This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



# An Efficient Globally Optimal Algorithm for Asymmetric Point Matching

Wei Lian, Lei Zhang, and Ming-Hsuan Yang

**Abstract**—Although the robust point matching algorithm has been demonstrated to be effective for non-rigid registration, there are several issues with the adopted deterministic annealing optimization technique. First, it is not globally optimal and regularization on the spatial transformation is needed for good matching results. Second, it tends to align the mass centers of two point sets. To address these issues, we propose a globally optimal algorithm for the robust point matching problem where each model point has a counterpart in scene set. By eliminating the transformation variables, we show that the original matching problem is reduced to a concave quadratic assignment problem where the objective function has a low rank Hessian matrix. This facilitates the use of large scale global optimization techniques. We propose a branch-and-bound algorithm based on rectangular subdivision where in each iteration, multiple rectangles are used to increase the chances of subdividing the one containing the global optimal solution. In addition, we present an efficient lower bounding scheme which has a linear assignment formulation and can be efficiently solved. Extensive experiments on synthetic and real datasets demonstrate the proposed algorithm performs favorably against the state-of-the-art methods in terms of robustness to outliers, matching accuracy, and run-time.

**Index Terms**—branch and bound, concave optimization, linear assignment, point correspondence, robust point matching

## 1 Introduction

The goal of point matching is to find correspondence between two point sets and recover the underlying transformation. It plays an indispensable role for numerous applications in computer vision and pattern recognition. Accurate point matching is challenging as matching algorithms need to account for non-rigid deformation, positional noise, outliers from images. Numerous methods have been proposed to address the point matching problem [5], [45], among which the robust point matching (RPM) method [9], [10] is widely used for non-rigid registration, and several variants [34], [39], [19] have been developed subsequently.

The RPM method jointly estimates correspondence and transformation by formulating point matching as a mixed linear assignment least squares optimization problem. Based on this formulation, it relaxes the point correspondence variables to be continuous valued subject to the one-to-one correspondence constraint (i.e., soft assignment) and employs a deterministic annealing technique [42] to gradually estimate point correspondence. However, a deterministic annealing method involves heuristics and there is no global optimality guarantee. Consequently, the spatial transformation variable needs to be regularized to avoid undesirable matching results. Furthermore, the deterministic annealing technique initially matches points in two sets with equal chance which causes the mass centers of the sets to be aligned. This trend may persist and result in aligning the centers of two point sets.

To address the issues with the use of deterministic annealing for non-rigid registration, we propose an asymmetric point matching (APM) algorithm for the RPM problem where each model point has a counterpart in scene set. The APM algorithm operates by eliminating the transformation variables of the RPM objective function to reduce it to a function of point correspondence with the following properties.

First, the function is concave. Taking into account the total unimodularity of the constraints on point correspondence [33], [18], we show that the optimal solution by the proposed algorithm is directly binary valued. Thus, there is no need for discretizing solutions which can cause errors and inaccurate estimates [20], [22], [27].

Second, the function is quadratic with a Hessian matrix whose rank equals the number of transformation parameters. For the case when transformation has few parameters (e.g., 2D similarity or 2D/3D affine transformation), the rank is low and thus large scale global optimization techniques can be applied [18]. Consequently, the proposed algorithm is globally optimal and scalable.

To optimize the proposed objective function, we propose a branch-and-bound (BnB) algorithm based on rectangular subdivision [18], where in each iteration, instead of only subdividing the rectangle yielding the lower bound for the feasible region, more rectangles are used to increase the chances of subdividing the one containing the global optimal solution and thereby improve the convergence speed. Experimental results


- *W. Lian is with the Department of Computer Science, Changzhi University, Changzhi, Shanxi, China, 046011, and is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail:* lianwei3@foxmail.com.
- *L. Zhang is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail:* cslzhang@comp.polyu.edu.hk.
- *M.-H. Yang is with School of Engineering, University of California, Merced, CA 95344. E-mail:* mhyang@ucmerced.edu.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



show the proposed algorithm achieves 10-fold speed-up for large size problems.

In order to further improve the efficiency of the proposed BnB algorithm where computation of the lower bound is the bottleneck, we propose a lower bounding scheme which has a linear assignment formulation and can be efficiently solved. Experimental results show that 500-fold speed-up can be obtained by the proposed scheme. In addition, the proposed formulation yields tight lower bounds.

The contributions of this work are summarized as follows. First, the proposed algorithm can match point sets when transformations are not regularized. In these cases, the proposed algorithm is invariant to the corresponding transformations and match two point sets with unknown poses. In contrast, it is necessary to use regularization techniques for existing RPM methods to avoid undesirable matching results. Second, as the proposed algorithm computes globally optimal solutions, it is robust to outliers or extraneous structures than existing methods that find local solutions. Validated by the experiments, the proposed algorithm is more robust than the state-of-the-art RPM methods. Third, the proposed algorithm performs robustly against the RPM method for non-rigid registration. Experimental results show that the RPM method needs to use the thin plate spline transformation to perform as well as the proposed algorithm with affine transformation.

## 2 Related Work and Problem Context

Thorough reviews on point matching and registration can be found in [5], [45]. In this section, we discuss the most relevant algorithms on non-rigid point matching and put this work in proper context.

**Point Matching:** Numerous methods on modeling both point correspondence and spatial transformation have been proposed. The iterative closest point (ICP) methods [3], [43] determine the nearest neighbors for point correspondence and from the spatial transformation. Due to the discrete nature of point correspondence, it is well known that ICP is prone to be trapped into local minimum. The RPM methods [9], [10] pose the task as a mixed linear assignment least square problem in which the soft assignment and deterministic annealing techniques are used for optimization. However, the RPM methods tend to align the mass centers of two point sets. As such, the covariance matrix of the alignment error is used for point matching in the presence of missing or extraneous structures [39]. Since the size of the covariance matrix is square times the number of transformation parameters, this method is efficient only when transformation has few parameters. In addition, the robust $L_2E$ estimator [32] is proposed for point matching based on assumptions that the noise on the inliers is a Gaussian with zero mean and uniform standard deviation. As the above methods are developed based on stronger assumptions (e.g., number of transformation parameters and noise model), they may not perform well for large datasets with significant amount of outliers.

Another category of point matching methods model only spatial transformation. These methods are developed mainly based on distribution models of point sets. The coherent point drift (CPD) method [34] formulates point matching as fitting a Gaussian Mixture Model (GMM) representing one point set to the other one. Without directly finding point correspondence, Glaunes *et al.* [14] formulate point matching as aligning two weighted sum of Dirac measures representing two point sets. Instead of using Dirac measures, GMMs are used to represent point sets [19] and the $L_2$ distance between them is minimized for matching. The kernel correlation point matching method [40] can be seen as a special case of the Gaussian mixture based approach [19] and its variant based on the log-exponential function [4]. Instead of directly aligning two distributions representing two point sets, Ho *et al.* [17] propose to match the moments of distributions. This results in a system of polynomial equations which can be solved by algebraic geometric techniques. However, this method is sensitive to occlusions and outliers due to the use of moments. The Schrödinger distance transformation is used to represent point sets [11] and the point set registration problem is converted into the task of computing the geodesic distance between two points on a unit Hilbert sphere. A common problem with the above methods is that the one-to-one correspondence constraint is not enforced. Hence, these methods tend to generate inaccurate matching results for complex point sets.

Numerous methods have been developed for modeling only point correspondence as a graph matching problem. Both integer programming [25] and random walk [7] methods have been developed for graph matching. In addition, max pooling [8] is used to deal with outliers. As only pairwise affinities between correspondences, these method are less effective in coping with large non-rigid deformations such as scaling and shear. To address this problem, tensor is used to encode affinities between tuples of correspondences for invariance to transformations such as similarity transformation [12]. However, the tensor power method does not handle outliers well. Consequently, the tensor block coordinate ascend method [35] is extended to deal with outliers at the expense of large memory load. On the other hand, Zheng *et al.* use the relaxation labeling techniques to preserve local neighborhood structures of graphs [44], [24]. Nevertheless, these methods need to be initialized by using the shape context features [2].

Methods have been developed to model point correspondence as a Markov random field (MRF) problem. Graphs with special topology suitable for optimization by dynamic programming are designed to achieve invariance to various types of transformations in [6]. In [20], local lengths and orientations of shapes are preserved via linear programming. This method is extended to be invariant to similarity transformation [22],

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



[21] by explicitly modeling rotation and scaling between two point sets, and affine invariant [27] based on the local linear embedding algorithm [38]. To address the problem of rotation invariance for shape matching in cluttered scenes, fan-shaped graphs are introduced [28], [31] where the elongated edges are used to determine the orientations of point sets.

**Concave Optimization:** For feature matching, Maciel and Costeira [33] convert a correspondence problem to an equivalent concave optimization task. However, it is difficult to solve the concave problem due to lack of good structures for optimization. In contrast, the proposed formulation enforces low rank properties, thereby accommodating large scale global optimization techniques.

**Branch-and-bound:** BnB methods have been widely used in vision and learning problems for global optimization. While it has been used to recover 3D rigid transformation [36], the point-to-point, point-to-line or point-to-plane correspondence needs to be known a priori, which limits the application domain of this method. In [37], BnB technique is used to optimize the RPM objective function where branching over the correspondence variables and branching over the transformation variables are both considered. However, this method is developed for small 2D datasets and the extension to 3D or higher problems are not trivial due to higher dimensionality of the search space. Recently, BnB techniques are used to optimize the ICP objective function [41] by exploiting the special structure of the geometry of 3D rigid motions. However, this method can only be used to recover rotation as well as translation from point sets (rather than scale).

For robust geometry estimation, a BnB algorithm [26] for consensus set maximization is developed to determine the best transformation by maximizing the number of inliers. In addition, this method has been applied to the correspondence and grouping problems [1]. However, it is computationally expensive to solve the resulting bounding problems due to lack of efficient optimization techniques. In contrast, the proposed algorithm is efficient partly due to the linear assignment formulation of the lower bounding problem which can be efficiently solved.

**Problem Context:** Early results of this work are reported in [29] where a concave quadratic objective function for point correspondence is proposed and solved with a BnB method. In this work, we propose a branch-and-bound algorithm and a fast lower bounding scheme, which together significantly increase the efficiency of the previous work, analyze the developed BnB algorithm, discuss the impact of the regularization term on convergence, and present extensive experimental results as well as discussions. An extension of [29] to deal with outliers in both point sets is proposed in [30]. However, the objective function of [30] is not separable and thus the search space is larger than that of the proposed algorithm in this work. In addition, the lower bounding function of [30] can only be constructed for simplexes instead of rectangles, which results in looser bound and slower convergence than that of this work.

