

## Registering sets of points using Bayesian regression

Demetrios Gerogiannis, Christophoros Nikou\*, Aristidis Likas

Department of Computer Science, University of Ioannina, PO Box 1186, 45110 Ioannina, Greece

### ARTICLE INFO

#### Article history:

Received 20 October 2011

Received in revised form

20 February 2012

Accepted 22 February 2012

Communicated by C. Fyfe

Available online 17 March 2012

#### Keywords:

Registration of sets of points

Relevance vector machine (RVM)

Hungarian algorithm

### ABSTRACT

This work addresses the problem of non-rigid registration between two 2D or 3D points sets as a novel application of Relevance Vector Machines (RVM). An iterative framework is proposed which consists of two steps: at first, correspondences between distinct points are estimated by the Hungarian algorithm and then a regression procedure based on a Bayesian linear model (RVM) maps the two sets of points. By these means, a large variety of transformation is captured without imposing any prior knowledge on the form of the point sets. The proposed algorithm provides a smooth transformation even if the correspondence between the points in the two sets contains erroneous matches. The algorithm was successfully evaluated on sets of points with varying difficulty and favorably compared with state-of-the-art methods in cases of noise.

© 2012 Elsevier B.V. All rights reserved.

### 1. Introduction

Registration of two sets of points is a common step in many applications in computer vision, pattern recognition, image processing and medical image analysis. The problem consists in determining a geometrical transformation that brings two sets of points into alignment. This could be achieved, for instance, through the establishment of correspondences. However, the problem is not always well-posed and becomes more complicated by the existence of noise or outliers, making the determination of correspondences harder. Another drawback rises from the geometric transformation itself, as there may be an infinite number of allowed high dimensional mappings. Also, the definition of the similarity measure is an open issue, since one can choose from a variety of metrics.

Many methods have been proposed to solve the correspondence problem. A straightforward approach is based on the nearest neighbor criterion to establish correspondences, as in the Iterated Closest Point (ICP) algorithm [1]. However, despite its simplicity, this method results in many local minima, providing a suboptimal solution, and does not guarantee that the correspondence is one-to-one. Many variants of this algorithm have been proposed improving the behavior of the method in the presence of noise. A detailed review can be found in [2]. Nevertheless, in all cases, ICP methods necessitate a good initialization near to the optimal solution in order to prevent the energy function from getting trapped in local minima.

\* Corresponding author. Tel.: +30 26510 08802.

E-mail addresses: [dgerogia@cs.uoi.gr](mailto:dgerogia@cs.uoi.gr) (D. Gerogiannis), [cnikou@cs.uoi.gr](mailto:cnikou@cs.uoi.gr) (C. Nikou), [arly@cs.uoi.gr](mailto:arly@cs.uoi.gr) (A. Likas).

The Robust Point Matching (RPM) algorithm [3] relies on a deterministic annealing scheme. The algorithm applies the soft-assign principle [4] for matching and the thin-plate spline interpolation [5] for non-rigid mapping. The rationale is to transfer the assignment problem from a hard approach to a soft one, that is to define a probability for each assignment.

The Coherent Point Drift (CPD) algorithm was also proposed in [6], where the registration is treated as a Maximum Likelihood (ML) estimation problem with motion coherence constraints over the velocity field such that one point set moves coherently in order to be aligned with the other. In that case, transformation parameter estimation and determination of correspondences are simultaneously handled.

Mixture models were proposed as a framework to solve the registration problem (GMMReg) [7]. Each set of points is represented by a mixture of Gaussians and registration is defined as a problem of aligning the two mixtures. The  $L_2$  metric is used as a measure of mixture alignment. An extension of the method using robust Student's- $t$  modeling for the data was also proposed in [8].

Shape context was considered in [9], where an iterative algorithm is designed to account for the shape matching, registration and detection. The problem is formulated in terms of probabilistic inference using a generative model and the EM algorithm. Shape features are used in a data-driven technique to address the problem of initialization.

A technique for establishing correspondences is proposed in [10], where features of a 2D point set which are invariant with respect to a projective transformation are extracted. The proposed algorithm is based on the comparison of two projective and permutation invariants of five-tuples of the points. The best-matched five-tuples are then used for the computation of the

projective transformation and the one having the maximum number of corresponding points is used.

Moreover, in [11], a novel technique is introduced to solve the rigid point pattern matching problem in Euclidean spaces of any dimension. The point pattern matching is modeled as a weighted graph, where nodes represent points and the weights of the edges are equal to the Euclidean distances between nodes. The graph matching is formulated as a problem of finding a maximum probability configuration in a graphical model.

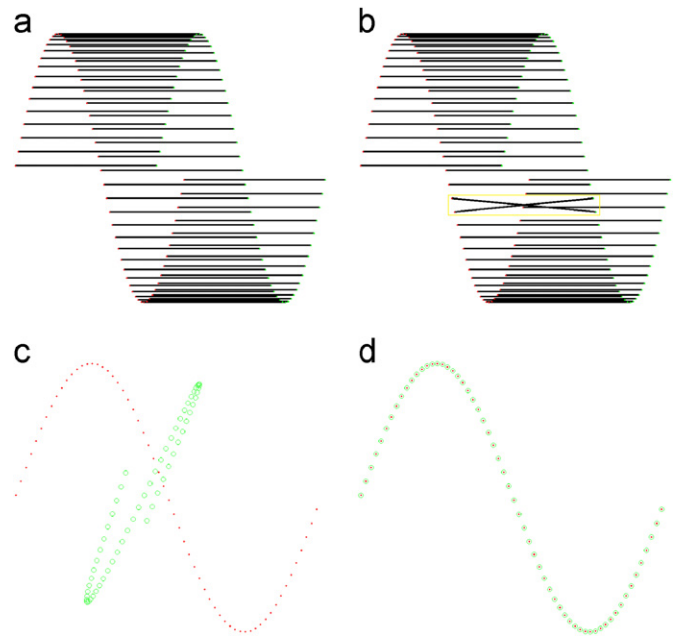
In [12], the notion of a neighborhood structure for the general point matching problem is introduced. Then, the point matching problem is formulated as an optimization problem to preserve local neighborhood structures during matching. The method has a simple graph matching interpretation, where each point is a node in the graph, and two nodes are connected by an edge if they are neighbors. The optimal match between two graphs is the one that maximizes the number of matched edges.

The majority of the registration methods mentioned so far, model the non-rigid mapping through a spline interpolation method, and in particular with the thin-plate spline (TPS) [5]. In this work, we consider the transformation parameter estimation issue as a regression problem and a Bayesian model, namely Relevance Vector Machine (RVM) [13] is used to solve this problem. We consider here the standard RVM although the method may employ other variants such as the twinned RVM [14] which applies double training or the multivariate RVM [15].

Our work is motivated by the pioneering research presented in [16] where correspondences are estimated using a softassign approach. Softassign is a technique for solving an assignment problem. As opposed to hard assignment, softassign weights each matching to indicate the quality of the correspondence. Hard assignment is the limit version of softassign. In the work herein, instead of solving the assignment problem based on the smallest distance, we utilize this distance to create a cost matrix that describes the cost of an assignment. Then, the correspondences are extracted with a combinatorial optimization algorithm, the Hungarian algorithm [17]. The rationale behind this algorithm is to assign a single task to a single worker, based on an assignment cost matrix, such that the total cost is minimum. The temporal complexity of the algorithm is polynomial, and in particular  $\mathcal{O}(n^3)$ . After the correspondence between points has been established, a Bayesian regression model (RVM) is used to infer the transformation parameters.

The Hungarian algorithm has also been used in [18], where a feature-based registration method is demonstrated. Points are assumed to describe a shape and a histogram (*shape context*) is calculated, describing the space distribution of points. This histogram is used to define the cost matrix of the Hungarian algorithm. Our work differentiates from [18] in the way the geometric transformation is treated. We estimate the transformation's parameters (both rigid and non-rigid) through regression (RVM) while the latter method uses thin-plate splines interpolation. Another substantial difference is that in our approach points are not considered as parts of a shape representation, since our method is more general.

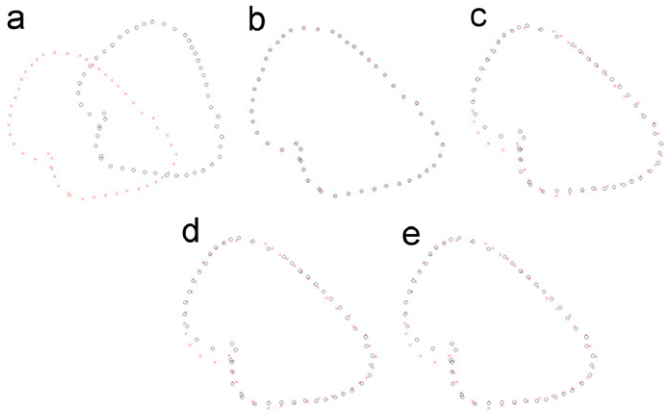
The main contribution of this work is that using a regression framework based on RVM addresses the problem of eventual false correspondences with respect to methods relying on interpolation schemes, like TPS. More precisely, a single false correspondence may lead to a totally erroneous registration if TPS is used. For example, this is a case of the RPM [3], or the GMMReg [7] methods. In Fig. 1(a), the correct correspondences between the source and the target sets are represented by line segments. In Fig. 1(b), two points were falsely matched on purpose simulating a wrong correspondence solution. The result of TPS [5] is shown in Fig. 1(c), where large registration errors are present. The result



**Fig. 1.** A false matching simulation example, with a point set used in [3]. (a) Correct correspondences between the source and the target sets are represented by line segments. (b) Two points were falsely matched on purpose simulating a wrong correspondence solution. The yellow box depicts the false matched points. (c) The result of TPS [5]. Notice that large registration errors are present. (d) The result of the proposed registration scheme was based on RVM regression. In this case the registration is correct. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of the proposed registration scheme based on RVM regression is shown in Fig. 1(d), where the registration is correct. TPS by its definition tries to minimize the total bending energy to provide a smooth model, which is an approach that restricts the capability of providing good results in areas where the correspondence is correctly established. On the other hand, RVM considers only the local neighborhood to extract the regression model, due to the priors it implies on each point. The closed form solution for the transformation model provided by RVM is continuous and locally smooth depending on the assignment solution and more importantly, it is robust to false matches. The correspondence estimation step used in this work is the Hungarian algorithm. Alternative methods could also be used to solve the correspondence problem, like the standard softassign approach [4].

