Point Matching in the Presence of Outliers in Both Point Sets: A Concave Optimization Approach

Wei Lian
Dept. of Computer Science
Changzhi University, Changzhi, 046011, Shanxi, China
lianwei3@gmail.com

Lei Zhang
Dept. of Computing
The Hong Kong Polytechnic University
cslzhang@comp.polyu.edu.hk

Abstract

Recently, a concave optimization approach has been proposed to solve the robust point matching (RPM) problem. This method is globally optimal, but it requires that each model point has a counterpart in the data point set. Unfortunately, such a requirement may not be satisfied in certain applications when there are outliers in both point sets. To address this problem, we relax this condition and reduce the objective function of RPM to a function with few nonlinear terms by eliminating the transformation variables. The resulting function, however, is no longer quadratic. We prove that it is still concave over the feasible region of point correspondence. The branch-and-bound (BnB) algorithm can then be used for optimization. To further improve the efficiency of the BnB algorithm whose bottleneck lies in the costly computation of the lower bound, we propose a new lower bounding scheme which has a k-cardinality linear assignment formulation and can be efficiently solved. Experimental results show that the proposed algorithm outperforms state-of-the-arts in its robustness to disturbances and point matching accuracy.

1. Introduction

Point matching is a fundamental yet challenging problem in areas such as computer vision, pattern recognition and medical image analysis. Disturbances such as deformations, noise and outliers often makes this problem difficult. Many methods (see Sec. 2 for an overview) have been proposed to address these difficulties, e.g., the classical robust point matching (RPM) method [6]. However, most of them are heuristic schemes, which means their performance depends on if the employed heuristics can well fit the problem.

Recently, Lian and Zhang [14] proposed a concave optimization approach for the objective function of RPM. The method has several desirable advantages over the original RPM. First, it is globally optimal and hence more robust to disturbances; second, it can be rendered invariant to the corresponding transformation when simple transformations such as similarity are employed. But this method assumes that each model point has a counterpart in the data point set. This assumption may fail to hold in certain applications when there are outliers in both point sets.

This paper aims to solve this problem. Similar to [14], by eliminating the transformation variables, we reduce the RPM objective function to a function of point correspondence with few nonlinear terms. Therefore, large scale global optimization techniques can be applied to our problem and our method has the merit of scaling well with problem size as in [14]. Different from [14], however, the resulting function is no longer quadratic, which makes the optimization harder. But we prove that the derived objective function is still concave over the feasible region of point correspondence. Then the branch-and-bound (BnB) algorithm can be employed for optimization. One bottleneck of the BnB algorithm is the costly computation of the lower bound. To address this problem, we further propose a new lower bounding scheme which has a k-cardinality linear assignment formulation and can be efficiently solved.

The rest of the paper is organized as follows. We review related work in Sec. 2 and RPM in Sec. 3. We then discuss the new energy function and its optimization in Sec. 4 and 5. We finally present the experimental results in Sec. 6 and conclude the paper in Sec. 7.

2. Related work

Point matching: The iterative closest point (ICP) method [3] iterates between finding point correspondence based on the nearest neighbor relationship and updating the transformation as a least square problem. ICP is prone to be trapped in local minima because of the discrete nature of point correspondence. To address this problem, RPM [6] relaxes point correspondence to be continuously valued and employs deterministic annealing (DA) for optimization. The coherent point drift (CPD) method [17] uses one point set to represent the Gaussians mixture model (GMM) centroids, and converts point matching into the problem of fitting the GMM centroids to another point set. Kernel correlation was used in [22] for point matching. This method was later generalized in [11] by minimizing the distance between two mixtures of Gaussians (MG) representing two point sets. In [4], an amendment to the objective function in [11] was proposed by using the log-exponential function. Under this formulation, ICP can be interpreted as a special case. The method in [11] was later generalized to group-wise point-set registration in [5]. The covariance driven correspondence method [21] uses the covariance of the transformation to guide the determination of point correspondences, but this method is only suitable for rigid registration. A robust way of estimating transformation was proposed in [15] by utilizing the L_2E estimator. The above methods are all heuristic schemes, hence they cannot guarantee the global optimality of the solutions. Roots-finding technique was used in [9] for affine invariant point set registration. But the method is sensitive to outliers due to use of moments.