## 3 Energy Function of RPM

Since our energy function originates from the energy function of RPM [10], we briefly review the formulation of RPM in this section. Suppose there are two point sets in $\mathbb{R}^d$ to be matched, the model set $\mathscr{X}=\{\mathbf{x}_i, i=1,\dots,n_x\}$ with point $\mathbf{x}_i=\left[x_i^1,\dots,x_i^d\right]^\top$, and the scene set $\mathscr{Y}=\{\mathbf{y}_j, j=1,\dots,n_y\}$ with point $\mathbf{y}_j=\left[y_j^1,\dots,y_j^d\right]^\top$. To solve this problem, the RPM method jointly estimates transformation parameters and point correspondence by formulating point matching as a mixed linear assignment least square problem [10]:

$$\min \; E(P,\boldsymbol{\theta}) = \sum_{i,j} p_{ij}\|\mathbf{y}_j - T(\mathbf{x}_i|\boldsymbol{\theta})\|_2^2 + g(\boldsymbol{\theta}) \tag{1}$$

$$\text{s.t.} \quad P\mathbf{1}_{n_y} \le \mathbf{1}_{n_x}, \quad \mathbf{1}_{n_x}^\top P \le \mathbf{1}_{n_y}^\top, \quad P \ge 0 \tag{2}$$

Here $\|\cdot\|_2$ denotes the $l_2$-norm of a vector. $P=\{p_{ij}\}$ denotes a correspondence matrix with $p_{ij}=1$ if there is a match between $\mathbf{x}_i$ and $\mathbf{y}_j$ and 0 otherwise. In addition, $\mathbf{1}_{n_x}$ denotes an $n_x$-dimensional vector of all ones, $T(\cdot|\boldsymbol{\theta})$ describes a spatial transformation with a parameter vector $\boldsymbol{\theta}$, and $g(\boldsymbol{\theta})$ is a regularizer on $\boldsymbol{\theta}$. To solve problem (1), (2), the RPM method relaxes $P$ to be fuzzily valued and uses deterministic annealing for optimization. However, the RPM method is less robust to outliers due to the adopted heuristic minimization scheme. Based on (1), in the next section we present a new energy function which is more amenable to global optimization.

## 4 Proposed Energy Function

In this section, we derive a new energy function based on the energy function of RPM. To make this problem tractable, we assume that the spatial transformation $T(\mathbf{x}_i|\boldsymbol{\theta})$ is linear with respect to its parameters $\boldsymbol{\theta}$, i.e., $T(\mathbf{x}_i|\boldsymbol{\theta}) = J(\mathbf{x}_i)\boldsymbol{\theta}$, where $J(\mathbf{x}_i)$ is the Jacobian matrix. Examples of the Jacobian matrix include (19), (21) and (23) for 2D and 3D point matching described in Section 6. We consider the following regularization form on $\boldsymbol{\theta}$: $g(\boldsymbol{\theta}) = (\boldsymbol{\theta}-\boldsymbol{\theta}_0)^\top H(\boldsymbol{\theta}-\boldsymbol{\theta}_0) - \boldsymbol{\theta}_0^\top H\boldsymbol{\theta}_0 = \boldsymbol{\theta}^\top H\boldsymbol{\theta} - 2\boldsymbol{\theta}_0^\top H\boldsymbol{\theta}$, i.e., $\boldsymbol{\theta}$ is required to be close to a constant vector $\boldsymbol{\theta}_0$. Here $H$ is a positive semidefinite matrix whose entries represent the weights assigned to the elements of $\boldsymbol{\theta}$.

With the above assumptions, the energy function (1) takes the following form:

$$\begin{aligned} E(P,\boldsymbol{\theta}) &= \sum_{i,j} p_{ij}\|\mathbf{y}_j - J(\mathbf{x}_i)\boldsymbol{\theta}\|_2^2 + \boldsymbol{\theta}^\top H\boldsymbol{\theta} - 2\boldsymbol{\theta}_0^\top H\boldsymbol{\theta} \\ &= \boldsymbol{\theta}^\top[J^\top(\mathrm{diag}(P\mathbf{1}_{n_y})\otimes I_d)J + H]\boldsymbol{\theta} + \mathbf{1}_{n_x}^\top P\mathbf{z} \\ &\quad - 2\boldsymbol{\theta}^\top[J^\top(P\otimes I_d)\mathbf{y} + H\boldsymbol{\theta}_0] \end{aligned} \tag{3}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



where $J \triangleq \left[J^\top(\mathbf{x}_1), \ldots, J^\top(\mathbf{x}_{n_x})\right]^\top$, $\mathbf{y} \triangleq \left[\mathbf{y}_1^\top, \ldots, \mathbf{y}_{n_y}^\top\right]^\top$ and $\mathbf{z} \triangleq \left[\|\mathbf{y}_1\|_2^2, \ldots, \|\mathbf{y}_{n_y}\|_2^2\right]^\top$. In addition, the function diag$(\cdot)$ converts a vector into a diagonal matrix, $I_d$ denotes the $d$-dimensional identity matrix, and $\otimes$ is the Kronecker product.

It is difficult to optimize (3) as the first term is a cubic polynomial in $P$ and $\boldsymbol{\theta}$. However, adding the constraint $P\mathbf{1}_{n_y} = \mathbf{1}_{n_x}$ simplifies it to a quadratic polynomial in $P$ and $\boldsymbol{\theta}$, which is easier to optimize.

Therefore, we add the assumption $P\mathbf{1}_{n_y} = \mathbf{1}_{n_x}$ to our formulation. This constraint has a natural physical meaning: each model point should have a counterpart in the scene set, or namely the model shape should not be occluded. This constraint is also widely adopted in the literature, especially by methods for detecting objects in a scene [20], [31].

By enforcing the constraint $P\mathbf{1}_{n_y} = \mathbf{1}_{n_x}$ in (2), we have:

$$P\mathbf{1}_{n_y} = \mathbf{1}_{n_x}, \quad \mathbf{1}_{n_x}^\top P \leq \mathbf{1}_{n_y}^\top, \quad P \geq 0 \tag{4}$$

The constraints in (4) satisfy the total unimodularity property [33], which means that the vertices of the polytope (i.e., bounded polyhedron) determined by (4) have integer valued coordinates. By taking into account $0 \leq p_{ij} \leq 1$, the coordinates of the vertices are actually binary valued as 0 or 1.

With the constraint $P\mathbf{1}_{n_y} = \mathbf{1}_{n_x}$, the energy function $E$ becomes quadratic:

$$\begin{aligned} E(P, \boldsymbol{\theta}) =& \boldsymbol{\theta}^\top (J^\top J + H)\boldsymbol{\theta} - 2\boldsymbol{\theta}^\top [J^\top (P \otimes I_d)\mathbf{y} + H\boldsymbol{\theta}_0] \\ &+ \mathbf{1}_{n_x}^\top P\mathbf{z} \end{aligned} \tag{5}$$

Given the correspondence matrix $P$, $E$ is a convex quadratic function of $\boldsymbol{\theta}$. Therefore, the minimum solution $\widehat{\boldsymbol{\theta}}$ of $E$ can be obtained in closed form by letting $\frac{\partial E}{\partial \boldsymbol{\theta}} = 0$, i.e.,

$$\widehat{\boldsymbol{\theta}} = (J^\top J + H)^{-1} \left[J^\top (P \otimes I_d)\mathbf{y} + H\boldsymbol{\theta}_0\right] \tag{6}$$

By substituting $\widehat{\boldsymbol{\theta}}$ back into (5), the variable $\boldsymbol{\theta}$ is eliminated and we arrive at an energy function with only one variable $P$,

$$\begin{aligned} E(P) =& -\mathbf{y}^\top (P \otimes I_d)^\top J (J^\top J + H)^{-1} J^\top (P \otimes I_d)\mathbf{y} \\ &- 2\boldsymbol{\theta}_0^\top H (J^\top J + H)^{-1} J^\top (P \otimes I_d)\mathbf{y} + \mathbf{1}_{n_x}^\top P\mathbf{z} \end{aligned} \tag{7}$$

The form of $E$ is cluttered. To change it into a neat form, we compute the Cholesky factorization [15] of the positive definite matrix $(J^\top J + H)^{-1}$ as

$$U^\top U = (J^\top J + H)^{-1} \tag{8}$$

where $U$ is an upper triangular matrix. By substituting (8) into (7), we obtain

$$\begin{aligned} E(P) =& -\|UJ^\top (P \otimes I_d)\mathbf{y}\|_2^2 \\ &- 2\boldsymbol{\theta}_0^\top H (J^\top J + H)^{-1} J^\top (P \otimes I_d)\mathbf{y} + \mathbf{1}_{n_x}^\top P\mathbf{z} \end{aligned} \tag{9}$$

Clearly $E$ is a concave quadratic function of $P$ and we have the following proposition.

*Proposition 1:* There exists a binary solution for any local minimum (including the global minimum) of function $E(P)$ under the constraints in (4).

*Proof:* We already show that $E$ is concave. It is well known that any local minimum (including the global minimum) of a concave function over a polytope can be attained at one of its vertices. The proposition follows by combining this result with the total unimodularity of constraints in (4) as stated previously. □

This proposition implies that minimization of our objective function by simplex-like algorithms results in a binary valued solution. This is of great importance as it avoids the need of discretizing solutions which can cause errors and lead to inaccurate estimates [20], [22], [27].

For convenience of discussion in the following, the matrix $P$ needs first to be vectorized. Contrary to the convention, we define the vectorization of a matrix as the concatenation of its rows instead of its columns:

$$\text{vec}(P) = [p_{11}, \ldots, p_{1n_y}, \ldots, p_{n_x 1}, \ldots, p_{n_x n_y}]^\top$$

Let $\mathbf{p} = \text{vec}(P)$. Based on the formula $\text{vec}(B_1 B_2 B_3) = (B_1 \otimes B_3^\top)\text{vec}(B_2)$ for any matrices $B_1$, $B_2$ and $B_3$, $E$ can be expressed in terms of vector $\mathbf{p}$ as:

$$\begin{aligned} E(\mathbf{p}) =& -\| \left[(UJ^\top) \otimes \mathbf{y}^\top\right] W\mathbf{p}\|_2^2 \\ &+ \left\{\mathbf{1}_{n_x}^\top \otimes \mathbf{z}^\top - 2\left[\left(\boldsymbol{\theta}_0^\top H (J^\top J + H)^{-1} J^\top\right) \otimes \mathbf{y}^\top\right] W\right\}\mathbf{p} \end{aligned} \tag{10}$$

Here the $n_x n_y d^2 \times n_x n_y$ matrix

$$W \triangleq I_{n_x} \otimes \left[I_{n_y} \otimes (\mathbf{e}_d^1)^\top, \ldots, I_{n_y} \otimes (\mathbf{e}_d^d)^\top\right]^\top$$

satisfies $\text{vec}(P \otimes I_d) = W\text{vec}(P)$, where the $d$-dimensional vector $\mathbf{e}_d^i$ has a single unity element at coordinate $i$. In addition, $W$ is a large but sparse matrix and can be implemented using the function `speye` in Matlab.