The proposed method is similar in spirit with RPM [3] and CPD [6] in the sense that it employs a framework of iteratively updating the correspondence and the estimation of the transformation parameters. Both RPM and CPD are based on an expectation-maximization (EM) [19] framework. In RPM and CPD the E-step estimates soft correspondences through a posterior distribution. In our method the E-step involves any correspondence estimation algorithm, which in our case is the Hungarian algorithm. A major difference in our approach with respect to RPM and CPD is that in the M-step the transformation is estimated using Bayesian regression (RVM). On the other hand RPM uses TPS interpolation. Moreover, RVM provides a closed form transformation both for the rigid and non-rigid cases compared to CPD, where the two cases have to be modeled with different set of parameters [6]. A modeling of a rigid registration case with a non-rigid model may provide inaccuracies in the CPD result. An example is presented in Fig. 2, where a rigid transformation by a rotation of  $25^\circ$  and a translation vector  $[0.2, 0.1]^T$  is applied on a point set used in [3]. The results of CPD by modeling the



**Fig. 2.** A rigid registration of a point set used in [3]. See text for transformation details. (a) The initial point sets. The CPD [6] registration result by modeling the transformation (b) with a rigid model (c) with a non-rigid model. Notice that in the second case huge registration errors occur. The GMMReg [7] registration result by modeling the transformation (d) with a rigid model (e) with a non-rigid model. The corresponding solution of our method is similar with the rigid solution in (b).

transformation with a rigid and non-rigid model are shown in Fig. 2(b) and (c) respectively. Notice that in the case where the rigid transformation was modeled by a non-rigid set of parameters some registration errors are present, as we try to model a simple transformation with a more complex model. The corresponding solution of our method is similar with the rigid solution in Fig. 2(b). The same also applies for the GMMReg methodology [7] (Fig. 2(d) and (e)).

We evaluated our method by comparing it with the CPD [6], the RPM [3] and the GMMReg [7] algorithms for both rigid and non-rigid transformations. The results indicate that our method is more accurate than the state-of-the-art methods compared concerning the robustness against false matching during the correspondence estimation step and the parametrization of the method. The innovation of our method is that by utilizing a robust correspondence estimation algorithm initially, we could relax the constraints imposed in the transformation modeling step so as to handle any erroneously matched points.

The remainder of the paper is organized as follows: In Section 2, we formulate the problem and preview the theory of the Hungarian algorithm (Section 2.1) and the RVM (Section 2.2). Experimental results are presented in Section 3 and conclusions are drawn in Section 4.

## 2. Registration of sets of points via regression

In this section we describe the theoretical basis of the proposed method. At first, the Hungarian algorithm and the Relevance Vector Machines approach are reviewed and then we describe how the regression method is integrated within a registration framework.

### 2.1. The Hungarian algorithm

The Hungarian algorithm is a combinatorial optimization method which solves the assignment problem. Assume that there are  $m$  tasks that have to be assigned to  $n$  workers. Each assignment is weighted with a cost (or profit), thus a complete bipartite weighted graph is produced, having as vertices the workers and the tasks. The goal is to calculate that particular assignment such that the total cost is minimum. The assignment has to be one to one. Sometimes the algorithm is used to maximize the total profit.

In that case, we subtract the maximum entry of the cost matrix from all its cells. In case  $m \neq n$ , the problem is called unbalanced and the standard Hungarian algorithm may provide a false solution. A modification of the algorithm to handle rectangular cost matrices is introduced in [20]. Algorithm 2 presents the steps of this modification. In Algorithms 1 and 2 the terms starred, primed, covered and uncovered are characterizations assigned to a zero element (starred, primed) or to rows and columns (covered, uncovered), that guides the execution of the algorithm and distinguish the examined elements (zeros and rows or columns). The algorithm along with a detailed description may also be found in [20].

Hereafter we consider that  $m=n$ , and thus we will exploit only variable  $n$  to indicate the dimension of the problem. The output of the Hungarian algorithm is the optimal assignment, that minimizes the total cost. The complexity of the algorithm is  $\mathcal{O}(n^3)$  in case of a balanced problem, while it may be increased in case of unbalanced problems, as a lot of trials are made to extract the solution of the problem.

More specifically, suppose we have a weighted undirected bipartite graph with  $n$  nodes, with  $c_{ij}$  indicating the weight of edge from node  $i$  to node  $j$ . The variable  $\delta_{ij}$ , where  $i, j \in \{1, \dots, n\}$  indicates whether edge  $(i, j)$  is included in the matching. More specifically,  $\delta_{ij} = 1$  means that the corresponding edge is included in the matching, whereas  $\delta_{ij} = 0$  signifies that the edge  $(i, j)$  is not part of the matching process. The following restrictions apply:

- $\sum_{i=1}^n \delta_{ij} = 1$ ,
- $\sum_{j=1}^n \delta_{ij} = 1$ ,
- $\delta_{ij} > 0, \forall i, j \in \{1, \dots, n\}$ .

The goal of the Hungarian algorithm is the following:

Given a  $n \times n$  matrix  $C$ , where  $C_{ij}$  is the weight of assigning worker  $i$  with task  $j$ , minimize  $\sum_{i=1}^n \sum_{j=1}^n \delta_{ij} C_{ij}$ .

The steps of the Hungarian algorithm, or Hungarian method as it is met regularly in the literature are described in Algorithm 1. Details may be found in [17].

**Algorithm 1.** The Hungarian algorithm for square cost matrices.

- 1 From each row subtract off the row min.
- 2 From each column subtract off the row-column min.
- 3 Use as few lines (vertical, horizontal) as possible to cover all rows and columns containing zeros in the matrix (trial and error). Suppose  $k$  lines are used for covering.
  - if  $k < n$ , let  $m$  be the minimum uncovered number. Subtract  $m$  from every uncovered number. Add  $m$  to every number covered with two lines. Goto step 3.
  - if  $k = n$ , goto step 4.
- 4 Starting with the top row, go downwards making assignments. An assignment can be (uniquely) made **only when** there is **exactly** one zero in the row.

In original version, the Hungarian algorithm assumes a square cost matrix, i.e. equal number of tasks and workers. A modification of the algorithm to handle rectangular cost matrices is introduced in [20], solving thus problems with different number of workers and tasks (unbalanced problems). Algorithm 2 presents the related procedure. The reader should notice that our goal is to propose a method that can model a registration transformation upon an assignment between two point sets has been determined. The Hungarian algorithm is a solution to that problem. In order to provide a complete framework, the revised Hungarian algorithm, that handles unbalanced sets is also included in our work.

**Algorithm 2.** The Hungarian algorithm for rectangular cost matrices (unbalanced problems).

- 0 Let  $k = \min(n, m)$  and  $l = \max(n, m)$  for a cost matrix  $A$ ,  $m \times n$ .
- 1 If number of rows is larger than number of columns goto 3.
- 2 If number of rows is less than number of columns goto 5.
- 3 Update  $A$ : For each row of  $A$ , subtract the minimum element from each element in the row.
- 4 Update  $A$ : For each column of  $A$ , subtract the minimum element from each element in the column.
- 5 Find a zero at location  $Z$  of the matrix  $A$ . If there is no starred zero in its row nor its column then star  $Z$ .  
Repeat step 5 for each zero of the matrix.
- 6 Cover every column containing a  $0^*$ . If  $k$  columns are covered, the starred zeros form the desired independent set (assignment solution). STOP
- 7 Choose a non-covered zero and prime it; then consider the row containing it.
  - If there is no starred zero  $Z$  in this row, go to step [9].
  - If there is a starred zero  $Z$  in this row, cover this row and uncover the column of  $Z$ .  
Repeat step 8 until all zeros are covered.
- 8 Go to step 11.
- 9 There is a sequence of alternating starred and primed zeros constructed as follows:
  - Let  $Z_0$  denote the uncovered  $0'$ .
  - Let  $Z_1$  denote the  $0^*$  in  $Z_0$ 's column (if any).
  - Let  $Z_2$  denote the  $0'$  in  $Z_1$ 's row.
 Continue in a similar way until the sequence stops at a  $0'$ ,  $Z_{2k}$ , which has no  $0^*$  in its column.  
Unstar each starred zero of the sequence, and star each primed zero of the sequence.  
Erase all primes and uncover every line.
- 10 Return to step 6.
- 11 Let  $h$  denote the smallest non-covered element of the matrix; it will be positive.  
Add  $h$  to each covered row; then subtract  $h$  from each uncovered column.  
Return to step 7 without altering any asterisks, primes, or covered lines.

