

Active Graph Matching Based on Pairwise Probabilities between Nodes^{*}

Xavier Cortés, Francesc Serratosa, and Albert Solé-Ribalta

Universitat Rovira i Virgili, Departament d'Enginyeria Informàtica i Matemàtiques, Spain
{xavier.cortes, francesc.serratosa, albert.sole}@urv.cat

Abstract. We propose a method to perform active graph matching in which the active learner queries one of the nodes of the first graph and the oracle feedback is the corresponding node of the other graph. The method uses any graph matching algorithm that iteratively updates a probability matrix between nodes (Graduated Assignment, Expectation Maximisation or Probabilistic Relaxation). The oracle's feedback is used to update the costs between nodes and arcs of both graphs. We present and validate four different active strategies based on the probability matrix between nodes. It is not needed to modify the code of the graph-matching algorithms, since our method simply needs to read the probability matrix and to update the costs between nodes and arcs. Practical validation shows that with few oracle's feedbacks, the algorithm finds the labelling that the user considers optimal because imposing few labellings the other ones are corrected automatically.

Keywords: Machine Learning, Active Graph Matching, Interactive Graph Matching, Least Confident, Maximum Entropy, Expected Model Change.

1 Introduction

Generally speaking, *machine learning* is a discipline concerned with the design and development of algorithms that allow computers to evolve behaviours based on examples [1]. In this discipline, a learner can take advantage of examples to capture characteristics of interest from the data respect of their class and to be able to deduct the class that new examples could belong to. *Error-tolerant graph matching* [2] is another discipline that aims to find the best labelling between nodes of both graphs such that the cost of this optimal labelling is the minimum among all possible labellings. If we put together machine learning and error-tolerant graph matching disciplines, we can define a model in which examples are composed by the set of nodes of one of the graphs and classes are the nodes of the other graphs. Therefore, what we want to learn is which is the matching between two graphs that is considered to be the best.

The key idea behind *active learning* [3, 7, 20] is that a machine learning algorithm can achieve a greater accuracy with fewer classified training examples if it is allowed to

^{*} This research is supported by Consolider Ingenio 2010: project CSD2007-00018 & by the CICYT project DPI 2010-17112.

choose the data from which it learns. Active learning is well motivated in many modern machine-learning problems, where unclassified examples may be abundant but finding the class is difficult, time-consuming or expensive to obtain [1, 21]. Active learning has been applied in several fields such as speech recognition [6, 21], information extraction [10, 13, 15, 17], robotics [12], transcription of text images [22, 28] or object classification in general [4, 5, 6, 9, 14, 18]. And in general, for parameter selection [8].

We present a model in which we have put together the active learning and graph matching concepts. In this case, the learner queries the node that it is supposed to produce a greater impact on the labelling between both graphs. In our case, the active learner may ask queries in the form of graph nodes and asks which are the nodes of the other graph that they have to be matched. The answerer of the query might be another automatic system or a human annotator (in general, it is called an *oracle*).

Active learning scenarios are usually classified in three classes. The active learner generates a query de novo [18], the active learner receives a stream of objects and decides whether query them or not [19] and the pool-based active learning, in which the active learner decides to query an element from a sub-set of unlabelled objects [4, 11], or with sub-sampling [5]. In our case, we always have nodes of both graphs, for this reason, our scenario can be classified as the third scenario: *pool-based active learning*.

All active learning scenarios are involved in evaluating the informativeness of unclassified examples, which can either be generated de novo or sampled from a given distribution. It has been proposed many ways of formulating such query strategies in the literature [3]. We have considered two main strategies: *Uncertainly Sampling* and *Expected Model Change*. We have modelled the first one in three different strategies and the second one in one strategy.

The rest of the paper is organised as follows. In the next section, we present the graph-matching problem. In section 3, we present four different active strategies applied to the problem of finding the best labelling. In section 4, we show the algorithm to compute the active graph matching. Finally, in section 5 we show the practical evaluation and we conclude the paper in section 6.