Define the $\mathrm{n_\theta \times n_x n_y}$ matrix $A = \left[(UJ^\top) \otimes \mathbf{y}^\top\right] W$ where $n_\theta$ denotes the dimension of $\boldsymbol{\theta}$, and the $n_x n_y$-vector $\mathbf{b} = \left\{\mathbf{1}_{n_x}^\top \otimes \mathbf{z}^\top - 2\left[\left(\boldsymbol{\theta}_0^\top H (J^\top J + H)^{-1} J^\top\right) \otimes \mathbf{y}^\top\right] W\right\}^\top$, then the above energy function can be written in a concise form as:

$$E(\mathbf{p}) = -\|A\mathbf{p}\|_2^2 + \mathbf{b}^\top \mathbf{p} \tag{11}$$

A key feature with this function is that its Hessian matrix $-2A^\top A$ has the same rank as that of $A$ which in turn has a rank equal to $n_\theta$. For transformations such as 2D similarity and 2D/3D affine transformations, the rank is low. This is the key feature for the proposed algorithm to be applicable to large scale problems. In the next section, we present an efficient and large scale global optimization algorithm to minimize this function.

## 5 Optimization

It is well known that a quadratic function can be reduced to a separable form by a linear transformation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



of its variables. In this work, the normal rectangular algorithm [18], a BnB procedure specifically tailored to separable functions, is used to minimize the energy function (11).

We first reduce (11) to a separable form via eigen decomposition, and derive the convex envelope (i.e., the tightest convex underestimator) of the resulting function taken over a rectangle (i.e., a set of type $\{t_i | r_i \leq t_i \leq s_i, i = 1 \ldots, l\}$). We next present the bisection scheme to divide a rectangle and propose a variant of the normal rectangle algorithm [18] for minimizing our energy function. To address the complexity issue associated with the above algorithm where computation of the lower bound is the bottleneck, we formulate a new lower bounding problem that can be solved more efficiently. In addition, we discuss the choice of tolerance error and analyze the effect of the regularization term on convergence.

### 5.1 Separable Form of the Energy Function

A quadratic function can be converted into a separable form by a linear transformation of its variables, and the choice of the transformation is based on the eigen decomposition of the quadratic part of the function. The quadratic part of (11) is $-\mathbf{p}^\top A^\top A \mathbf{p}$. Since the dimensionality of $A^\top A$ is $n_x n_y \times n_x n_y$, which is high in reality, directly applying eigen decomposition to $A^\top A$ is impractical. Instead, we present an efficient way of finding the nonzero eigenvalues and eigenvectors of $A^\top A$. Let the QR decomposition of $A^\top$ be:

$$QR = A^\top$$

where $R$ is an $n_\theta \times n_\theta$ upper triangular matrix and the columns of the $n_x n_y \times n_\theta$ matrix $Q$ are orthogonal unity vectors, i.e., $Q^\top Q = I_{n_\theta}$. Thus, we have

$$A^\top A = QRR^\top Q^\top$$

Denote the eigenvalues and eigenvectors of $RR^\top$ as $\lambda_i$ and $\mathbf{u}_i^R$, $i = 1, \ldots, n_\theta$, respectively. The nonzero eigenvalues and eigenvectors of $A^\top A$ are $\lambda_i$ and $\mathbf{u}_i = Q\mathbf{u}_i^R$, $i = 1, \cdots, n_\theta$, respectively. Thus, we have the separable form of the quadratic part of (11) as:

$$-\mathbf{p}^\top A^\top A \mathbf{p} = -\sum_{i=1}^{n_\theta} \lambda_i (\mathbf{u}_i^\top \mathbf{p})^2 \tag{12}$$

### 5.2 Convex Envelope of the Energy Function

With (12), the energy function $E$ has the following separable form:

$$E(\mathbf{p}) = -\sum_{i=1}^{n_\theta} \lambda_i (\mathbf{u}_i^\top \mathbf{p})^2 + \mathbf{b}^\top \mathbf{p} \tag{13}$$

For a separable function over a rectangle, its convex envelope can be readily obtained based on the following proposition [18].

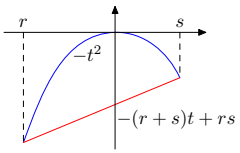



Fig. 1. The convex envelope $-(r+s)t+rs$ of the function $-t^2$ taken over an interval $[r, s]$.

*Proposition 2:* The convex envelope of a separable function $\sum_{i=1}^{l} f_i(t_i)$ taken over a rectangle $M = \{t_i | r_i \leq t_i \leq s_i, i = 1 \ldots, l\}$ is equal to the sum of the convex envelopes of its components $f_i(t_i)$ taken over the corresponding intervals $[r_i, s_i], i = 1, \ldots, l$.

The convex envelope of $f(t) = -t^2$ taken over an interval $[r, s]$ is an affine function that agrees with $f$ at the endpoints of this interval: $f_M(t) = -(r+s)t + rs$, as illustrated in Fig. 1. Based on this fact and the above proposition, we have the convex envelope of function (13) taken over a rectangle $M = \{\mathbf{p} | r_i \leq \mathbf{u}_i^\top \mathbf{p} \leq s_i, i = 1, \ldots, n_\theta\}$ as

$$E_M(\mathbf{p}) = -\sum_{i=1}^{n_\theta} \lambda_i (r_i + s_i) \mathbf{u}_i^\top \mathbf{p} + \sum_{i=1}^{n_\theta} \lambda_i r_i s_i + \mathbf{b}^\top \mathbf{p} \tag{14}$$

This function is used for computing the lower bounds in the proposed BnB algorithm.

### 5.3 Bisection of a Rectangle

In the branching phase of our BnB algorithm, a rectangle $M = \{\mathbf{p} | r_i \leq \mathbf{u}_i^\top \mathbf{p} \leq s_i, i = 1, \ldots, n_\theta\}$ is divided into two subrectangles. Toward that, we need to decide the dimension along which to divide the rectangle, and along a chosen dimension where to divide the rectangle. In this work, we use the bisection scheme [18] to address these issues because of its simplicity and effectiveness. For bisection, the second issue is addressed by choosing the midpoint as the dividing location, and the first issue is addressed based on the following proposition [18].

*Proposition 3:* For $r \leq t \leq s$, the difference between $f(t) = -t^2$ and its convex envelope $f_M(t) = -(r+s)t + rs$ satisfies:

$$\max\{f(t) - f_M(t), r \leq t \leq s\} = \frac{1}{4}(s-r)^2$$

*Proof:* It follows that

$$f(t) - f_M(t) = -t^2 + (r+s)t - rs = (t-r)(s-t)$$

Since the sum of $t-r$ and $s-t$ is a constant, their product is maximum when they equal, and the maximum product is $\frac{1}{4}(s-r)^2$. □

Based on this result, the dimension along which to subdivide a rectangle is chosen as $j \in \arg\max_i \frac{1}{4}\lambda_i (s_i - r_i)^2$ in order to guarantee the convergence of our BnB algorithm [18]. Given the optimal splitting dimension $j$, two subrectangles are generated from this bisection scheme: $M_1 = \{\mathbf{p} \in M | \mathbf{u}_j^\top \mathbf{p} \leq \frac{1}{2}(r_j + s_j)\}$ and $M_2 = \{\mathbf{p} \in M | \mathbf{u}_j^\top \mathbf{p} \geq \frac{1}{2}(r_j + s_j)\}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



### 5.4 Normal Rectangular Subdivision

We propose a branch-and-bound algorithm based on normal rectangular subdivision [18] to find the global $\epsilon$-minimum solution of $E$ (i.e., a solution with function value no larger than $\epsilon$ from the global minimum of $E$). During initialization, the bounding rectangle (i.e., the smallest rectangle containing the feasible region $\Omega$) is computed (step 3 in Algorithm 1). In each iteration of the algorithm, multiple rectangles yielding the lowest lower bounds among all the rectangles are subdivided to improve the lower bound of $E$ for $\Omega$ (steps 9 to 10 in Algorithm 1). Meanwhile, the upper bound is updated by evaluating $E$ with solutions of the linear programs used to compute the lower bounds (step 6 in Algorithm 1). The main steps of the algorithm are summarized in Algorithm 1.

Different from the standard normal rectangular algorithm where in each iteration only the rectangle yielding the lower bound for $\Omega$ is subdivided, in this work more rectangles are subdivided (step 9 in Algorithm 1 where we choose $n_2 = 2^{n_1}$ in this paper) since the globally optimal solution may not lie in the rectangle yielding the lower bound for $\Omega$ and using more rectangles can increase the chances of subdividing the rectangle containing the globally optimal solution, thereby improving the convergence speed. Step 4 of Algorithm 1 ensures that there are sufficient number of rectangles for subdivision in step 9. Note that problem (16) can be solved independently for rectangles in $\mathscr{N}_k$, and thus it can be implemented to run in parallel. This can further speed up the proposed algorithm. However for fair comparisons with other methods, we do not pursue this route in the present paper. Also note that the problems in (15) are linear assignment problems which can be efficiently solved by the Jonker-Volgenant algorithm [23].

We show that the lower bound of our algorithm is always improved in the iterations in the following proposition.

*Proposition 4:* The lower bound of $E$ for the feasible region $\Omega$ is monotonically non-decreasing with the increase of iterations.

*Proof:* The lower bound of $E$ for $\Omega$ at iteration $k$ is chosen as the lowest of all the lower bounds corresponding to the rectangles in $\mathscr{M}_k$. Assume $M_k$ is the rectangle yielding this bound. At iteration $k+1$, there are three cases for the rectangle yielding the lower bound for $\Omega$: 1) one of the two rectangles $M_{ki}$, $i \in \{1,2\}$ from the subdivision of $M_k$; 2) one of the two rectangles $M_{li}$, $i \in \{1,2\}$ from the subdivision of $M_l$ which does not yield the lower bound for $\Omega$ but is divided at iteration $k$; and 3) rectangle $M_l'$ which is not divided at iteration $k$.

If the first case holds, since $M_{ki} \subset M_k$, we have $E_{M_{ki}}(\mathbf{p}) \geq E_{M_k}(\mathbf{p})$ for $\mathbf{p} \in M_{ki}$ because of the concavity of $E$, as illustrated in Fig. 2. Consequently,

$$\min_{\mathbf{p}\in\Omega\cap M_{ki}} E_{M_{ki}}(\mathbf{p}) \geq \min_{\mathbf{p}\in\Omega\cap M_{ki}} E_{M_k}(\mathbf{p}) \quad \geq \min_{\mathbf{p}\in\Omega\cap M_k} E_{M_k}(\mathbf{p})$$

where the second inequality holds since $M_{ki} \subset M_k$. Thus, we have $\beta(M_{ki}) \geq \beta(M_k)$.

If the second case is true, we have $\beta(M_{li}) \geq \beta(M_l) \geq \beta(M_k)$, where the first inequality holds as $\beta(M_{ki}) \geq \beta(M_k)$ in case 1, and the second inequality holds since $M_k$ is the lower bound for $\Omega$ at iteration $k$ according to the assumption.