### 2.2. Relevance Vector Machines

The RVM model can be used to solve either the problem of classification or regression. In general, in order to use a RVM, we have to assume that we have a set of examples of input vectors  $X = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$  along with corresponding scalar targets  $\mathbf{t} = \{t_i\}_{i=1}^N$ . Our goal is to train a model so as to learn the functional mapping between input vectors  $\mathbf{x}_i$  and targets  $t_i$ . Since the points in a registration problem lay in a continuous space, it is implied that the target variable  $\mathbf{t}$  is continuous, leading to a regression problem. A detailed description of RVM theory may be found in [13,21].

More specifically, we seek that particular model  $f$  with parameters  $\mathbf{w} = \{w_1, w_2, \dots, w_N\}$  such that  $f(\mathbf{x}_i; \mathbf{w}) \simeq t_i, i = 1, \dots, N$ , assuming that  $\mathbf{x}_i$  corresponds to  $t_i$ . The model  $f$  may be analyzed into a finite linear sum of  $N$  non-linear functions  $\phi_j$ , called kernels. Thus,

$$f(\mathbf{x}_i; \mathbf{w}) = \sum_{j=1}^N w_j \phi_j(\mathbf{x}_i) = \mathbf{w}^T \Phi(\mathbf{x}_i), \quad (1)$$

where  $\Phi(\mathbf{x}_i) = (\phi_1(\mathbf{x}_i), \phi_2(\mathbf{x}_i), \dots, \phi_N(\mathbf{x}_i))^T$ .

Assume now that the targets  $\{t_i\}_{i=1}^N$  are samples drawn from the model with additive noise  $\epsilon_i$ :

$$t_i = f(\mathbf{x}_i; \mathbf{w}) + \epsilon_i, \quad (2)$$

where  $\epsilon_i$  are independent samples from some noise process. Hereafter we will assume a Gaussian distribution with zero mean and variance  $\sigma^2$  for  $\epsilon_i$ . Thus, a probability density model occurs:

$$p(t_i | \mathbf{x}_i) = \mathcal{N}(t_i | f(\mathbf{x}_i; \mathbf{w}), \sigma^2), \quad (3)$$

where  $\mathcal{N}$  is a Gaussian distribution over  $t_i$  with mean  $f(\mathbf{x}_i; \mathbf{w})$  and variance  $\sigma^2$ .

A second assumption concerns the statistical independence of target variables  $t_i$ . The likelihood of the target vector  $\mathbf{t}$  is

$$p(\mathbf{t} | \mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2} \|\mathbf{t} - \Phi\mathbf{w}\|^2\right\}, \quad (4)$$

where  $\mathbf{t} = (t_1 \dots t_N)^T$ ,  $\mathbf{w} = (w_1 \dots w_N)^T$  and  $\Phi = (\Phi(x_1) \dots \Phi(x_N))$ .

In Bayesian methodology, a common practice to prevent overfitting, caused by the large number of parameters, is to impose some additional constraints, penalizing the complexity of the model. These hyperparameters are imposed over parameters  $\mathbf{w}$  of the linear model in (1). The goal is to reduce the number of discrete functions of the sum, thus occurring a less complex model. This is achieved by adopting a zero-mean Gaussian prior over  $\mathbf{w}$ , or

$$p(\mathbf{w} | \mathbf{a}) = \prod_{i=1}^N \mathcal{N}(w_i | 0, a_i^{-1}), \quad (5)$$

where  $\mathbf{a} = (a_1 \dots a_N)^T$  with  $a_i$  representing the precision of the corresponding parameter  $w_i$ . One can explain these hyperparameters as selectors over each parameter  $w_i$  which is the weight of function  $\phi_i$  participating in the total sum. If the variance of the corresponding prior is large then the resulting probability is low, eliminating the term in the sum. This means that the corresponding basis function  $\phi_i(\mathbf{x}_j)$  plays no role in the prediction made by the model.

The posterior distribution of weights is Gaussian and takes the form

$$p(\mathbf{w} | \mathbf{t}, X, \mathbf{a}, \beta) = \mathcal{N}(\mathbf{w} | \mathbf{m}, \Sigma), \quad (6)$$

where  $\beta$  is the inverse of  $\sigma$  in (4) and

$$\mathbf{m} = \beta \Sigma \Phi^T \mathbf{t}, \quad (7)$$

$$\Sigma = (\mathbf{A} + \beta \Phi^T \Phi)^{-1}, \quad (8)$$

with  $\mathbf{A} = \text{diag}\{a_i\}$ .

Eventually, an iterative learning process occurs. Initially, we choose some values for  $\mathbf{a}$ ,  $\beta$ , thus evaluating the mean and covariance of the posterior using (7) and (8). Then we iterate, until a convergence criterion is satisfied, by re-estimating the hyperparameters:

$$a_i = \frac{\gamma_i}{m_i^2}, \quad (9)$$

$$\beta^{-1} = \frac{\|\mathbf{t} - \Phi\mathbf{m}\|^2}{N - \sum_{i=1}^N \gamma_i}, \quad (10)$$

where  $m_i$  is the  $i$ th component of the posterior mean defined by (7) and the quantity  $\gamma_i$  is computed as

$$\gamma_i = 1 - a_i \Sigma_{ii}, \quad (11)$$

where  $\Sigma_{ii}$  is the  $i$ th diagonal component of the covariance matrix  $\Sigma$  given by (8).

The result of the training process described above is learning parameters  $\mathbf{w}$  of Eq. (1).

### 2.3. Registration by regression

In a point set registration problem two sets of points are involved. The source point set  $X = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^{N_x}$  and the target

point set  $T = \{\mathbf{t}_i \in \mathbb{R}^d\}_{i=1}^{N_t}$ . In our experiments, we assume that  $N_x = N_t = N$ . In case the two sets have different cardinalities, we add extra points (as described in Section 2.1). In our method the registration transformation is modeled by a RVM. However, one would observe that the RVM described in Section 2.2 is defined for univariate output vectors. In other words, the target variable has to be a scalar. However, in our case  $\mathbf{t}_i \in \mathbb{R}^d$ , and thus, in order to overcome this difficulty, we used  $d$  distinct RVMs, one for each dimension  $k$ . Alternatively, one could use a multivariate RVM, as described in [15] or the twinned RVM [14].

Eventually, the proposed model is a *vector valued* function  $\mathcal{T}$ , having parameters  $W$ , representing the geometric transformation bringing set  $X$  into alignment with set  $T$ . Thus, ideally we would have for every point  $\mathbf{t}_i \in T$ ,  $i = 1, \dots, N$ :

$$\mathbf{t}_i = \mathcal{T}(\mathbf{x}_{C_i}; W) = [\mathcal{T}^1(\mathbf{x}_{C_i}; \mathbf{w}^1), \dots, \mathcal{T}^k(\mathbf{x}_{C_i}; \mathbf{w}^k), \dots, \mathcal{T}^d(\mathbf{x}_{C_i}; \mathbf{w}^d)]^T, \quad (12)$$

where  $W = \{\mathbf{w}^k \in \mathbb{R}^N\}_{k=1}^d$ , with  $\mathbf{w}^k$  being the weight vector of dimension  $N$  for the  $k$ th RVM,  $\mathcal{T}^k$  is a RVM as described by (1) and  $C_i$  is the index of the point in  $X$  corresponding to the  $i$ -th point of  $T$ . In other words, the ideal transformation is

$$t_i^k = \mathcal{T}^k(\mathbf{x}_{C_i}; \mathbf{w}^k), \quad i = 1, \dots, N, \quad k = 1, \dots, d, \quad (13)$$

with  $t_i^k$  representing the  $k$ th component of point  $\mathbf{t}_i \in T$ .

The proposed method consists of an iterative scheme, that, at each iteration alternates between the method for establishing correspondences (e.g. Hungarian algorithm) and the method for estimating the registration transformation (RVM training). The corresponding objective function that is minimized has the following form:

$$J(\delta, W) = \sum_{i=1}^N \sum_{j=1}^N \delta_{ij} \mathcal{C}_{\mathbf{x}_i, \mathcal{T}(\mathbf{x}_j; W)} + \sum_{i=1}^N \sum_{j=1}^N \delta_{ij} \|\mathbf{t}_i - \mathcal{T}(\mathbf{x}_j; W)\|^2. \quad (14)$$

Its optimization involves two steps at each iteration. In the first step, we assume a known registration transformation  $\mathcal{T}$  (RVM) and try to estimate the optimal correspondences  $\delta_{ij}$  with the Hungarian algorithm. Thus, the objective function is minimized with respect to  $\delta_{ij}$ ,  $\forall i, j$ . In the second step, the correspondences  $\delta_{ij}$  are fixed to the values computed in the first step and a RVM training process takes place to update the registration transformation  $\mathcal{T}$  in order to match the estimated correspondences. Thus, in this second step, the objective function is minimized with respect to the set of weights  $W = \{\mathbf{w}^k \in \mathbb{R}^N\}_{k=1}^d$ . Since both computational steps at each iteration minimize the objective function  $J$ , the whole iterative process converges to a minimum of  $J$ .