2 Graph Matching and Isomorphism between Graphs

Let g^1 and g^2 be two attributed graphs. We suppose that g^1 and g^2 have the same number of nodes n since they have been enlarged enough to incorporate null nodes. We define nodes in g^1 and g^2 as $v_i^1 \in \Sigma_v^1$ and $v_a^2 \in \Sigma_v^2$ and we define arcs as $e_{ij}^1 \in \Sigma_e^1$ and $e_{ab}^2 \in \Sigma_e^2$, $\forall i, j, a, b \in \{1, \dots, n\}$. Moreover, let f be a bijective labelling between nodes of both graphs. The cost of matching graphs g^1 and g^2 , given this isomorphism f , is represented by

$$C_f(g^1, g^2) = \sum_{v_i^1 \in \Sigma_v^1} c_v(v_i^1, v_a^2) + \sum_{e_{ij}^1 \in \Sigma_e^1} c_e(e_{ij}^1, e_{ab}^2) \quad (1)$$

where $f(v_i^1) = v_a^2$ and $f(v_j^1) = v_b^2$. That is, the cost is defined as the addition of the pairwise costs of matching nodes and arcs [10]. These local costs can be represented through two matrices $C_v \in \mathbb{R}^{+}$, $C_v[i, a] = c_v(v_i^1, v_a^2)$ and $C_e \in \mathbb{R}^{+}$, $C_e[i, j, a, b] = c_e(e_{ij}^1, e_{ab}^2)$ and their definition depends on the application. Usual examples are

the Euclidean distance, when attributes have the position of the node in the image or the distance between local features such as SIFTs or HOGs.

There are several error-tolerant graph-matching algorithms that, using a minimization criteria, such as eq. (1), return the best isomorphism f between two graphs. For instance: probabilistic relaxation [23], Graduated-Assignment [24] or Expectation-Maximisation [25]. In fact, the input of these algorithms can be matrices C_v and C_e instead of graphs g^1 and g^2 since matrices capture all the differences between graphs and the minimisation cost is defined through these matrices (eq. 1). Considering that the involved graphs have a degree of disturbance and also the exponential complexity of the problem, these algorithms do not return exactly the isomorphism f but a probability matrix related to it. We represent this matrix by P where each cell contains $P[v_i^1, v_a^2] = Prob(f(v_i^1) = v_a^2)$. Thus, given the probability matrix P it is necessary to derive the final labelling f by a discretization process. There are several techniques to perform this discretization, e.g. [30]. Figure 1 represents the probabilistic graph-matching paradigm.

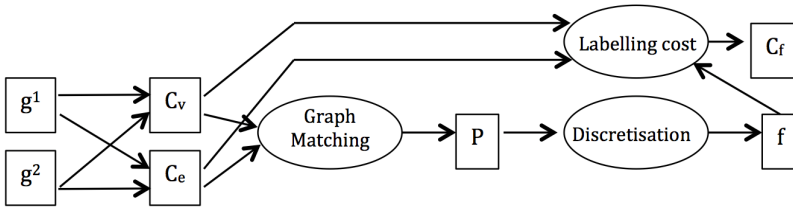


Fig. 1. Probabilistic graph matching framework

In the next section, we present four different strategies that, with the information of the probability matrix P , derive the node that has to be queried. Besides in section 4, we show how to use the interactive algorithm presented in [31] that modifies matrices C_v and C_e to consider the oracle feedback.

3 Active Learning Strategies

In this section, we present four strategies to select a node v^{1*} of g^1 that have to be queried to an oracle since the model assumes the exact knowledge of its mapping will increase the accuracy of the system. Therefore, given the selected node v^{1*} we ask for $f(v^{1*})$ and the oracle feedback is v^{2*} . In all strategies, the pool of nodes to be queried is composed by the nodes that have never been queried before. The logical function $Q(i)$ shows if node v_i^1 has been queried. When the active algorithm is initialised, $Q(i)$ takes the *False* value for all nodes of g^1 and this value is changed to *True* in each query. This logical function is used to assure a node is not queried several times. Note that in the case that $Q(i) = True$ for all nodes of g^1 then the following strategies return a null value. Nevertheless, in this case, the active algorithm (section 4) has to stop since the whole nodes have been queried. Besides, the computation of the following strategies is performed at function *Oracle_Feedback* in the algorithm.

The four strategies we present are classified on *Uncertainly Sampling* and *Expected Model Change* [3].

Uncertainly Sampling. The active learner queries the instances about which it is least certain how to classify. We define three different strategies.

