matching Intelligence. 34 (8), 2012.
34. http://www.featurespace.org
35. 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, pp: 2493-2502 , 2013.

36. X. Cortés & F. Serratosa, Learning Graph-Matching Edit-Costs based on the Optimality of the Oracle's Node Correspondences, Pattern Recognition Letters, 56, pp: 22 - 29, 2015.
37. http://deim.urv.cat/~francesc.serratosa/databases/