The overall procedure is presented in Algorithm 3. Each iteration of the registration algorithm involves two steps. At first, correspondences between points of the source set  $X$  and the target set  $T$  are estimated by the Hungarian algorithm and then based on these correspondences,  $d$  RVMs are trained, one per dimension, to solve the regression problem of transforming the source set to the target set. We initialize the registration transformation as the identity mapping.

**Algorithm 3.** The RVM–Hungarian method for registration of sets of points.

- 1 **Initialize the registration transformation as the identity mapping and select the type of basis function  $\phi_i(x)$ ,  $i = 1, \dots, N$  for the RVM.**
- 2 **Determine the correspondences between sets of points  $X, T$ .**
  - 2.1 Calculate distance matrix  $C_{ij} = \|\mathcal{T}(\mathbf{x}_i; W) - \mathbf{t}_j\| \forall i = 1, \dots, N, j = 1, \dots, N$ .
  - 2.2 Solve the assignment problem with the Hungarian algorithm, where  $\mathcal{C}$  is an assignment cost matrix.

### 3 Transformation parameters estimation—train one RVM per dimension of point set $X$ .

For each RVM $_k$ ,  $k = 1, \dots, d$ :

- 3.1 Calculate  $\mathbf{m}^k$  and  $\Sigma^k$  by (7) and (8).
  - 3.2 Calculate  $a^k, \beta^k, \gamma^k$  by (9)–(11).
  - 3.3 Iterate steps (3.1) and (3.2) until convergence to obtain the new RVM $_k$ , with  $\mathbf{w}^k = \mathbf{m}^k$ .
- 4 **Iterate steps 2, 3 until convergence of the objective function  $J(\delta, W)$  (14).**

## 3. Experimental results and discussion

In order to evaluate our method, several experiments were conducted in a collection of sets of points, firstly used in [16], and widely used in the related literature (Fig. 3). The algorithm was tested both for its accuracy and its robustness to noise. Experiments with real data were also conducted. For that purpose we used the 2D range data from [22] (Fig. 4(a)) and the 3D face of [6] was also used in our experiments (Fig. 4(b)).

Experiments are divided into two types, according to the transformation type (either rigid or non-rigid). In case of non-rigid transformation, the non-rigid deformation was followed by a rigid one, to make the problem more challenging. In that case the whole transformation remains non-rigid. In all cases the registration transformation was initialized to the identity mapping. The angle of the rigid transformation varied between  $[0^\circ, 10^\circ]$ , while the translation, varied between  $[-0.2, 0.2] \times [-0.2, 0.2]$ . The registration error is defined as the Euclidean distance between the reference point and its corresponding registered. Points of Fig. 3 range in  $[0, 1] \times [0, 1]$ , while those of Fig. 4(a) range in  $[-40, 10] \times [-10, 30]$  and of Fig. 4(b) in  $[-2, 2] \times [-2, 2] \times [-2, 2]$ .

In our implementation, different kernels were examined (Gaussian, Student's  $t$ -kernel and Laplacian) as described in [13] and implemented in [23]. The Laplacian kernel,  $K(x, y) = e^{-|x-y|/\sigma}$ , proved to be the most efficient model for the tested input data shown in Fig. 3. However, the differences are not considerable as the registration accuracies of the compared methods differ at the third decimal digit. Table 1 presents the registration error statistics of the rigid case, while Table 2 demonstrates the results of the non-rigid case. In all cases, variable kernel widths were used in the range between 5% and 30% of the mean variance of the reference set. The Laplacian kernel proved to be less sensitive to changes in the variations of the kernel width compared to the Gaussian and Student's  $t$ -kernels.

In order to compare our method with the state-of-the-art, our results were compared with the CPD [6], the RPM [3] and the GMMReg [7] algorithms. In this experimental configuration the kernel width  $\sigma$  was set to 20% of the variance of source point set  $X$  for all cases of our experiments.

The code for implementing these algorithms was found in the web pages of the corresponding authors. RPM was implemented in Matlab environment, while CPD and GMMReg were programmed in C/C++ (Mex files). Therefore, this has an impact on the different execution times of the algorithms. Our method was partially implemented in Matlab (RVM training [13], by the official web page of Mike Tipping [23]) and in C (Hungarian algorithm for rectangular and square cost matrices, an implementation found in the Mathwork File Exchange web page). Several experiments were conducted (rigid and non-rigid transformations) and apart from the registration error (root mean squared error) the execution time and convergence rate (i.e. how many iterations were necessary for the algorithm to converge) were also measured. A general conclusion is that the proposed Bayesian regression framework provides better results compared

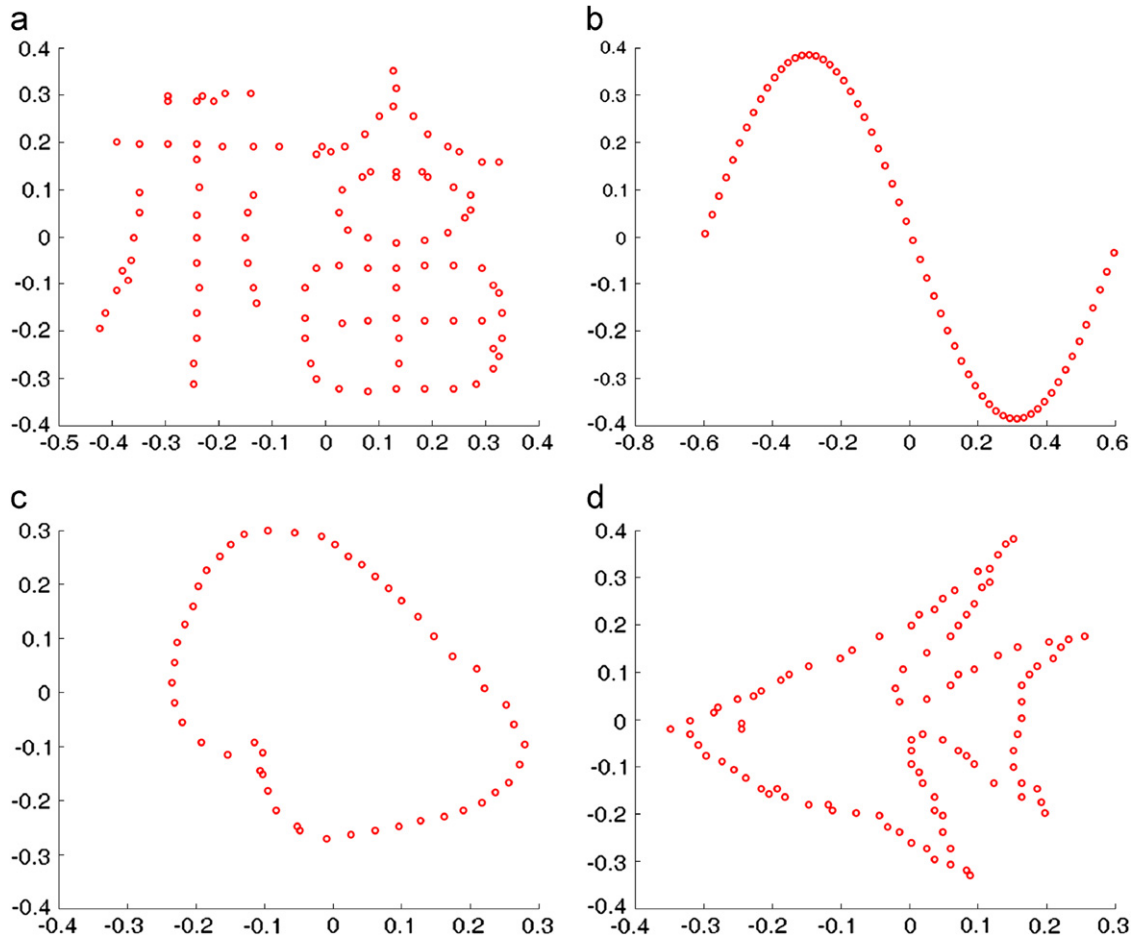


Fig. 3. The initial set of points used in our experiments [3]. (a) Sine, (b) blob, (c) fish and (d) ideogram.

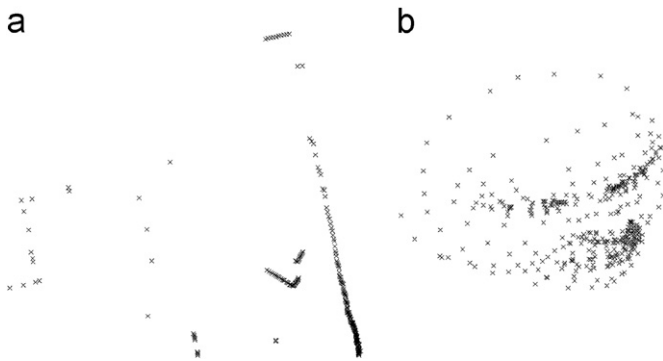


Fig. 4. (a) The 2D range data used in our experiments [22]. (b) The 3D set of points representing a face used in our experiments (3D face) [6].

Table 1  
Registration error statistics for rigid transformations using different kernels on the shapes of Fig. 3. The kernel width varies between 5% and 30% of the mean variance of the reference set.

Kernel	Mean	Std	Median	Min	Max
Gaussian	0.0021	0.0019	0.0013	0.0009	0.0049
Student's <i>t</i>	0.0007	0.0012	0.0001	0.0000	0.0026
Laplacian	0.0000	0.0000	0.0000	0.0000	0.0000

to RPM, where this algorithm either demands an extra post processing refinement step (e.g. registration of the centroids, Fig. 7(c)) or completely fails (e.g. Tables 3, 4, 7, 8).