Least Confident (LC): This strategy queries the element that its highest probability of belonging to a class is the lower one between all the elements. In our model, the learner queries the node v_{LC}^{1*} of g^1 that has not been previously queried and whose maximum probability given the nodes of g^2 is the lower. Node v_{LC}^{1*} is obtained in two steps. Firstly, we obtain the set of nodes in g^2 : $\{v^{2(1)}, \dots, v^{2(i)}, \dots, v^{2(n)}\}$ such that,

$$v^{2(i)} = \operatorname{argmax}_{\forall j=\{1,\dots,n\}} P[v_i^1, v_j^2]; \forall i = \{1, \dots, n\} \quad (2)$$

note that some of the nodes in this set can appear several times, $v^{2(i)} = v^{2(j)}$; $i \neq j$.

And secondly, we select the node in g^1 such that its respective node in the set obtains the minimum probability,

$$v_{LC}^{1*} = \operatorname{argmin}_{\forall i=\{1,\dots,n\}|Q(i)=False} P[v_i^1, v^{2(i)}] \quad (3)$$

Least Confident given the Current Labelling (LCCL): The aim of this strategy is to query the nodes that are matched through the current labelling but they have not been queried. Therefore, it could be seen as the method tries to minimise the hamming distance between the current labelling and the ideal labelling (the labelling that would have been predicted by the oracle if all the nodes were queried). The learner queries node v_{LCCL}^{1*} of g^1 that has not been previously queried and it has the minimum probability given the current labelling f . Formally,

$$v_{LCCL}^{1*} = \operatorname{argmin}_{\forall i=\{1,\dots,n\}|Q(i)=False} P[v_i^1, f(v_i^1)] \quad (4)$$

Maximum Entropy (ME): This strategy queries the element with maximum Shannon Entropy given the probabilities. The main idea of the method is to query the elements that they are more difficult to be classified. In our model, the selected node v_{ME}^{1*} is,

$$v_{ME}^{1*} = \operatorname{argmax}_{\forall i=\{1,\dots,n\}|Q(i)=False} - \sum_{j=1}^n P[v_i^1, v_j^2] \cdot \log(P[v_i^1, v_j^2]) \quad (5)$$

Expected Model Change (EMC): An active learner queries the instances that would impart the greatest change to the current model if we knew its class. A possible query strategy could be the “expected gradient length”. Since graph-matching probabilistic models are usually trained using gradient-based optimisation; the change imparted to the model can be measured by the length of the training gradient. In other words, the learner should query the instance that if changed its labelling, the gradient between the current labelling and the new one would have the largest magnitude. Considering this aim, we propose to query the node v_{EMC}^{1*} defined through the following equation,

$$v_{EMC}^{1*} = \operatorname{argmax}_{\forall i=\{1,\dots,n\} \wedge Q(i)=False} \{R_i\} \quad (6)$$

The value R_i shows the maximum magnitude of any possible change of the current labelling at node v_i^1 .

$$R_i = \max_{\forall j=\{1,\dots,n\}} \{P[v_i^1, v_j^2]\} - P[v_i^1, f(v_i^1)] \quad (7)$$

If $R_i > 0$, the current labelling of v_i^1 is not the ideal one, considering only probabilities $P[v_i^1, v_j^2]$, $\forall j = \{1, \dots, n\}$. On the contrary, if $R_i = 0$ then the current labelling is the one that obtains the maximum probability, so, it is the ideal case. Note that $R_i < 0$ is not possible.

4 Active Algorithm

Algorithm *Interactive Graph Matching* presented in [31] obtains a labelling between nodes of attributed graphs g^1 and g^2 considering the human feedback. That is, it computes several times a sub-optimal graph-matching algorithm (for instance [23, 24, 25]), but in each step, the cost matrices C_v and C_e are modified through the current user feedback. In fact, we assume the input of the algorithm is not both graphs but matrices C_v and C_e . The feedback of the user is introduced into the algorithm through a vector of simple actions w . One of the actions is $w_q = \text{Set}(v_i^1, v_a^2)$, in which the user imposes that the labelling has to be $v_a^2 = f(v_i^1)$. The matrix costs C_v and C_e are updated considering the human feedback through functions *Interactive_Node_Costs* and *Interactive_Edge_Costs*. See [31] for more details.

The active algorithm we present has a similar structure. We have only added function *Active_Query* and the expression $w_1 = \text{Set}(v^{1*}, v^{2*})$ with the aim of functions *Interactive_Node_Costs* and *Interactive_Edge_Costs* being compatible with the interactive algorithm in [31]. The algorithm stops when the oracle returns a special node (for instance a negative value) or all the nodes of g^1 have been queried. The final labelling cost obtained at the end of the algorithm is computed through the original costs, C_v^0 and C_e^0 .