**Algorithm 1:** Normal rectangular algorithm for minimizing $E$

1 **Initialization**

2 Select tolerance error $\epsilon > 0$.

3 Solve the $2n_\theta$ linear programs

$$\min_{\mathbf{p}\in\Omega} \mathbf{u}_i^\top \mathbf{p}, \quad \max_{\mathbf{p}\in\Omega} \mathbf{u}_i^\top \mathbf{p} \tag{15}$$

to obtain the basic optimal solutions $\mathbf{p}^{0i}$, $\bar{\mathbf{p}}^{0i}$ and the optimal values $\eta_i$, $\bar{\eta}_i$. Clearly, $\Omega \subset M_0 = \{\mathbf{p} | \eta_i \leq \mathbf{u}_i^\top \mathbf{p} \leq \bar{\eta}_i, i = 1, \ldots, n_\theta\}$. Let $\mathbf{p}^0 = \arg\min\{E(\mathbf{p}^{0i}), E(\bar{\mathbf{p}}^{0i}), i = 1, \ldots, n_\theta\}$.

4 Recursively divide $M_0$ into $2^{n_1}$ subrectangles $M_{0i}$, $i = 1, \ldots, 2^{n_1}$ based on the bisection rule stated in Section 5.3. Let $\mathscr{M}_1 = \mathscr{N}_1 = \{M_{0i}, i = 1, \ldots, 2^{n_1}\}$, where $\mathscr{M}_k$ and $\mathscr{N}_k$ denote the collection of all rectangles and the collection of active rectangles at iteration $k$, respectively.

5 **for** $k = 1, 2, \ldots$ **do**

6 For each rectangle $M \in \mathscr{N}_k$, construct the convex envelope $E_M(\mathbf{p})$ according to (14), and solve the linear program

$$\begin{aligned} &\min E_M(\mathbf{p}) \\ &\text{s.t.} \quad \mathbf{p} \in \Omega \cap M \end{aligned} \tag{16}$$

to obtain a basic optimal solution $\boldsymbol{\omega}(M)$ and the optimal value $\beta(M)$. $\beta(M)$ is the lower bound for region $\Omega \cap M$.

7 Let $\mathbf{p}^k$ equal to the best among all feasible solutions so far encountered: $\mathbf{p}^{k-1}$ and all $\boldsymbol{\omega}(M), M \in \mathscr{N}_k$. Delete all rectangles $M \in \mathscr{M}_k$ such that $\beta(M) \geq E(\mathbf{p}^k) - \epsilon$. Let $\mathscr{R}_k$ be the remaining collection of rectangles.

8 If $\mathscr{R}_k = \emptyset$, terminate: $\mathbf{p}^k$ is the global $\epsilon$-minimum solution.

9 Select $n_3 \triangleq \min(|\mathscr{R}_k|, n_2)$ rectangles $M_{ki} \in \mathscr{R}_k$, $i = 1, \ldots, n_3$ yielding the lowest lower bounds and divide $M_{ki}$ into two subrectangles $M_{ki1}$, $M_{ki2}$ based on the bisection rule stated in Section 5.3. Here $|\cdot|$ denotes the cardinality of a set.

10 Let $\mathscr{N}_{k+1} = \{M_{ki1}, M_{ki2}, i = 1, \ldots, n_3\}$ and $\mathscr{M}_{k+1} = (\mathscr{R}_k \backslash \{M_{ki}, i = 1, \ldots, n_3\}) \cup \mathscr{N}_{k+1}$.

11 **end**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



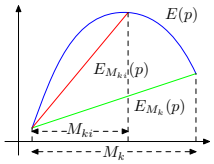



Fig. 2. If $M_{ki} \subset M_k$, we have $E_{M_{ki}}(\mathbf{p}) \geq E_{M_k}(\mathbf{p})$ for $\mathbf{p} \in M_{ki}$ because of the concavity of $E$.

If the third case holds, we have $\beta(M_l') \geq \beta(M_k)$ as $\beta(M_l) \geq \beta(M_k)$ in case 2.

Thus, in each case, the lower bound for $\Omega$ at iteration $k+1$ is not smaller than that at iteration $k$. □

As an instance of the BnB approaches, the worst case time complexity of the proposed algorithm is exponential. Nevertheless, it is well known that the convergence speed of a BnB algorithm mainly depends on the dimension of the search space. That is, a BnB algorithm converges faster when the search space is smaller. For the problems considered in this work, the search space has the dimension of $n_\theta$, which is low and independent of the sizes of two point sets. Thus, our algorithm converges quickly and scales well with problem size, as opposed to existing methods [37].

### 5.5 Fast Lower Bounding Scheme

The proposed BnB algorithm contains three basic subroutines: branching, finding upper and lower bounds. It is clear that the lower bounding subroutine (16) is computationally more expensive than the other two subroutines since it is a generic linear program for which there are no efficient algorithms. To address this issue, we propose a novel lower bounding scheme which is computationally efficient and retains the tightness of the original bounding scheme.

A natural idea to find lower bounds efficiently is to relax the constraint $\mathbf{p} \in \Omega \cap M$ to $\mathbf{p} \in \Omega$, such that we have the following problem:

$$\begin{aligned} &\min E_M(\mathbf{p}) \\ &\text{s.t.} \quad \mathbf{p} \in \Omega \end{aligned} \tag{17}$$

This is a linear assignment problem which can be efficiently solved by the Jonker-Volgenant algorithm [23]. The following proposition shows the validity of (17) for computing the lower bound of $E$ for $\Omega$.

*Proposition 5:* The optimal function value of (17) is a lower bound of $E$ for region $\Omega \cap M$.

*Proof:* Problem (17) is a relaxed problem of (16) by removing the constraint $\mathbf{p} \in M$. Thus, the optimal function value of (17) is not greater than that of (16), whereas solving (16) yields a lower bound of $E$ for the region $\Omega \cap M$. □

In the following, we refer to the methods based on (17) or (16) as the fast bounding or the original schemes, respectively.

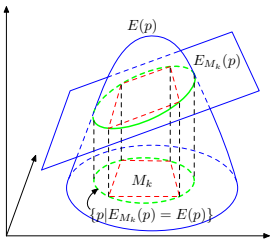



Fig. 3. Ellipse $\{\mathbf{p}|E_{M_k}(\mathbf{p}) = E(\mathbf{p})\}$ is the circumscribed ellipse of rectangle $M_k$ and their centers also coincides.

What remains is to show whether the lower bound by the fast bounding scheme is as tight as the original scheme. At iteration $k$ of our algorithm, let's suppose that $M_k = \{\mathbf{p}|r_i^k \leq \mathbf{u}_i^\top \mathbf{p} \leq s_i^k, i = 1, \ldots, n_\theta\} \in \mathscr{M}_k$ is the rectangle yielding the lower bound for $\Omega$ when using the fast bounding scheme. Clearly, there are two possibilities for the location of the optimal solution $\widehat{\mathbf{p}} = \arg\min_{\mathbf{p}\in\Omega} E_{M_k}(\mathbf{p})$: either $\widehat{\mathbf{p}} \in \{\mathbf{p}|E_{M_k}(\mathbf{p}) \leq E(\mathbf{p})\}$ or $\widehat{\mathbf{p}} \in \{\mathbf{p}|E_{M_k}(\mathbf{p}) > E(\mathbf{p})\}$, as illustrated in Fig. 3. We note that the latter case does not occur since it means $E_{M_k}(\widehat{\mathbf{p}})$ is strictly larger than the minimum of $E$ over $\Omega$, contradicting the assumption that $E_{M_k}(\widehat{\mathbf{p}})$ is a lower bound of $E$ for $\Omega$.

Thus, the only possible case is $\widehat{\mathbf{p}} \in \{\mathbf{p}|E_{M_k}(\mathbf{p}) \leq E(\mathbf{p})\}$. By substituting the formulas of $E_{M_k}$ and $E$ into this inequality, we have:

$$\sum_{i=1}^{n_\theta} \lambda_i (\mathbf{u}_i^\top \mathbf{p} - \frac{r_i^k + s_i^k}{2})^2 \leq \frac{1}{4} \sum_{i=1}^{n_\theta} \lambda_i (s_i^k - r_i^k)^2$$

Since $\lambda_i > 0$ for all $i$, we conclude that this region is an ellipsoid with a center coinciding with that of $M_k$. In addition, the vertices of $M_k$ satisfy this formula with equality. Therefore, the ellipse $\{\mathbf{p}|E_{M_k}(\mathbf{p}) = E(\mathbf{p})\}$ is the circumscribed ellipse of rectangle $M_k$, as illustrated in Fig. 3. The length of the $j$-th axis of the ellipsoid is $\frac{1}{2}\sqrt{\frac{\sum_i \lambda_i (s_i^k - r_i^k)^2}{\lambda_j}}$, which is determined by lengths $|s_i^k - r_i^k|$ of all edges of rectangle $M_k$. If the length of the longest edge of $M_k$ decreases to 0, the lengths of all axes of the ellipsoid also decrease to 0, and the lower bound for $\Omega$ by the fast bounding scheme and $\min_{p\in\Omega\cap M_k} E_{M_k}(\mathbf{p})$ is close to each other. This is the case with our algorithm since the rectangle yielding the lower bound for $\Omega$ is chosen to be further subdivided in each iteration and thus shrinks with the increase of iterations.

Regarding the difference between the lower bounds for $\Omega$ by the two bounding schemes, we have the following proposition.

*Proposition 6:* At iteration $k$, assume the same set $\mathscr{M}_k$ of rectangles is used by the two bounding schemes, then the lower bound for $\Omega$ by the original scheme is not smaller than that by the fast scheme and not larger than $\min_{p\in\Omega\cap M_k} E_{M_k}(\mathbf{p})$.

*Proof:* Assume $M_k^*$ is the rectangle yielding the lower bound for $\Omega$ when using the original bounding scheme,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



we have $\min_{p\in\Omega\cap M_k^*} E_{M_k^*}(\mathbf{p}) \leq \min_{p\in\Omega\cap M_k} E_{M_k}(\mathbf{p})$. Thus, the second part of the proposition is proved.

For the first part of the proposition, it is clear that $\min_{p\in\Omega} E_{M_k^*}(\mathbf{p}) \leq \min_{p\in\Omega\cap M_k^*} E_{M_k^*}(\mathbf{p})$ since $\Omega\cap M_k^* \subset \Omega$. We also have $\min_{p\in\Omega} E_{M_k}(\mathbf{p}) \leq \min_{p\in\Omega} E_{M_k^*}(\mathbf{p})$ since $M_k$ is the rectangle yielding the lower bound for $\Omega$ when using the fast bounding scheme. It follows that $\min_{p\in\Omega} E_{M_k}(\mathbf{p}) \leq \min_{p\in\Omega\cap M_k^*} E_{M_k^*}(\mathbf{p})$. Thus, the first part of the proposition is proved. □

Based on the above analysis, we show that the difference between the lower bounds by the two schemes decreases with the increase of iterations. Therefore, the fast bounding scheme can serve as a good candidate for computing lower bound in the proposed algorithm. Experimental results in Section 6.1.2 also show that the lower bounds by the two schemes differ only slightly for practical problems. We note that Proposition 4 no longer holds for the fast bounding scheme. Nevertheless, since the lower bound by the fast bounding scheme is a close approximation of that by the original scheme, the properties of the original scheme are all retained.