Table 2  
Registration error statistics for non-rigid transformations using different kernels on the shapes of Fig. 3. The kernel width varies between 5% and 30% of the mean variance of the reference set.

Kernel	Mean	Std	Median	Min	Max
Gaussian	0.0029	0.0022	0.0020	0.0013	0.0061
Student's <i>t</i>	0.0009	0.0015	0.0002	0.0001	0.0031
Laplacian	0.0001	0.0001	0.0000	0.0000	0.0002

Table 3  
Mean registration error for rigid transformations.

Point set	Hungarian-RVM	CPD [6]	RPM [3]	GMMReg [7]
Sine	0.00	0.00	14.62	0.00
Blob	0.00	0.00	11.53	0.00
Fish	0.00	0.00	22.60	0.00
Ideogram	0.00	0.00	23.62	0.00
2D range	0.01	0.04	Fail	0.00

Each experiment was run 20 times and error statistics were calculated. In each configuration, a different registration transformation parameter set was used. The execution times are presented in Table 5, along with the convergence rate in Table 6 for experiments with noise free data and points in presence of noise. The initial sets of points, with representative results are demonstrated in Fig. 5 for the rigid case and in Figs. 6 and 7 for the non-rigid case. Also, to investigate the robustness of the

**Table 4**  
Mean registration error for non-rigid transformations.

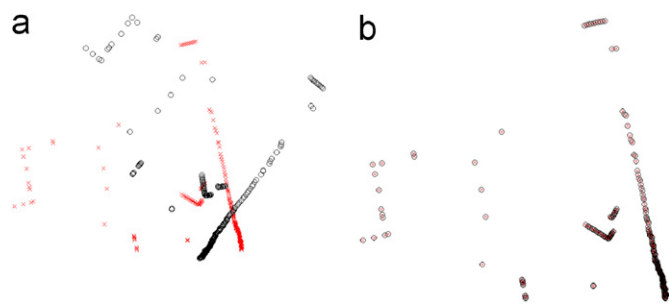
Point set	Hungarian-RVM	CPD [6]	RPM [3]	GMMReg [7]
Sine	0.00	0.00	13.45	0.00
Blob	0.00	0.00	11.19	0.00
Fish	0.00	0.00	21.87	0.00
Ideogram	0.00	0.00	23.50	0.00
3D face	0.00	0.08	Fail	0.00

**Table 5**  
Mean execution time (s) of the compared methods for the whole set of experiments presented in Section 3. The Hungarian-RVM is partially implemented in Matlab (RVM training) and C (Hungarian algorithm). RPM is totally implemented in Matlab while both CPD and GMMReg are totally implemented in C.

Point set	Hungarian-RVM	CPD [6]	RPM [3]	GMMReg [7]
Pure data	0.43	0.08	1.97	0.19
Gaussian noise	0.30	0.08	2.53	0.49

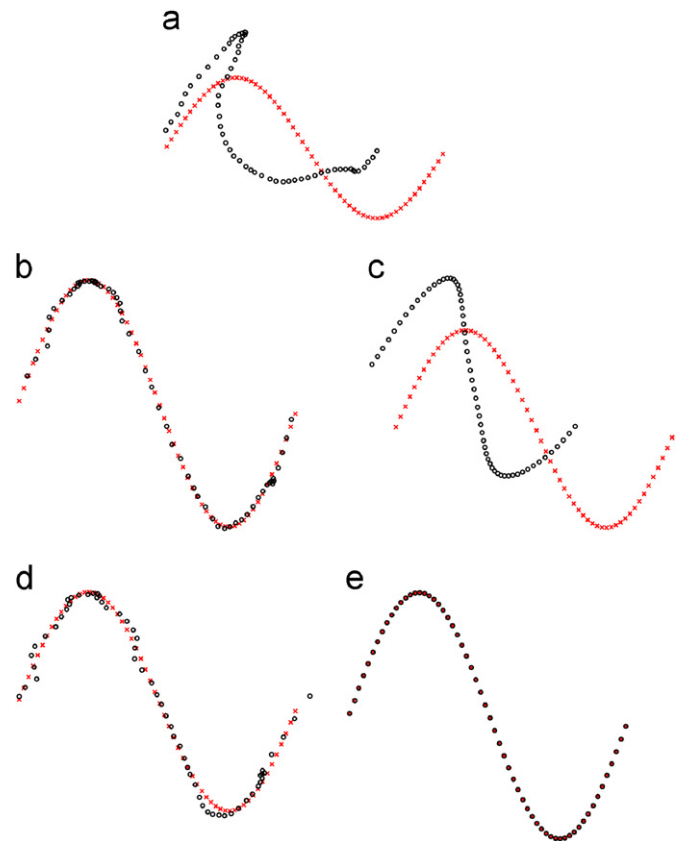
**Table 6**  
Average number of iterations of the compared methods for the whole set of experiments presented in Section 3.

Point set	Hungarian-RVM	CPD [6]	RPM [3]	GMMReg [7]
Pure data	2	21	97	55
Gaussian noise	2	20	87	56



**Fig. 5.** Rigid transformation experiment with 2D points of a range scan [22]. (a) Reference set of points (red) and deformed set of points (black) of a 3D face. (b) Registration result for the proposed method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

algorithm to noise, the points were corrupted by Gaussian noise (with zero mean and small variance so as the shape does not change significantly) (Fig. 8). The initial sets of points, with the estimated results are demonstrated in Figs. 9–12, while the statistics are presented in Tables 3 (rigid case) and 4 (non-rigid case) for noise free points and in Tables 7 (rigid case) and 8 (non-rigid case) for points corrupted by Gaussian noise with zero mean value. To further support the statistical presentation of the registration error results, the  $p$ -value was computed, so as to verify the statistical significance of the analysis. Notice that in case of uncorrupted data, the deviations between real and computed values are too small for all the studied methods, and thus the  $p$ -value was not computed. In the case of data corrupted by Gaussian noise, there are differences between the results provided by each method. As it may be observed in the last row of Tables 7 and 8, in all cases, the computed  $p$ -value is less than a threshold of significance level of 5%, which is usually employed.

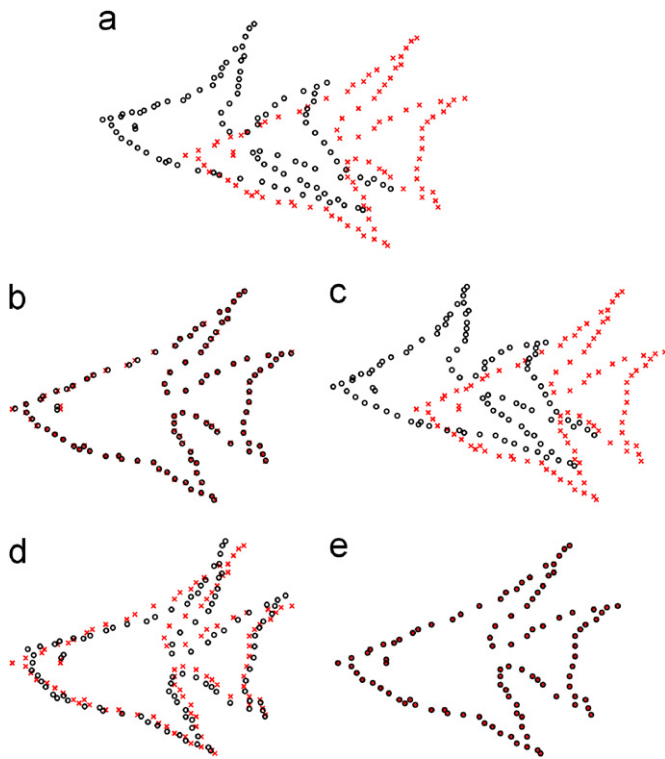


**Fig. 6.** Non-rigid transformation experiment. (a) Reference set of points (red) and deformed set of points (black). Registration result for (b) CPD, (c) RPM, (d) GMMReg and (e) the proposed method. The difference is better highlighted in color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

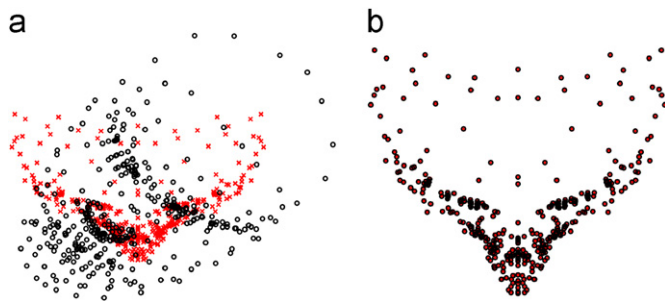
One can observe that the proposed method, provides better results in general compared to CPD, RPM and GMMReg. Observe for example the concentration of estimated target points in an erroneous space (no underlying corresponding source points) in Fig. 7(b) and (d), even in the case of noise free data. The same also applies in case of points corrupted by noise, where CPD and GMMReg provided results that describe the shape of the target set quite well but there are points with no underlying correspondences, e.g. Fig. 10(c) and (e) or Fig. 11(c) and (e). On the other hand, RPM computed a good matching between the registered shapes but a refinement step is needed to achieve perfect registration, e.g. Fig. 7(c). In general, RPM proved too difficult to be tuned, and therefore provided a high rate of failures.