Algorithm *Active Graph Matching*

Input: Attributed Graphs g^1 and g^2

Output: Labelling f and Cost C_f

$C_v^0, C_e^0 = \text{Initialise_Cost}(g^1, g^2)$; $C_v = C_v^0$; $C_e = C_e^0$.

$f = \text{Graph_Matching}(C_v, C_e)$.

Do

$v^{1*} = \text{Active_Query}(P, f)$.

$v^{2*} = \text{Oracle_Feedback}(g^1, g^2, v^{1*}, f)$.

$w_1 = \text{Set}(v^{1*}, v^{2*})$.

$C_v = \text{Interactive_Node_Costs}(w, C_v)$.

$C_e = \text{Interactive_Edge_Costs}(w, C_e)$.

$f = \text{Graph_Matching}(C_v, C_e)$.

Since Stop

Compute $C_f(C_v^0, C_e^0)$

End Algorithm

Figure 2 shows the probabilistic graph-matching framework with active learning. Dashed lines connect the active modules that do not appear in the classical framework shown in figure 1. Moreover, we have added the original costs C_v^0 and C_e^0 , since the labelling cost C_f is computed through these costs.

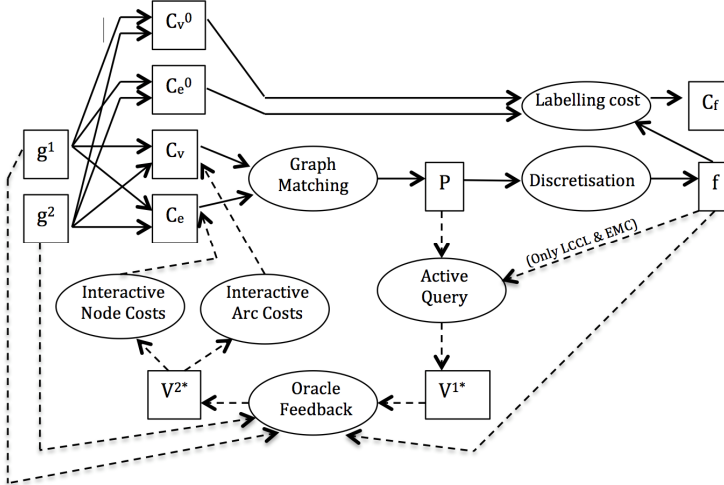


Fig. 2. Probabilistic graph matching framework with active learning

5 Practical Evaluation

We have used the CMU “house” and “castle” sequences. There are two datasets consisting of 111 frames of a toy house and a castle. Each frame in these sequences has been hand-labelled, with the same 30 landmarks identified in each frame [29]. From each landmark, we have only considered their bidirectional position in the image. From each frame, we have defined an attributed graph of 30 nodes using the 3-nearest neighbour technique. Nodes represent these landmarks and arcs represent proximity. Attributes on nodes are the position of the landmarks and arcs do not have attributes. The cost between nodes $C_v[v_i^1, v_a^2]$ is the Euclidean distance of their image positions. The cost between arcs $C_e[e_{ij}^1, e_{ab}^2]$ is 0 if both arcs exist or do not exist and 1 if only one of the arcs exists. We have used all pairs of graphs that have been extracted from images that the separation between frames is 60. The final result values are the average of these experiments. We have used the Graduated Assignment algorithm [24] and, in each iteration of the active algorithm, we only permitted a maximum of 30 iterations of the external loop and 20 iterations of the internal loop.

We assess the quality of the current labelling through the Hamming distance between the current labelling and the hand-made labelling [29]. Figure 3 shows this Hamming distance throughout the number of iterations of the active algorithm and using the four previously commented active strategies and also a random strategy. In this random strategy, the *Active Learning* sequentially returns the nodes of g^1 . The algorithm stops when all the 30 nodes have been queried.