### 5.6 Tolerance Error

The tolerance error $\epsilon$ serves to determine when to terminate our algorithm. It is the maximum permitted error between the upper and lower bounds of the energy function. Here we analyze the form of the energy function to determine its value.

When two point sets are matched, the energy function (3) can be interpreted as the sum of squared distances between the model points and their matched scene points as well as the error contributed by regularization of the spatial transformation. If the latter part can be ignored (this is the case when there is no regularization on transformation or when the actual transformation is close to the expected transformation) and if we require that the average distance between each model point and its matched scene point should not exceed $\epsilon_d$, then the tolerance error $\epsilon$ should be set as $n_x\epsilon_d^2$.

### 5.7 Weight Matrix $H$ and Convergence

The use of the weight matrix $H$ in the energy function $E$ helps increase the convergence rate of the proposed algorithm. From the formula of $E$ in (8) and (9), with the increase of the norm of $H$, the relative weight of the quadratic term over the linear terms in $E$ decreases and $E$ becomes less concave and more linear. Thus, the difference between $E$ and its lower bounding function (which is linear) decreases, which leads to better convergence of the proposed algorithm. Experimental results in Section 6.2.1 show that the run time of the proposed algorithm decreases nearly logarithmically with the increase of the norm of $H$.

## 6 Experiments

We implement the proposed APM algorithm and state-of-the-art methods in Matlab on a computer with 3.3 GHz CPU and 16 GB RAM. For the evaluated methods that generate only point correspondence, the affine transformation is used to model the non-rigid spatial mapping between two point sets. The matching error metric is the average Euclidean distance between the transformed model points and their ground truth data. We set $n_1$ to be 9 for the proposed algorithm (the reason for this choice is given in Section 6.1.3). In the following, we distinguish the cases that $\boldsymbol{\theta}$ is not regularized (i.e., we set the weight matrix $H = 0$) and regularized. For the former case, our method is invariant to the corresponding transformation. The source code and datasets used in the experiments are available at http://www4.comp.polyu.edu.hk/~cslzhang/APM.htm.

### 6.1 Case One: $\theta$ is not Regularized

Both 2D similarity and affine transformations are considered for evaluation. A 2D similarity transformation has the form:

$$T(\mathbf{x}_i|\boldsymbol{\theta}) = \begin{bmatrix}\theta_1 & -\theta_2\\ \theta_2 & \theta_1\end{bmatrix}\begin{bmatrix}x_i^1\\ x_i^2\end{bmatrix} + \begin{bmatrix}\theta_3\\ \theta_4\end{bmatrix} = J(\mathbf{x}_i)\boldsymbol{\theta} \tag{18}$$

where the parameters $\boldsymbol{\theta}$ consist of $[\theta_1,\ldots,\theta_4]^\top$ and $[\theta_3,\theta_4]^\top$ describe translation and $\theta_1 = s\cos(\phi)$, $\theta_2 = s\sin(\phi)$. Here $s$ denotes scale and $\phi$ denotes rotation angle. Therefore the Jacobian matrix is

$$J(\mathbf{x}_i) = \begin{bmatrix}x_i^1 & -x_i^2 & 1 & 0\\ x_i^2 & x_i^1 & 0 & 1\end{bmatrix} \tag{19}$$

A 2D affine transformation has the form:

$$T(\mathbf{x}_i|\boldsymbol{\theta}) = \begin{bmatrix}\theta_1 & \theta_2\\ \theta_3 & \theta_4\end{bmatrix}\begin{bmatrix}x_i^1\\ x_i^2\end{bmatrix} + \begin{bmatrix}\theta_5\\ \theta_6\end{bmatrix} = J(\mathbf{x}_i)\boldsymbol{\theta} \tag{20}$$

where the parameter vector $\boldsymbol{\theta}$ consists of $[\theta_1,\ldots,\theta_6]^\top$ with $[\theta_1,\ldots,\theta_4]^\top$ and $[\theta_5,\theta_6]^\top$ being linear transformation and translation parts, respectively. Therefore the Jacobian matrix is

$$J(\mathbf{x}_i) = \begin{bmatrix}x_i^1 & x_i^2 & 0 & 0 & 1 & 0\\ 0 & 0 & x_i^1 & x_i^2 & 0 & 1\end{bmatrix} \tag{21}$$

#### 6.1.1 Experiments on Synthetic Data

We compare APM with state-of-the-art methods including minimum spanning tree induced triangulation (MSTT) [31], integer projected fixed-point (IPFP) [25], reweighted random walk (RRWM) [7], max pooling matching (MPM) [8] and tensor matching (TM) [12] schemes. Note these methods are rotation invariant and MSTT is globally optimal.

We first use the synthetic data set by Chui and Rangarajan [10] to evaluate robustness of these methods to non-rigid deformation and positional noise. In each test, the model shape is subject to random rotation and one of the disturbances to generate a scene point set (for the noise test, a moderate degree of non-rigid deformation is also included). Fig. 4 shows the model shapes (a tropical fish and a Chinese character) and examples of scene point sets in the two categories of tests.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



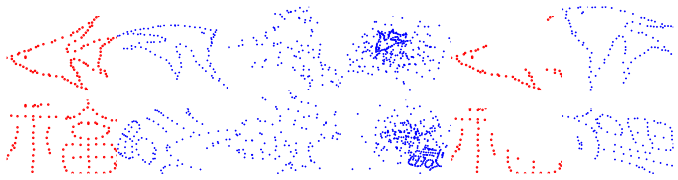


Fig. 4. First 4 columns: model point sets (left column) and examples of scene point sets in the deformation, noise and outlier tests, respectively (column 2 to 4). Last 2 columns: examples of model (column 5) and scene (right column) point sets in the clutter test.

TABLE 1
Average run time (in seconds).

| | Deformation | Noise | Outliers | Clutter |
|---|---|---|---|---|
| APM (simi, 0.1) | **0.4** | **0.3** | 11.6 | 0.4 |
| APM (simi, 0.2) | **0.4** | **0.3** | 3.4 | **0.2** |
| APM (simi, 0.3) | 0.5 | 0.4 | **1.6** | **0.2** |
| APM (aff, 0.1) | **0.4** | 0.4 | 828.2 | 13.3 |
| APM (aff, 0.2) | 0.5 | 0.4 | 114.0 | 1.2 |
| APM (aff, 0.3) | 0.5 | 0.5 | 28.1 | 0.3 |
| MSTT | 11.9 | 12.8 | 29.3 | 3.4 |
| RRWM | 40.5 | 37.7 | 119.9 | 6.5 |
| IPFP | 72.4 | 54.3 | 157.0 | 6.5 |
| MPM | 13.0 | 10.9 | 52.2 | 24.7 |
| TM | 5.7 | 5.9 | 43.5 | 2.4 |

We then evaluate the robustness of these methods to outliers. Points drawn from a normal distribution with unit variance are added to randomly rotated and moderately non-rigidly deformed model shape to generate the scene point set, as illustrated in the fourth column of Fig. 4. Here the mean of the normal distribution is drawn from another standard normal distribution.

We next evaluate the robustness of these methods to clutter. We clip away a disc region of points from the prototype shape to generate the model point set (the clipped points correspond to clutter). The randomly rotated and moderately non-rigidly deformed prototype shape is used as the scene point set. The two shapes (the fish and the Chinese character) are used as the prototype shape, respectively. Fig. 4 shows examples of model and scene point sets (last two columns).

Fig. 5 shows the average matching errors over 100 random trials. The APM algorithm performs well particularly for the outlier test where the errors of APM using affine transformation are close to zero. In contrast, the second order graph matching methods (i.e., RRWM, IPFP and MPM) are not robust to deformation and the tensor based graph matching method is not robust to outliers. For different transformation choices of the APM algorithm, the one using an affine transformation performs better than when using a similarity transformation. This is because an affine transformation accounts for non-rigid deformation well and the proposed algorithm is globally $\epsilon$-optimal with guarantee to find good solutions when $\epsilon_d$ is small.

Fig. 6 shows the results of APM with respect to different values of $\epsilon_d$ for the fish test. The APM algorithm with a similarity transformation is less sensitive to different choices of $\epsilon_d$ than that using affine transformation. This can be explained by that the proposed globally $\epsilon$-optimal algorithm is not guaranteed to find good solutions when $\epsilon_d$ is large. In such situations, the transformation function plays a more important role in shaping the matching results. The transformation function with fewer parameters (e.g., similarity transformation) can more effectively constrain matching results to be regular and desirable.

Table 1 lists the average run time of the evaluated methods. The APM algorithm with a similarity transformation is computationally efficient regardless of the choices of $\epsilon_d$ in all the tests. It is worth noticing that for all but the outlier test, the APM algorithm with a similarity transformation is at least one order of magnitude faster than the other methods. Similar observations can be made for the APM algorithm with an affine transformation on the deformation and noise tests. For the outlier test, the APM method with an affine transformation is considerably slower than that using a similarity transformation as more parameters are involved.

#### 6.1.2 *Effectiveness of the Fast Bounding Scheme*

To evaluate the effectiveness of the fast bounding scheme (17) in comparison to the original scheme (16), we use the outlier test as described in Section 6.1.1 for experiments as it has a large number of points. The outlier to data ratio is set to be $0.5$ and a similarity transformation is used by our algorithm. We set $n_1$ to be $0$ to eliminate the influence of speed-up improvement brought by subdividing multiple rectangles in Algorithm 1.

Fig. 7 shows the results. The lower bounds by the two schemes are the same initially and become slightly different with the increase of iterations. This demonstrates effectiveness of the fast bounding scheme. The average run time of the proposed algorithm through $500$ iterations using the original and fast bounding schemes are $5571.5$ and $11.61$ seconds, respectively. This $480$-fold speed-up improvement demonstrates high computational efficiency of the fast bounding scheme.