Concerning the CPD and the GMMReg methods, the time complexity of these algorithms are lower than ours which is partially implemented in Matlab (Table 5). The fact that under similar comparison conditions our algorithm may provide similar results is justified by the convergence comparison (Table 6), where one may observe that our technique converges quite faster than CPD and GMMReg. A general conclusion regarding the comparison of our method and CPD/GMMReg is, that, taking into account the registration error, the implementation and parameter tuning (e.g. selecting the type of transformation rigid or non-rigid) along with the time complexity our method may provide better registration results, under the condition that a good assignment solution is provided.

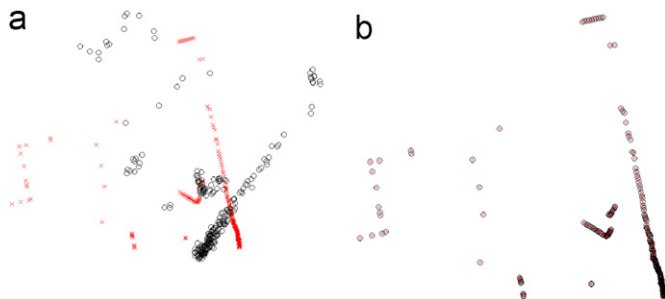
Another issue studied in our experiments is the integration of an annealing scheme, either in the correspondence establishment (step 1 of Algorithm 3) or in the RVM training (step 2 of Algorithm 3). One approach was to embed softassign [4], as solution to the



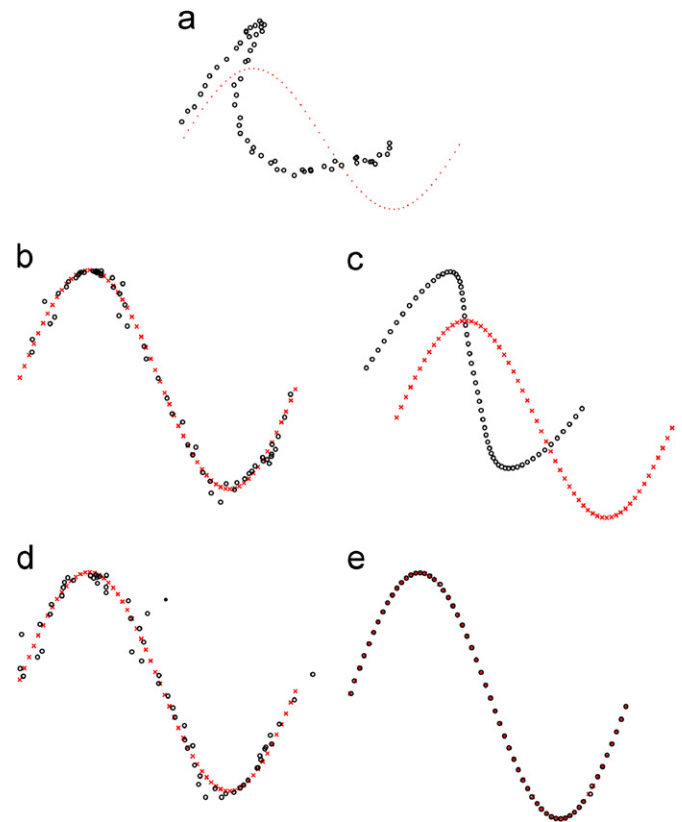
**Fig. 7.** Non-rigid transformation experiment. (a) Reference set of points (red) and deformed set of points (black). Registration result for (b) CPD, (c) RPM, (d) GMMReg and (e) the proposed method. The difference is better highlighted in color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 8.** Non-rigid transformation experiment with 3D points [6]. (a) Reference set of points (red) and deformed set of points (black) of a 3D face. (b) Registration result for the proposed method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 9.** Rigid transformation experiment in presence of noise. (a) Reference 2D range set of points (red) and deformed set of points (black), [22] corrupted with zero mean additive Gaussian noise. (b) Registration result for the proposed method. The difference is better highlighted in color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

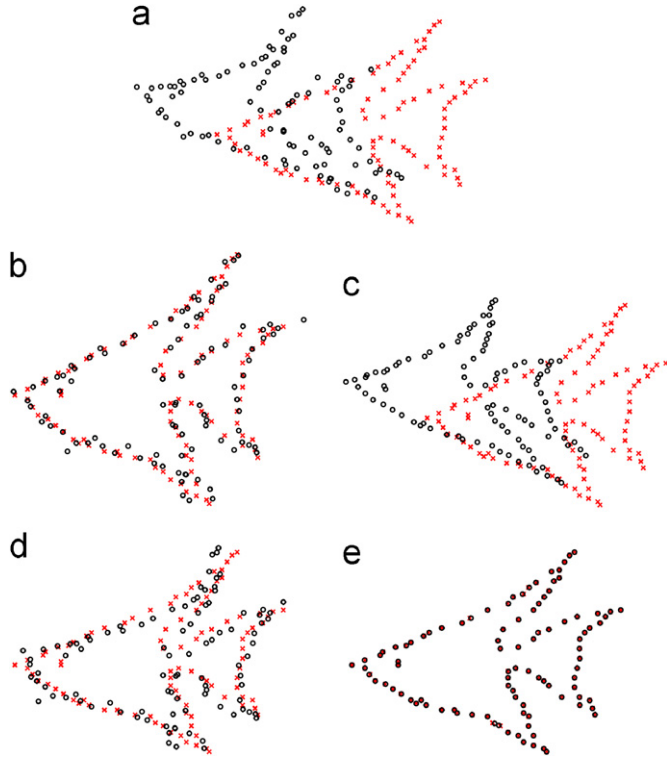


**Fig. 10.** Non-rigid transformation experiment in presence of noise. (a) Reference set of points (red) and deformed set of points (black) corrupted with zero mean additive Gaussian noise. Registration result for (b) CPD, (c) RPM, (d) GMMReg and (e) the proposed method. The difference is better highlighted in color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

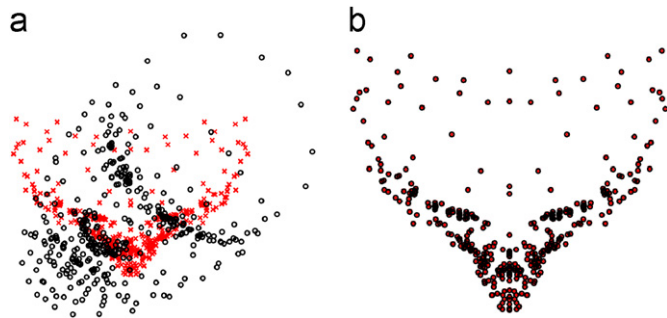
correspondence establishment instead of the Hungarian algorithm. The annealing temperature was initialized to 10% of the maximum pairwise distance between the points. After each iteration, the annealing temperature was reduced to 0.9 of its previous value. The results for the various combinations are presented in Tables 9 and 10. As it can be observed, all the matching variants provide similar accuracy, regarding the mean squared error. However, considering the complexity of the model, direct application of the Hungarian algorithm appeared to be the most efficient approach. A few parameters have to be estimated while the execution time is considerably smaller. A straightforward implementation of the Hungarian algorithm demands less than one third of the softassign execution time. Based on the aforementioned remarks, we prefer the combination of the Hungarian algorithm (for solving the correspondence problem) with RVMs (for estimating the transformation) without any annealing scheme.

More experiments were conducted to investigate the robustness of RVM regression with respect to TPS interpolation in the registration of point sets. For that purpose, we fixed the correspondences between the reference and the target sets in order to contain a number of false matches. Two types of experiments were performed. In the first type, the false matches preserved the one-to-one correspondence, that is, one point of the source set corresponds to exactly one point in the target set (one to one). In the other type of experiments, one point of the source set may correspond to one or more points in the target set (one to many). Then, we applied the transformation (TPS or RVM) and we counted the number of correct alignments. An alignment of two points was considered to be correct if the Euclidean distance





**Fig. 11.** Non-rigid transformation experiment in presence of noise. (a) Reference set of points (red) and deformed set of points (black) corrupted with zero mean additive Gaussian noise. Registration result for (b) CPD, (c) RPM, (d) GMMReg and (e) the proposed method. The difference is better highlighted in color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 12.** Non-rigid transformation experiment in presence of noise. (a) Reference 3D set of a face points (red) and deformed set of points (black), [6] corrupted with zero mean additive Gaussian noise. (b) Registration result for the proposed method. The difference is better highlighted in color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 7**  
Mean registration error for rigid transformations in presence of noise.

Point set	Hungarian-RVM	CPD [6]	RPM [3]	GMMReg [7]
<i>Sine</i>	0.00	0.01	14.62	0.01
<i>Blob</i>	0.00	0.01	11.53	0.01
<i>Fish</i>	0.00	0.01	22.61	0.01
<i>Ideogram</i>	0.00	0.01	23.86	0.01
<i>2D range</i>	0.01	0.51	Fail	0.48
<b>p-Value</b>	–	0.00	$10^{-15}$	0.00

between a transformed point and its corresponding was less than a predefined threshold. By varying the threshold we may plot a curve demonstrating the performance of the compared methods.