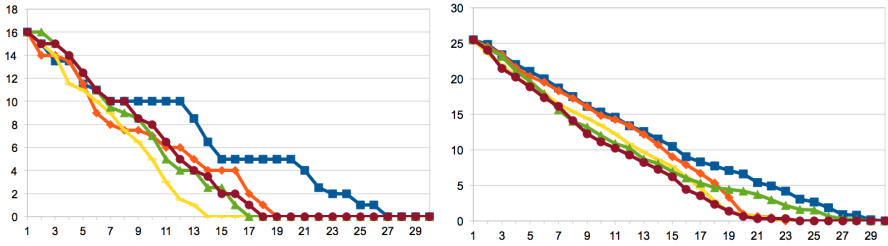


Fig. 3. Hamming distance respect of the number of iterations on the Hotel and House. LCCL: —, LC: —, ME: —, EMC: — and Random: —.

Table 1 shows the ratio between the initial hamming distance with respect to the maximum number of iterations. For instance, in the case of Hotel and LCCL, initial hamming distance = 16, number of iterations = 14, so $16/14 = 1.14$. This value represents the average decrease of the hamming distance in each iteration. The case that the value is higher than 1 appears when, in average and in each iteration, not only the labelling of the queried node is amended but also other ones. It seems as the LCCL method obtains the best results.

Looking at table 1 and figure 1 we realise that the random method, that is, without “intelligent” active learning, obtains the worst results. Moreover, the EMC method is very sensitive to the number of iterations of the Graduated Assignment. This method can deduct few information if most of the probabilities are 0 or 1 (a lot of iterations) or if most of the iterations are near to $1/n$ being n the number of nodes (few iterations).

Table 1. Ratio between hamming distance and iterations of the Hotel and House

	LCCL	LC	ME	EMC	Random
Hotel	1.14	0.84	0.88	0.94	0.59
House	1.10	1.10	1.06	0.87	0.85
Average	1.12	0.97	0.97	0.90	0.72

Figure 4 shows the evolution of the current labelling cost C_f . As described in the algorithm, this cost is computed through the original costs. All methods tend to obtain the optimal labelling, for this reason, at the end, the cost is similar for all of them. Note that in some steps, the cost increases. This means that forcing some nodes to be mapped, the sub-optimal graph-matching algorithm finds a worse labelling.

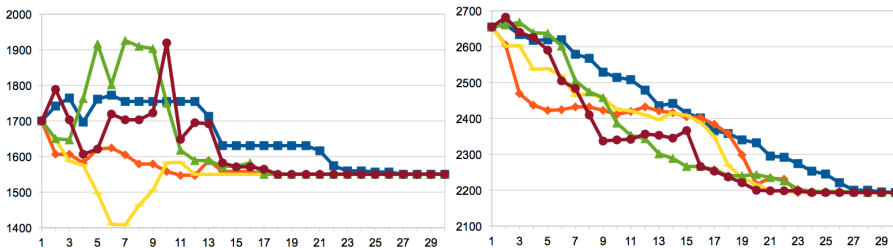


Fig. 4. Labelling cost respect of the number of iterations on the Hotel and House. LCCL: —, LC: —, ME: —, EMC: — and Random: —.

6 Conclusions and Future Work

We have presented four different strategies to be applied on an active graph-matching algorithm. These strategies are based on classical active machine learning but they are applied to the case of searching for the best labelling between nodes. Moreover they are based on the probability matrix between nodes that some sub-optimal algorithms use to iteratively find the best labelling. Due to the active algorithm only updates the costs between nodes and arcs and reads the probability matrix; it is not needed to modify the code of these well-known algorithms. Experimental validation shows that the Least Confident method that uses the current labelling (LCCL) tends faster to find the optimal labelling.

We have planned to apply this method to Expectation-Maximisation and Probabilistic Relaxation algorithms. Moreover, we have obtained different results while using different number of iterations of the two main loops of the Graduated Algorithm. These results have not been shown due to space problems and we have shown in this paper the ones we considered being the best ones. We wish to publish in a journal the whole method presented in this paper together with [31] and the other experiments commented before.