#### 6.1.3 *Choice of $n_1$*

To analyze the relationship between the choice of $n_1$ and the speed of APM, we use the outlier test as described in Section 6.1.1 for experiments. Fig. 8 shows the results where $n_1 = 0$ corresponds to the case that only one rectangle is subdivided in each iteration of our algorithm. When an affine transformation is used, the speed-up improvement is significant. With the increase of $n_1$, the run time of APM decreases quickly, particularly for large size problems (e.g., $15$-fold speed-up improvement can be achieved when outlier to data ratio is $1.5$), until $n_1$ reaches a certain value (e.g., $6$ when outlier to data ratio is $1.5$). When a similarity transformation is used, the APM algorithm achieves the fastest speed when $n_1 = 9$. When $n_1 > 9$, the run time of APM increases with the increase of $n_1$. This is because the search space of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



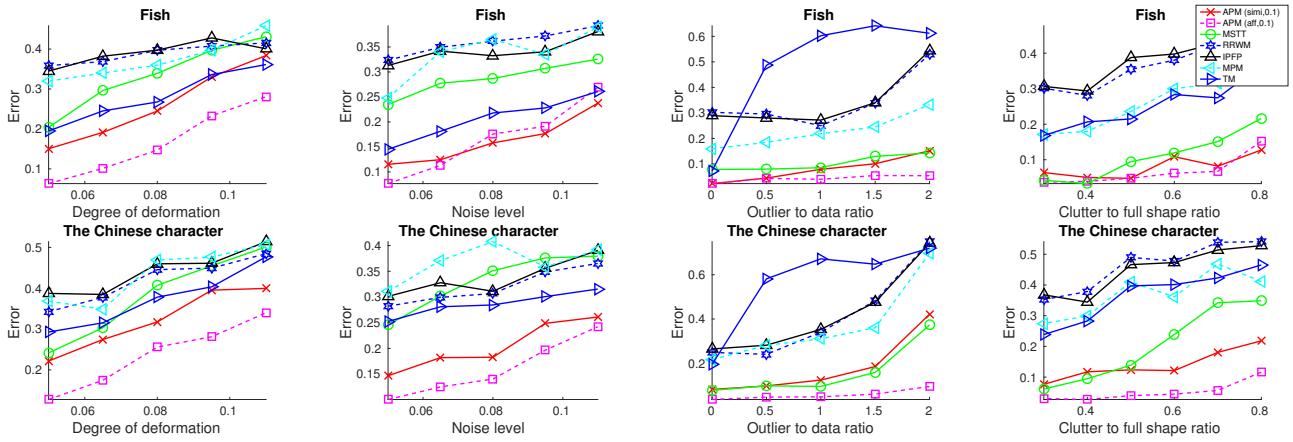



Fig. 5. Average matching errors by APM with $\epsilon_d = 0.1$ and other methods in the tests of deformation, noise, outliers and clutter.

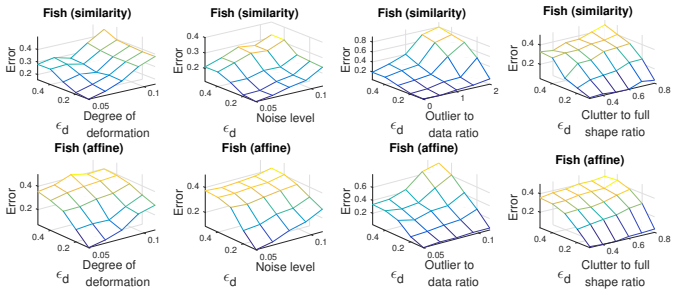



Fig. 6. Average matching error by APM with respect to different choices of $\epsilon_d$ in the tests of deformation, noise, outlier and clutter for the fish shape.

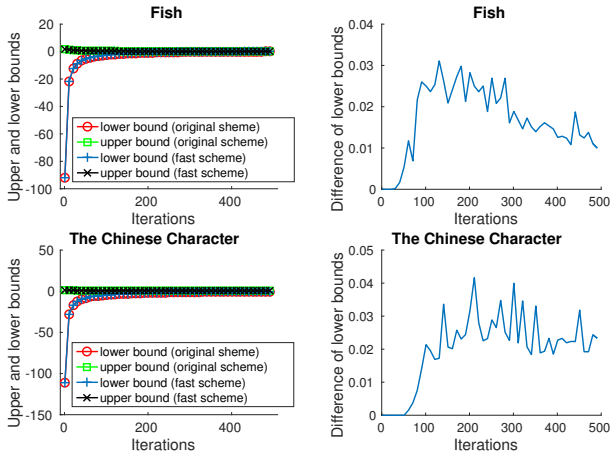



Fig. 7. Average upper and lower bounds for the feasible region by the two bounding schemes (left column) and average differences of the lower bounds (right column) in each iteration of our algorithm.

proposed algorithm in the case of similarity transformation is small and hence overly partitioning the search space is not effective. Based on the above analysis, we choose $n_1 = 9$ for the proposed APM algorithm in this paper.

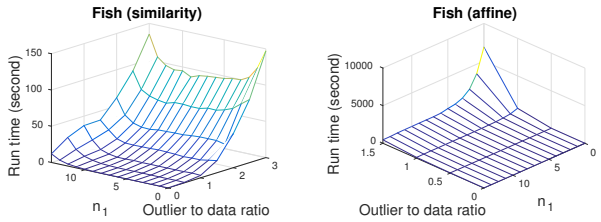



Fig. 8. Average run time of APM with respect to different choices of $n_1$ in the outlier test for the fish shape.

### 6.1.4 *Experiments on Images*

We evaluate the state-of-the-art methods and APM algorithm on point sets manually extracted from four categories from the Caltech-256 [16] and VOC2007 image databases [13], as illustrated in the left panel of Fig. 9. The middle and right panels of Fig. 9 show the matching accuracy (fraction of correct matches) and errors by the methods. Overall, the APM algorithm performs favorably against other methods in terms of matching accuracy. Some matching results by different methods are shown in Fig. 10.

## 6.2 Case Two: $\theta$ is Regularized

When $\boldsymbol{\theta}$ is regularized, the proposed APM algorithm becomes transformation variant. However, the matching accuracy, robustness to disturbances can be improved since the prior of $\boldsymbol{\theta}$ can be utilized. Efficiency can also be further improved since the weight matrix $H$ can be utilized for fast convergence.

### 6.2.1 *2D Point Matching*

Both 2D similarity and affine transformations are considered for evaluation. For similarity transformation, unless otherwise stated, we set the weight matrix $H = \text{diag}([1\ 1\ 0\ 0])$, i.e., we only regularize the linear part. In addition, we set $\boldsymbol{\theta}_0 = [1\ 0\ 0\ 0]^\top$, i.e., the linear part of the transformation should be close to the identity matrix. For affine transformation, unless otherwise stated, we set $H = \text{diag}([1\ 1\ 1\ 1\ 0\ 0])$, i.e., we only regularize

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



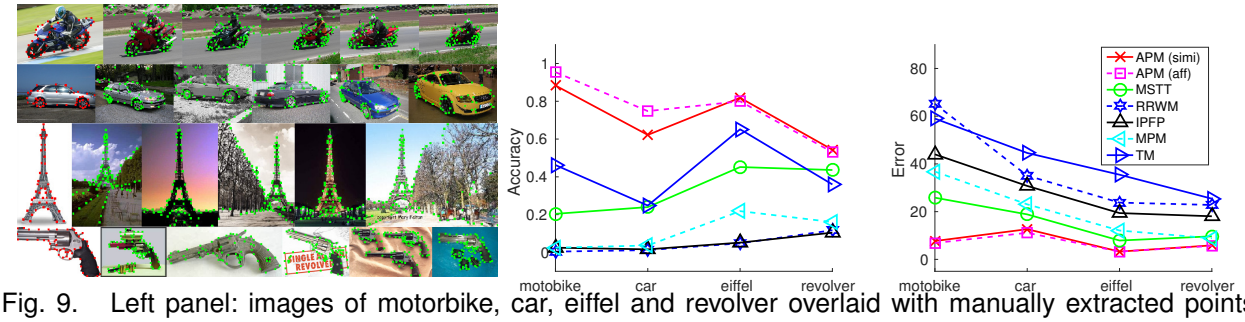



Fig. 9. Left panel: images of motorbike, car, eiffel and revolver overlaid with manually extracted points. The first image in each category is used to extract model points and the remaining images are respectively used to extract scene points. Middle and right panels: matching accuracy and errors by APM with $\epsilon_d = 1$ and other methods.

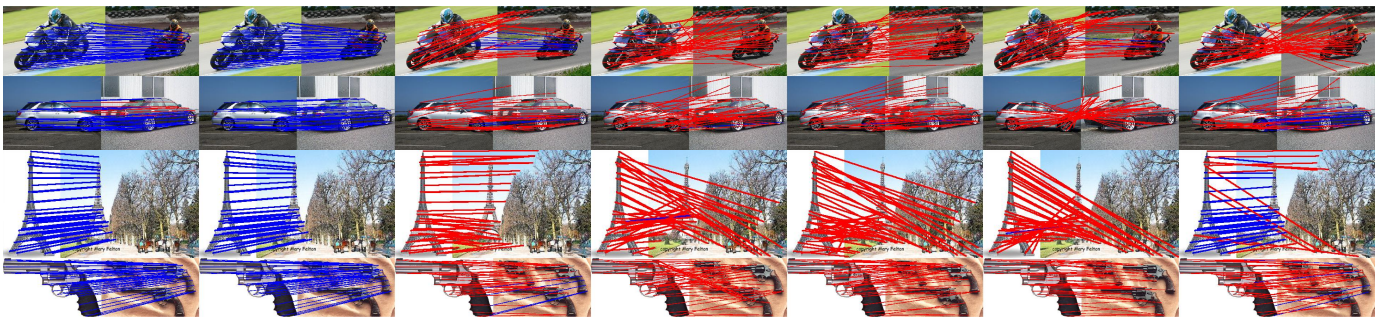


Fig. 10. Examples of matching results by (from left to right column) APM using similarity and affine transformation, MSTT, RRWM, IPFP, MPM and TM on images (from top to bottom) of motorbike, car, eiffel and revolver. Here blue lines indicate correct matches and red lines indicate wrong matches.

the linear part of the transformation. In addition, we set $\boldsymbol{\theta}_0 = [1\ 0\ 0\ 1\ 0\ 0]^\top$, i.e., the linear part of the transformation should be close to the identity matrix.

We compare the APM algorithm with the state-of-the-art RPM [10], CPD [34], Gaussian mixture model based registration (GMMREG) [19] and local neighborhood structure (LNS) preserving [44] methods using the original source codes. Affine transformation is used for CPD and GMMREG, while both affine and thin plate spline (TPS) transformations are considered for RPM. For fair comparisons, we adapt the original RPM algorithm to make it also satisfy the constraints in (4). We evaluate these methods in terms of robustness to non-rigid deformation, positional noise, outliers and clutters using the same experimental setup as in Section 6.1.1 except that there is no rotation between two point sets. We also evaluate these methods in terms of robustness to rotation changes between two point sets.

Fig. 11 shows the average matching errors over 100 random trials. The APM algorithm performs well particularly for the clutter test where the errors are close to zero. We note that the RPM method using an affine transformation does not perform well for deformation and noise tests. In addition, the RPM method needs to use a more complex transformation, TPS, to perform as well as the APM algorithm using an affine transformation. The APM algorithm using an affine transformation performs better than that using a similarity transformation in the deformation test since an affine transformation accounts for non-rigid deformation better.