**Table 8**  
Mean registration error for non-rigid transformations in presence of noise.

Point set	Hungarian-RVM	CPD [6]	RPM [3]	GMMReg [7]
<i>Sine</i>	0.00	0.01	14.68	0.01
<i>Blob</i>	0.00	0.01	11.47	0.01
<i>Fish</i>	0.00	0.01	22.48	0.01
<i>Ideogram</i>	0.00	0.01	23.72	0.01
<i>3D face</i>	0.00	0.01	Fail	0.00
<b>p-Value</b>	–	$10^{-4}$	0.05	0.03

**Table 9**  
Registration error statistics for non-rigid transformations.

Point set	Hungarian-RVM				
	Mean	Std	Median	Max	Min
<i>Sine</i>	0.00	0.00	0.00	0.00	0.00
<i>Blob</i>	0.00	0.00	0.00	0.00	0.00
<i>Fish</i>	0.00	0.00	0.00	0.00	0.00
<i>Ideogram</i>	0.00	0.00	0.00	0.00	0.00

**Table 10**  
Registration error statistics for non-rigid transformations.

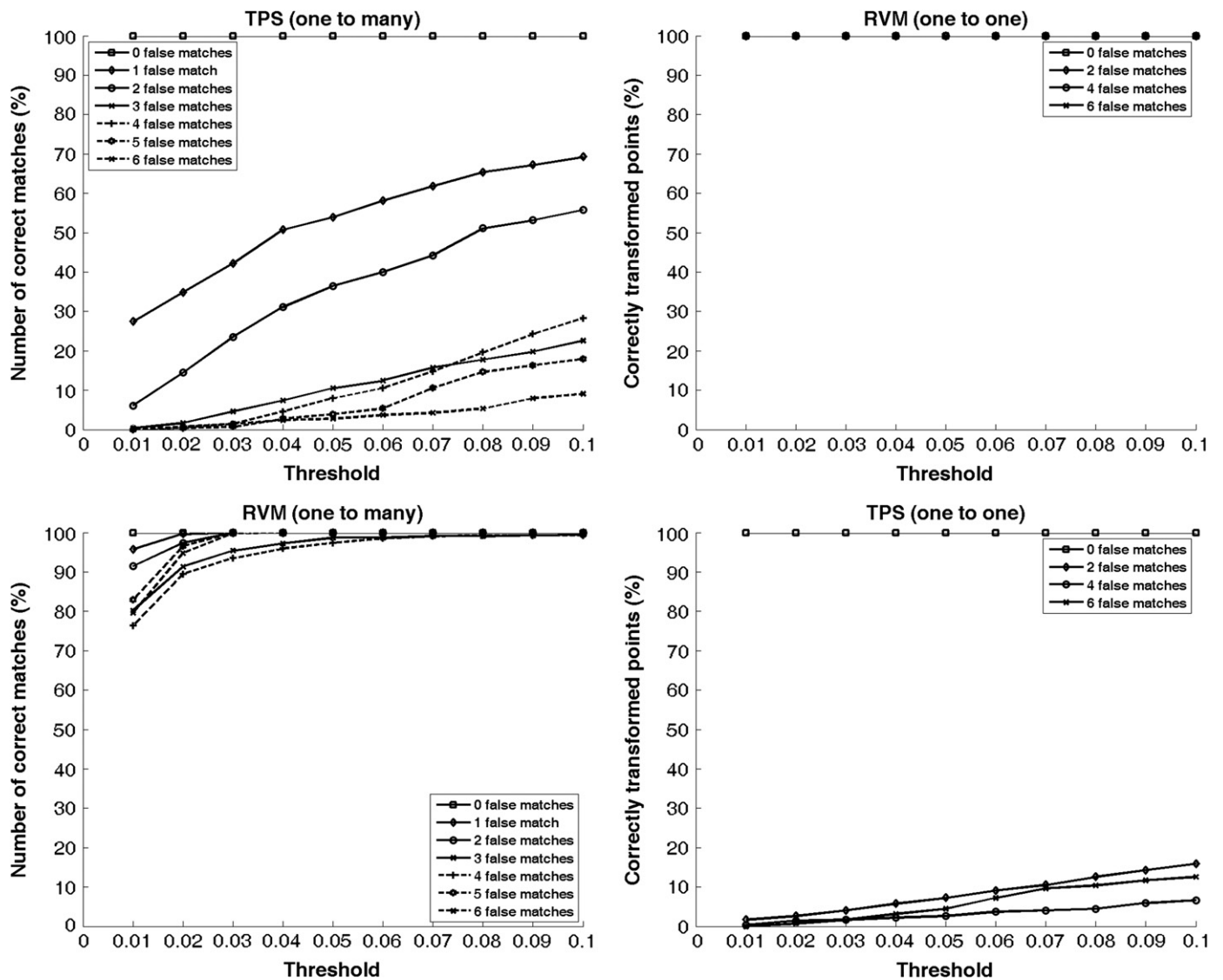
Point set	Softassign-RVM				
	Mean	Std	Median	Max	min
<i>Sine</i>	0.02	0.02	0.02	0.01	0.04
<i>Blob</i>	0.02	0.02	0.02	0.00	0.04
<i>Fish</i>	0.01	0.01	0.01	0.01	0.02
<i>Ideogram</i>	0.01	0.00	0.01	0.00	0.01

These curves are shown in Fig. 13 for various cases of false matches on a set of 60 2D points. The curves correspond to a rigid transformation on the set of Fig. 3(a). The translation parameters were fixed to  $[0.2, 0.3]^T$  and the rotation angle varied in the interval  $[0^\circ, 80^\circ]$  with a step of  $10^\circ$ . The curves in Fig. 13 show the average values between all angles examined per threshold. Notice that the RVM regression always provides an accurate result and justifies our claim that it can model better a registration transformation. In case of one to one correspondence, the target and the transformed sets almost coincide, while in the case of one to many correspondences, the registration result is close to the target set, and the shape is generally preserved. On the other hand, TPS completely fails to model the registration transformation even with few false matches. This behavior is justified by the fact that RVM does not consider the whole set for extracting the regression model. A representative example is also shown in Fig. 1.

In the same spirit, we examined the smoothness of the resulting transformation of RVM with respect to TPS. Following the same procedure, the number of false correspondences was gradually increased and the smoothness of the transformation was computed. We define the smoothness of a transformation  $\mathcal{T}$  that registers set  $X$  to  $Y$  as

$$\mathcal{S}(\mathcal{T}) = \sum_{\substack{\mathbf{x} \in X \\ \mathbf{x}' \in \mathcal{N}(\mathbf{x})}} (d(\mathbf{x}, \mathbf{x}') - d(\mathcal{T}(\mathbf{x}) - \mathcal{T}(\mathbf{x}')))^2, \quad (15)$$

where  $\mathcal{N}(\mathbf{x})$  is the set of nearest neighbors of  $\mathbf{x}$  in  $X$ ,  $d(\mathbf{p}, \mathbf{t}) = \|\mathbf{p} - \mathbf{t}\|$  is the Euclidean distance between points  $\mathbf{p}$  and  $\mathbf{t}$ , while  $\mathcal{T}$  is either the RVM or the TPS transformation. The quantity given by (15) has a high value (indicating non-smoothness) when a point and its neighbors in the source set have counterparts located at distant points in the target set. In other words, if the distance of the points to its neighbors in the source set is relatively different



**Fig. 13.** Curves representing the number of points correctly transformed with respect to a threshold determining the correct transformation using RVM (top row) and TPS (bottom row) when a number of initial false matches is established in source and target sets. A point in the source set is correctly transformed if, after transformation, its distance with respect to its correct counterpart is below the threshold. The left column shows results with false assignments that preserve the one-to-one matching. In that case the RVM provides a consistent behavior and its curves are all at 100% correct transformation. The right column shows results with false assignments that do not preserve the one-to-one matching.