References

1. Mitchell, T.: Machine Learning. Mcgraw-Hill International Edit (1997) ISBN-13: 978-0071154673
2. 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), 265–298 (2004)
3. Settles, B.: Active Learning Literature Survey. Computer Science Technical Report 1648, University of Wisconsin-Madison
4. Wang, R., Kwong, S., Chen, D.: Inconsistency-based active learning for support vector machines. *Pattern Recognition* 45(10), 3751–3767 (2012)
5. Gorisse, D., Cord, M., Precioso, F.: SALSAS: Sub-linear active learning strategy with approximate k-NN search. *Pattern Recognition* 44(10-11), 2343–2357 (2011)
6. Yu, D., Varadarajan, B., Deng, L., Acero, A.: Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion. *Computer Speech & Language* 24(3), 433–444 (2010)
7. Zhang, Q., Shiliang: Sun Multiple-view multiple-learner active learning. *Pattern Recognition* 43(9), 3113–3119 (2010)
8. Wang, Z., Yan, S., Zhang, C.: Active learning with adaptive regularization. *Pattern Recognition* 44(10-11), 2375–2383 (2011)
9. Patra, S., Bruzzone, L.: A cluster-assumption based batch mode active learning technique. *Pattern Recognition Letters* 33(9), 1042–1048 (2012)
10. Zhang, D., Wang, F., Shi, Z., Zhang, C.: Interactive localized content based image retrieval with multiple-instance active learning. *Pattern Recognition* 43(2), 478–484 (2010)
11. Kanamori, T.: Pool-based active learning with optimal sampling distribution and its information geometrical interpretation. *Neurocomputing* 71(1-3), 353–362 (2007)

12. Baranes, A., Oudeyer, P.-Y.: Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems* (available online, 2012)
13. Cord, M., Gosselin, P.H., Philipp-Foliguet, S.: Stochastic exploration and active learning for image retrieval. *Image and Vision Computing* 25(1), 14–23 (2007)
14. Lughofer, E.: Hybrid active learning for reducing the annotation effort of operators in classification systems. *Pattern Recognition* 45(2), 884–896 (2012)
15. Cheng, J., Wang, K.: Active learning for image retrieval with Co-SVM. *Pattern Recognition* 40(1), 330–334 (2007)
16. da Silva, A.T., Falcão, A.X., Magalhães, L.P.: Active learning paradigms for CBR systems based on optimum-path forest classification. *Pattern Recognition* 44(12), 2971–2978 (2011)
17. Gosselin, P.H., Precioso, F., Philipp-Foliguet, S.: Incremental kernel learning for active image retrieval without global dictionaries. *Pattern Recognition* 44(10-11), 2244–2254 (2011)
18. King, R.D., et al.: Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature* 427(6971), 247–252 (2004)
19. Yu, H.: SVM selective sampling for ranking with application to data retrieval. In: *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 354–363. ACM Press (2005)
20. Kotsiantis, S.B.: Supervised Machine Learning: A Review of Classification Techniques. *Informatica* 31, 249–268 (2007)
21. Sanchís, A., Juan, A., Vidal, E.: A Word-Based Naïve Bayes Classifier for Confidence Estimation in Speech Recognition. *IEEE Transactions on Audio, Speech & Language Processing* 20(2), 565–574 (2012)
22. Toselli, A.H., Romero, V., Pastor, M., Vidal, E.: Multimodal interactive transcription of text images. *Pattern Recognition* 43(5), 1814–1825 (2010)
23. Fekete, G., Eklundh, J.O., Rosenfeld, A.: Relaxation: Evaluation and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 3(4), 459–469 (1981)
24. Gold, S., Rangarajan, A.: A Graduated Assignment Algorithm for Graph Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(4), 377–388 (1996)
25. Luo, B., Hancock, E.R.: Structural graph matching using the EM algorithm and singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(10), 1120–1136 (2001)
26. Sanromà, G., Alquézar, R., Serratos, F., Herrera, B.: Smooth Point-set Registration using Neighbouring Constraints. Available on line *Pattern Recognition Letters* (2012)
27. Sanromà, G., Alquézar, R., Serratos, 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)
28. Romero, V., Toselli, A.H., Vidal, E.: Multimodal interactive handwritten text transcription. *Word Scientific* (2012)
29. Caetano, T.S., Caelli, T., Schuurmans, D., Barone, D.A.C.: Graphical Models and Point Pattern Matching. *IEEE Trans. Pattern Analysis and Machine Intelligence* 28(10), 1646–1663 (2006)
30. Kuhn, H.W.: The Hungarian method for the assignment problem *Export. Naval Research Logistics Quarterly* 2(1-2), 83–97 (1955)
31. Serratos, F., Cortés, X., Solé-Ribalta, A.: Interactive Graph Matching by means of Imposing the Pairwise Costs. In: *International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan* (accepted for publication, 2012)