Table 2 shows the average run time of the evaluated methods. For all but the outlier test, the APM algorithm using either similarity or affine transformation is as efficient as the other methods. For the outlier test, the APM algorithm with $\epsilon_d = 0.1$ using an affine transformation is not efficient as the other methods as more parameters are involved. Fig. 12 shows the average run time of the APM algorithm versus the number of scene points for the outlier test. The run time of the APM algorithm increases almost linearly with the problem size. Fig. 13 shows the average run time of the APM algorithm versus the spectral norm $\|H\|_2$ of the weight matrix $H$ for the outlier test, (the outlier to data ratio is set to 1.5). We set $H$ to be $\text{diag}([\omega\ \omega\ 0\ 0])$ for a similarity transformation and $H$ to be $\text{diag}([\omega\ \omega\ \omega\ \omega\ 0\ 0])$ for an affine transformation, where the parameter $\omega > 0$. The run time of the APM algorithm decreases nearly logarithmically with the increase of $\|H\|_2$. The results show the regularization term helps improve the convergence of the APM algorithm.

### 6.2.2 3D Point Matching

We consider 3D affine transformation for evaluation. A 3D affine transformation has the form:

$$T(\mathbf{x}_i|\boldsymbol{\theta}) = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \\ \theta_7 & \theta_8 & \theta_9 \end{bmatrix} \begin{bmatrix} x_i^1 \\ x_i^2 \\ x_i^3 \end{bmatrix} + \begin{bmatrix} \theta_{10} \\ \theta_{11} \\ \theta_{12} \end{bmatrix} = J(\mathbf{x}_i)\boldsymbol{\theta} \quad (22)$$

where the parameter vector $\boldsymbol{\theta}$ consists of $\left[\theta_1, \ldots, \theta_{12}\right]^\top$ with $[\theta_1, \ldots, \theta_9]^\top$ being the linear part of the transforma-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



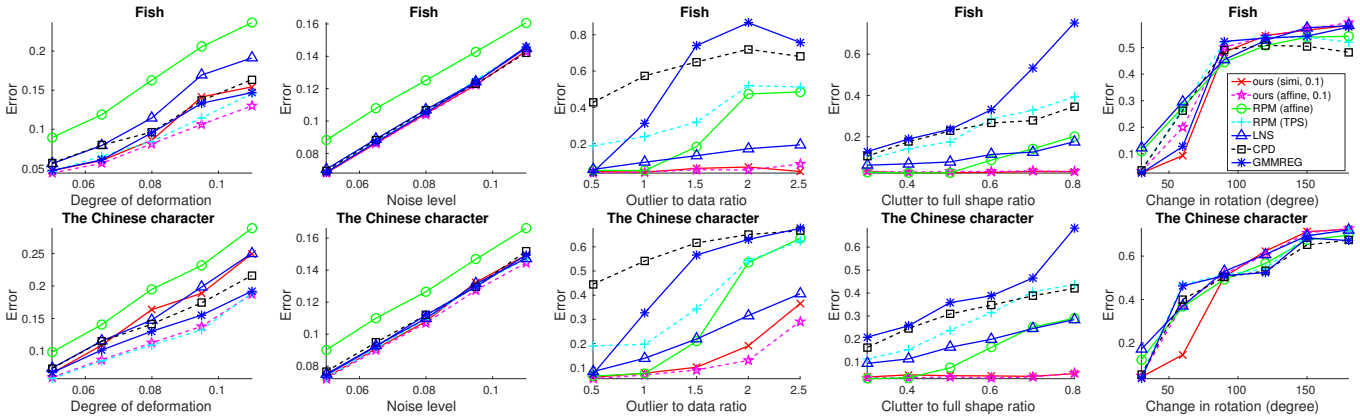



Fig. 11. Average matching errors by APM with $\epsilon_d = 0.1$ and other methods in the tests of deformation, noise, outlier, clutter and rotation.

TABLE 2
Average run time (in seconds).

| | Deformation | Noise | Outlier | Clutter | Rotation |
|---|---|---|---|---|---|
| APM (simi, 0.1) | 0.6 | 0.5 | 6.8 | 0.3 | 0.5 |
| APM (aff, 0.1) | 0.5 | 0.4 | 119.1 | 0.6 | 0.5 |
| RPM (aff) | 1.2 | 0.8 | 1.4 | 0.3 | 0.3 |
| RPM (TPS) | 1.8 | 1.4 | 1.8 | 0.3 | 1.5 |
| LNS | 3.9 | 4.4 | 19.0 | 1.4 | 5.4 |
| CPD | **0.1** | **0.1** | **0.1** | **0.1** | **0.1** |
| GMMREG | **0.1** | 0.2 | 0.5 | 0.2 | 0.2 |

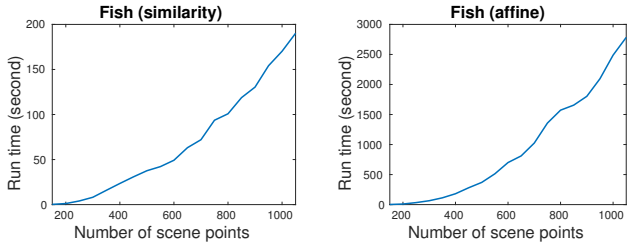



Fig. 12. Average run time of APM with $\epsilon_d = 0.1$ using similarity (left) or affine (right) transformation with respect to different number of scene points for the outlier test.

tion and $[\theta_{10}, \ldots, \theta_{12}]^\top$ being the translation component. Therefore the Jacobian matrix is

$$J(\mathbf{x}_i) = \begin{bmatrix} x_i^1 & x_i^2 & x_i^3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x_i^1 & x_i^2 & x_i^3 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_i^1 & x_i^2 & x_i^3 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$

For the APM algorithm, we set the weight matrix $H = 10 \cdot \mathrm{diag}([\underbrace{1, \ldots, 1}_{9}, 0, 0, 0])$, such that only the linear part of the transformation is regularized. In addition, we set $\boldsymbol{\theta}_0 = [1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^\top$ such that the linear part of the transformation is close to the identity matrix.

We compare the APM algorithm with the state-of-the-art methods except for LNS since it employs the shape context feature descriptor [2], which is only applicable to the 2D case. Affine transformation is used by the RPM, CPD and GMMREG methods. In addition, we compare

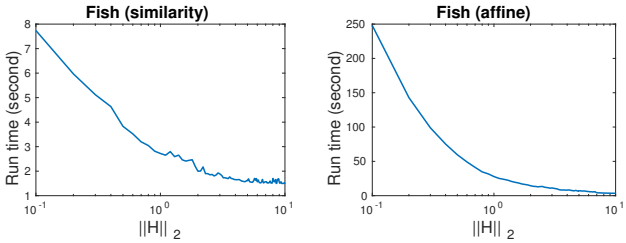



Fig. 13. Average run time of APM with $\epsilon_d = 0.1$ using similarity (left) or affine (right) transformation with respect to different spectral norm of $H$ for the outlier test.

with the Go-ICP method [41], which is globally optimal and applicable to rigid registration problems.

We use the point sets (with cardinality ranging from 353 to 462) subsampled from 10 different shapes from the Stanford 3D scanning repository (http://graphics.stanford.edu/data/3Dscanrep/) and the AIM@SHAPE shape repository (http://shapes.aimatshape.net/) as shown in Fig. 14 for experiments. The experimental setup for the deformation, noise and clutter tests is the same as in Section 6.2.1. Fig. 15 shows examples of model and scene point sets in these three tests. Fig. 16 shows the average matching errors over 100 random trials (where 10 random trials are respectively conducted for each of the 10 different shapes). Overall, the APM algorithm performs well particularly for the clutter test, where the error remains almost unchanged with the increase of severity of clutter. Fig. 17 shows examples of matching results by different methods in the clutter test. Table 3 lists the average run time of the evaluated methods. The APM algorithm is as efficient as the RPM method for the deformation and noise tests. In contrast, the Go-ICP scheme is not efficient for the deformation test.

## 7 Concluding Remarks

We proposed a novel point matching algorithm in this paper. The proposed algorithm reduces the objective function of the RPM method to a separable concave quadratic function with a few nonlinear terms, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



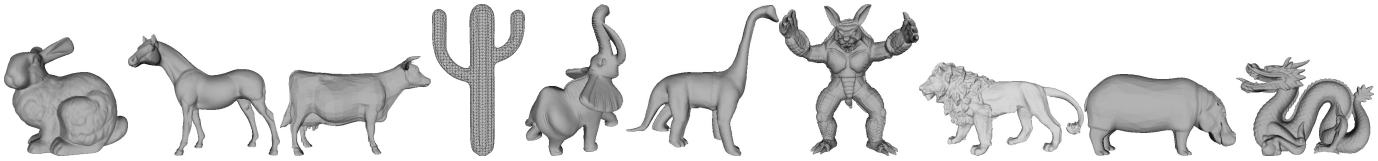


Fig. 14. 3D shapes used for generating prototype point sets.

Fig. 15. Left 3 columns: an example of model (left column) and scene point sets (represented by mesh nodes) in the deformation and noise tests, respectively (column 2 and 3). Right 2 columns: an example of model point set (represented by red mesh nodes in column 4, green mesh nodes correspond to clutter) and scene point set (right column) in the clutter test.

TABLE 3
Average run time (in seconds).

| | Deformation | Noise | Clutter |
|---|---|---|---|
| APM (0.2) | 17.5 | 10.9 | 25.8 |
| RPM | 13.8 | 8.2 | 3.2 |
| Go-ICP | 336.9 | 6.5 | 1.9 |
| CPD | **0.5** | **0.3** | **0.2** |
| GMMREG | 2.3 | 2.4 | 1.0 |

the optimal solution is computed by a normal rectangular branch-and-bound approach. By recasting the lower bounding problem, we obtain a linear assignment formulation which can be efficiently solved. The resulting algorithm finds global optimal solutions, and scales well with data size. Experimental results on both synthetic and real data sets showed that the proposed algorithm performs favorably in terms of robustness to disturbances and matching accuracy against the state-of-the-art methods with comparable computational load.