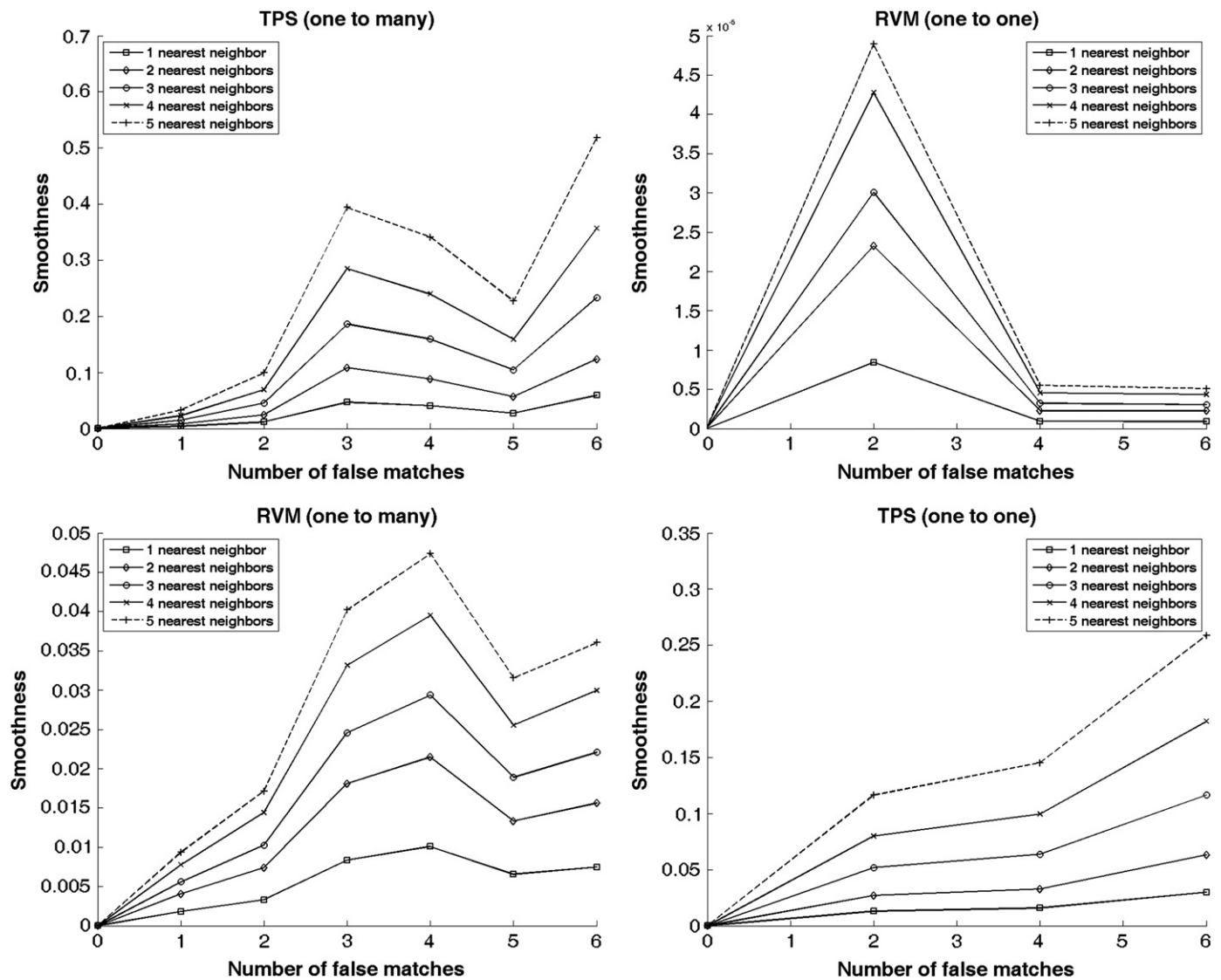
with respect to the distance of their counterparts in the target set a high penalty is added in the smoothness quantity. The curve in Fig. 14 presents the smoothness of the transformation by varying the number of false matches, and the number of neighbors in  $\mathcal{N}(x)$ . Notice that RVM provides a quite smoother transformation although it may result to foldings if the number of false matches is increased.

**4. Conclusion**

In this paper, we introduced a Bayesian regression method to solve the rigid registration problem between two 2D sets. An iterative framework is proposed alternating between correspondence establishment and registration transformation estimation. Correspondences between distinct points are estimated by the Hungarian algorithm which solves the assignment problem for a given cost matrix, determined by the Euclidean distances between points. Then, a regression procedure based on a Bayesian

linear model (Relevance Vector Machine—RVM) maps the two sets of points. The algorithm was successfully evaluated on examples with varying difficulty and favorably compared with state-of-the-art methods both in cases of noise and outliers. The contribution of our work is that it proposes and justifies the claim, that in case a good assignment algorithm is used, that provides few false matches, it is better to embody a regression technique (RVM) to model the registration transformation, instead of using the spline interpolation (TPS), as the result may be more accurate.

Concerning future work, more sophisticated features may be adopted, like the shape context information proposed in [18]. Also, the introduction of multivariate RVMs may be considered. In our method we trained a distinct RVM for each component of the output vector. A multivariate RVM could provide smoother results, as it imposes an implicit constraint to the estimated solution. Finally, the adoption of a set of control points to speed up the computation steps could also be introduced in the method, as the rationale of an RVM is to extract the most salient points that participate in the determination of the final regression



**Fig. 14.** Smoothness (15) of the RVM (top row) and TPS (bottom row) under various number of false matches. The left column shows results with false assignments that preserve the one-to-one matching. Notice that the scale of vertical axis at the top-left plot is  $10^{-5}$  indicating a very smooth transformation. The right column show results with false assignments that do not preserve the one-to-one matching.

model. In that sense, during the correspondence estimation step, only those points could be taken into account to define the assignment cost matrix in order to establish correspondences. This approach not only could reduce the complexity but it could also increase the robustness of the matching result, as it would provide a more sparse neighborhood preventing, thus, false correspondences. Finally, the employment of a twinned RVM, as suggested in [14] could improve the efficiency of the herein proposed method.

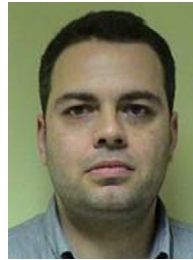
## References

- [1] P.J. Besl, N.D. McKay, A method for registration of 3-D shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (2) (1992) 239–256.
- [2] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, in: *Proceedings of the 3rd International Conference on 3D Digital Imaging and Modeling (3DIM'01)*, 2001.
- [3] H. Chui, A. Rangarajan, A new algorithm for non-rigid point matching, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*, vol. 2, 2000, pp. 40–51.
- [4] S. Gold, C.P. Lu, A. Rangarajan, S. Pappu, E. Mjolsness, New algorithms for 2D and 3D point matching: pose estimation and correspondence, in: G. Tesauro, D. Touretzky, T. Leen (Eds.), *Advances in Neural Information Processing Systems*, vol. 7, The MIT Press, 1995, pp. 957–964.
- [5] F.L. Bookstein, Principal warps: thin-plate splines and the decomposition of deformations, *IEEE Trans. Pattern Anal. Mach. Intell.* 11 (6) (1989) 567–585.
- [6] A. Myronenko, X. Song, Point-set registration: coherent point drift, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (2010) 2262–2275.
- [7] B. Jian, B.C. Vemuri, Robust point set registration using Gaussian mixture models, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (2011) 1633–1645.
- [8] D. Gerogiannis, C. Nikou, A. Likas, The mixtures of Student's *t*-distributions as a robust framework for rigid registration, *Image Vision Comput.* 27 (9) (2009) 1285–1294.
- [9] Z. Tu, S. Zheng, A. Yuille, Shape matching and registration by data-driven EM, *Comput. Vision Image Understanding* 109 (3) (2008) 290–304.
- [10] T. Suk, J. Flusser, Point-based projective invariants, *Pattern Recognition* 33 (2000) 251–261.
- [11] T.S. Caetano, T. Caelli, D. Schuurmans, D.A.C. Barone, Graphical models and point pattern matching, *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (10) (2006) 1646–1663.
- [12] Y. Zheng, D.S. Doermann, Robust point matching for nonrigid shapes by preserving local neighborhood structures, *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (4) (2006) 643–649.
- [13] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, *J. Mach. Learn. Res.* 1 (2001) 211–244.
- [14] L. Tan, C. Fyfe, Canonical correlation analysis—a relevance vector approach, in: *The 9th European Symposium on Artificial Neural Networks, ESANN2001*, April 2001.

- [15] Y. Iwai, R. Cipolla, Multivariate sparse Bayesian regression and its application for facial feature detection, in: Proceedings of the IAPR Conference on Machine Vision Applications (MVA'05), 2005.
- [16] H. Chui, A. Rangarajan, A new point matching algorithm for non-rigid registration, *Comput. Vision Image Understanding* 89 (2003) 114–141.
- [17] C.H. Papadimitriou, K. Steiglitz, *Combinatorial Optimization. Algorithms and Complexity*, Dover, 1998.
- [18] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (4) (2002) 509–522.
- [19] P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via EM algorithm, *J. R. Stat. Soc.* 39 (1) (1977) 1–38.
- [20] F. Bourgeois, J.C. Lassalle, An extension of the Munkres algorithm for the assignment problem to rectangular matrices, *Commun. ACM* 12 (December) (1971) 802–804.
- [21] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [22] Y. Tsin, T. Kanade, A correlation-based approach to robust point set registration, in: Proceedings of the European Conference on Computer Vision, 2004, pp. 558–569.
- [23] Matlab Code for Univariate RVM Implementation <<http://www.miketipping.com>>.



**Demetrios Gerogiannis** received the B.Sc. and M.Sc. from the Department of Computer Science, University of Ioannina, Greece in 2005 and 2007 respectively. Currently he is a PhD student at the same department. His research interests include computer vision, image processing and analysis, machine learning and pattern recognition. Since 2007, he is involved in applied research projects. He is also a Student member of IEEE. Twitter: @dgerog.



**Christophoros Nikou** received the Diploma degree in Electrical Engineering from the Aristotle University of Thessaloniki, Greece, in 1994 and the DEA and PhD degrees in image processing and computer vision from Louis Pasteur University, Strasbourg, France, in 1995 and 1999, respectively. During 2001, he was a Senior Researcher with the Department of Informatics, Aristotle University of Thessaloniki. From 2002 to 2004, he was with Compucon S.A., Thessaloniki. Since 2004, he is with the Department of Computer Science, University of Ioannina, Greece where he was a Lecturer (2004–2009) and he is now an Assistant Professor. His research interests mainly include image processing and computer vision and their application to medical imaging. He is an Associate Editor for EURASIP Journal on Advances in Signal Processing. He is a member of EURASIP and an IEEE Senior Member.



**Aristidis Likas** received the Diploma degree in Electrical Engineering and the PhD degree in Electrical and Computer Engineering, both from the National Technical University of Athens, Greece.

Since 1996, he has been with the Department of Computer Science, University of Ioannina, Greece, where he is currently an Associate Professor. His research interests include machine learning, neural networks, statistical signal processing, and bioinformatics.