## REFERENCES


[1] J.-C. Bazin, H. Li, I. S. Kweon, C. P. Vasseur, and K. Ikeuchi. A branch-and-bound approach to correspondence and grouping problems. *PAMI*, 35(7):1565–1576, 2013. 3

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002. 2, 12

[3] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *PAMI*, 14(2):239–256, 1992. 2

[4] D. Breitenreicher and C. Schnörr. Model-based multiple rigid object detection and registration in unstructured range data. *IJCV*, 92(1):32–52, Mar. 2011. 2

[5] L. G. Brown. A survey of image registration techniques. *ACM Comput. Surv.*, 24(4):325–376, 1992. 1, 2

[6] T. S. Caetano and T. Caelli. A unified formulation of invariant point pattern matching. In *ICPR*, pages 121–124, 2006. 2

[7] M. Cho, J. Lee, and K. M. Lee. the reweighted random walks matching algorithm. In *ECCV*, 2010. 2, 8

[8] M. Cho, J. Sun, O. Duchenne, and J. Ponce. Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2, 8

[9] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *CVPR*, volume 2, pages 44–51, 2000. 1, 2

[10] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *CVIU*, 89(2-3):114–141, 2003. 1, 2, 3, 8, 11

[11] Y. Deng, A. Rangarajan, S. Eisenschenk, and B. C. Vemuri. A riemannian framework for matching point clouds represented by the schrödinger distance transform. In *CVPR*, 2014. 2

[12] O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *PAMI*, (12):2383–2395, 2011. 2, 8

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 10

[14] J. Glaunes, A. Trouve, and L. Younes. Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and sub-manifolds matching. In *CVPR*, pages 712–718, 2004. 2

[15] G. H. Golub and C. F. V. Loan. *Matrix computations*. The Johns Hopkins University Press., 1996. 4

[16] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset, 2007. technical report, California Inst. of Technology. 10

[17] J. Ho, A. Peter, A. Rangarajan, and M.-H. Yang. An algebraic approach to affine registration of point sets. In *CVPR*, 2009. 2

[18] R. Horst and H. Tuy. *Global Optimization, Deterministic Approaches*. Springer-Verlag, 1996. 1, 5, 6

[19] B. Jian and B. C. Vemuri. A robust algorithm for point set registration using mixture of gaussians. In *ICCV*, volume 2, pages 1246–1251, 2005. 1, 2, 11

[20] H. Jiang, M. S. Drew, and Z.-N. Li. Matching by linear programming and successive convexification. *PAMI*, 29(6):959–975, 2007. 1, 2, 4

[21] H. Jiang, T.-P. Tian, and S. Sclaroff. Scale and rotation invariant matching using linearly augmented trees. *PAMI*, 37(12):2558–2572, 2015. 3

[22] H. Jiang and S. X. Yu. Linear solution to scale and rotation invariant object matching. In *CVPR*, pages 2474–2481, 2009. 1, 3, 4

[23] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340, 1987. 6, 7

[24] J.-H. Lee and C.-H. Won. Topology preserving relaxation labeling for nonrigid point matching. *PAMI*, 33(2):427–432, 2011. 2

[25] M. Leordeanu, M. Hebert, and R. Sukthankar. the integer projected fixed point matching algorithm. In *NIPS*, 2009. 2, 8

[26] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *CVPR*, 2009. 3

[27] H. Li, X. Huang, and L. He. Object matching using a locally affine invariant and linear programming techniques. *PAMI*, 35(2):411–424, 2013. 1, 3, 4

[28] W. Lian and L. Zhang. Rotation invariant non-rigid shape matching in cluttered scenes. In *ECCV*, 2010. 3

[29] W. Lian and L. Zhang. Robust point matching revisited: a concave optimization approach. In *ECCV*, 2012. 3

[30] W. Lian and L. Zhang. Point matching in the presence of outliers in both point sets: A concave optimization approach. In *CVPR*, 2014. 3

[31] W. Lian, L. Zhang, and D. Zhang. Rotation-invariant nonrigid point set matching in cluttered scenes. *TIP*, 21(5):2786–2797, 2012. 3, 4, 8

[32] J. Ma, J. Zhao, J. Tian, Z. Tu, and A. L. Yuille. Robust estimation of nonrigid transformation for point set registration. In *CVPR*, 2013. 2

[33] J. Maciel and J. Costeira. A global solution to sparse correspondence problems. *PAMI*, 25(2):187–199, 2003. 1, 3, 4

[34] A. Myronenko and X. Song. Point set registration: Coherent point drift. *PAMI*, 32(12):2262–2275, 2010. 1, 2, 11

[35] Q. Ngoc, A. Gautier, and M. Hein. A flexible tensor block coordinate ascent scheme for hypergraph matching. In *CVPR*, 2015. 2

[36] C. Olsson, F. Kahl, and M. Oskarsson. Branch-and-bound methods

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2603988, IEEE Transactions on Pattern Analysis and Machine Intelligence



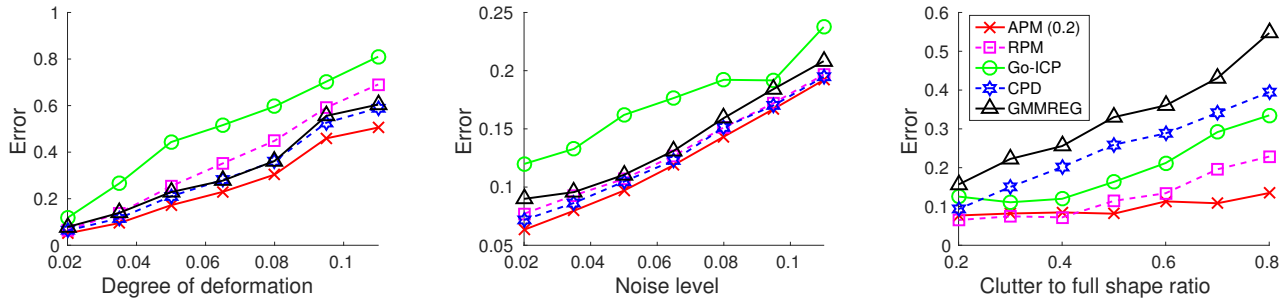



Fig. 16. Average matching errors by APM with $\epsilon_d = 0.2$ and other methods in the 3D tests of deformation, noise and clutter.

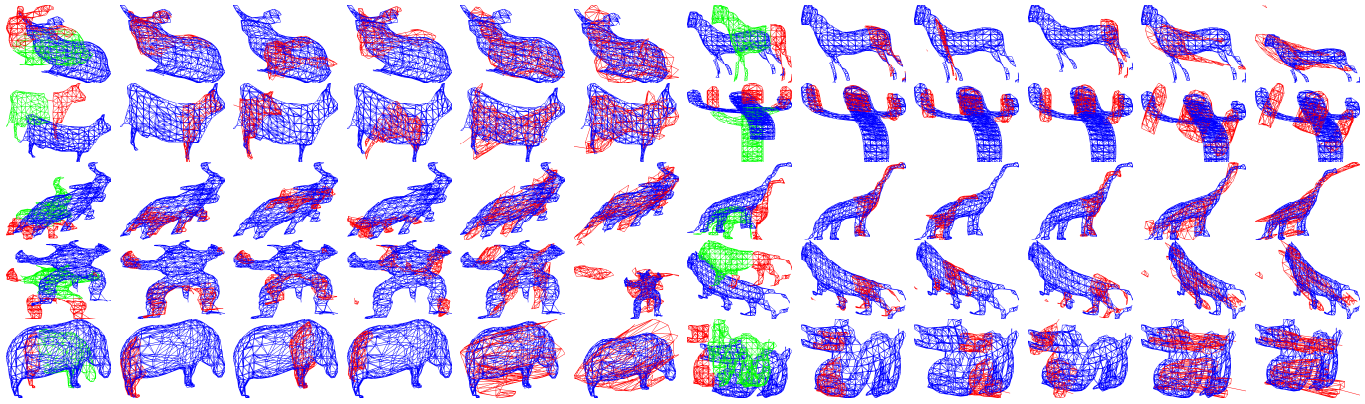
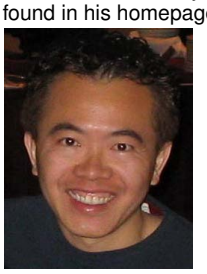


Fig. 17. Examples of matching results by different methods in the clutter test. For every 6 columns, from left to right: model point sets shown as red mesh nodes (green mesh nodes correspond to clutter) and scene point sets shown as blue mesh nodes, matching results by APM with $\epsilon_d = 0.2$, RPM, Go-ICP, CPD and GMMREG.

for euclidean registration problems. *PAMI*, 31(5):783–794, 2009. 3
[37] F. Pfeuffer, M. Stiglmayr, and K. Klamroth. Discrete and geometric branch and bound algorithms for medical image registration. *Annals of Operations Research*, 196(1):737–765, 2012. 3, 7
[38] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000. 3
[39] M. Sofka, G. Yang, and C. V. Stewart. Simultaneous covariance driven correspondence (cdc) and transformation estimation in the expectation maximization framework. In *CVPR*, 2007. 1, 2
[40] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *ECCV*, pages 558–569, 2004. 2
[41] J. Yang, H. Li, and Y. Jia. Go-icp: Solving 3d registration efficiently and globally optimally. In *ICCV*, 2013. 3, 12
[42] A. L. Yuille and J. J. Kosowsky. Statistical physics algorithms that converge. *Neural Comput.*, 6(3):341–356, 1994. 1
[43] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *IJCV*, 13(2):119–152, 1994. 2
[44] Y. Zheng and D. Doermann. Robust point matching for non-rigid shapes by preserving local neighborhood structures. *PAMI*, 28(4):643–649, 2006. 2, 11
[45] B. Zitov and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003. 1, 2

**Wei Lian** is an associate professor with Department of Computer Science at Changzhi University, Changzhi, China. He received the B.S. degree in automation from Taiyuan University of Technology, Taiyuan, China, in 2000 and the M.S. and Ph.D. degrees in automation control theory and engineering from Northwestern Polytechnical University, Xian, China, in 2003 and 2007, respectively. He was a research assistant/associate in the Dept. of Computing, the Hong Kong Polytechnic University From July 2007 to Feb. 2015.

**Lei Zhang** received the B.Sc. degree in 1995 from Shenyang Institute of Aeronautical Engineering, Shenyang, P.R. China, the M.Sc. and Ph.D degrees in Control Theory and Engineering from Northwestern Polytechnical University, Xian, P.R. China, respectively in 1998 and 2001. From 2001 to 2002, he was a research associate in the Dept. of Computing, The Hong Kong Polytechnic University. From Jan. 2003 to Jan. 2006 he worked as a Postdoctoral Fellow in the Dept. of Electrical and Computer Engineering, McMaster University, Canada. In 2006, he joined the Dept. of Computing, The Hong Kong Polytechnic University, as an Assistant Professor. Since July 2015, he has been a Full Professor in the same department. Dr. Zhang has published more than 200 papers. By 2016, his publications have been cited more than 18,000 times in literature. Dr. Zhang is an Associate Editor of IEEE Trans. on Image Processing, IEEE Trans. on CSVT and Image and Vision Computing. He was selected as the "Highly Cited Researcher" by Thomson Reuters, 2015. More information can be found in his homepage http://www4.comp.polyu.edu.hk/~cslzhang/.



**Ming-Hsuan Yang** is an associate professor in Electrical Engineering and Computer Science at University of California, Merced. He received the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2000. Prior to joining UC Merced in 2008, he was a senior research scientist at the Honda Research Institute. Yang served as an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence from 2007 to 2011, and is an associate editor of the International Journal of Computer Vision, Image and Vision Computing, and Journal of Artificial Intelligence Research. He received the Google Faculty Award in 2009 and the NSF CAREER Award in 2012. He is a senior member of the IEEE and the ACM.