Concave optimization: Concave optimization was used in [16] to solve correspondence problems arising from computer vision. The optimization technique employed in [16] is however only suitable for small scale problems. In contrast, our method scales well with problem size due to the special structure of our optimization scheme, which facilitates the use of large scale optimization techniques.

Branch-and-bound: BnB is a popular optimization technique with global optimality guarantee. It was used in [18] to estimate the transformation given the correspondence. In comparison, both types of variables are unknown in our problem, which makes our task more difficult. BnB was used in [13] for 3D rigid point set registration by traversing the space of rotations, but the method does not permit the presence of outliers. In contrast, our method can handle outliers in both point sets. In [1], BnB was used to find the maximum set of consistent point correspondences. But the method is very slow because a generic linear program needs to be solved to obtain the upper bound. In contrast, our method is much faster since, instead of a generic linear program, only a k-cardinality linear assignment problem needs to be solved to obtain the lower bound.

In [20], two BnB algorithms were proposed for the objective function of RPM. The first method directly branches over the correspondence variable. Because of the high dimensionality of the correspondence variable, this method is only applicable to small scale problems. In contrast, our method branches over a new variable whose dimension is only related to the transformation, which is in general very small. The second method in [20] branches over the transformation variable. Because of lack of concavity by the energy function, the lower bound of this method is not tight.

3. The energy function of RPM

Since our method originates from the energy function of RPM [14], we first briefly review RPM in this section. Suppose we are given two point sets: the model point set $\mathscr{X} = \{\mathbf{x}_i, i = 1, \dots, m\}$ with point $\mathbf{x}_i = \left[x_i^1; \cdots; x_i^d\right] \in \mathbb{R}^d$, and the data point set $\mathscr{Y} = \{\mathbf{y}_j, j = 1, \dots, n\}$ with point $\mathbf{y}_j = \left[y_j^1; \cdots; y_j^d\right] \in \mathbb{R}^d$. Our task is to find the transformation and the point correspondence between the two sets so that the transformed model points can best match the data points.

To solve this problem, RPM jointly estimates the transformation and the point correspondence. It models point matching as a mixed linear assignment$-$least square problem:

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

$$s.t. \quad P\mathbf{1}_n \leq \mathbf{1}_m, \quad \mathbf{1}_m^T P \leq \mathbf{1}_n^T, \quad P \geq 0 \tag{1b}$$

Here $P = \{p_{ij}\}$ is the correspondence matrix with $p_{ij} = 1$ if there is a matching between $\mathbf{x}_i$ and $\mathbf{y}_j$ and 0 otherwise. $\mathbf{1}_m$ denotes an $m \times 1$ vector with all ones. $T(\cdot|\boldsymbol{\theta})$ denotes the spatial transformation with parameters $\boldsymbol{\theta}$. $g(\boldsymbol{\theta})$ is a regularization term on $\boldsymbol{\theta}$. To solve problem (1a), (1b), RPM relaxes P to be continuously valued and employs deterministic annealing (DA) for optimization. However, DA is a heuristic scheme which causes RPM to be less robust to disturbances. In the next section, we present a new energy function based on RPM, which is more amenable to global optimization.

4. The new energy function

To make the problem tractable, similar to [14], we assume that the transformation $T(\mathbf{x}_i|\boldsymbol{\theta})$ is linear w.r.t. its parameters $\boldsymbol{\theta}$, i.e., $T(\mathbf{x}_i|\boldsymbol{\theta}) = J(\mathbf{x}_i)\boldsymbol{\theta}$, where $J(\mathbf{x}_i)$ is called the Jacobian matrix. Examples of $J(\mathbf{x}_i)$ are presented in Sec. 6. We consider the following regularizer for $\boldsymbol{\theta}$: $g(\boldsymbol{\theta}) = (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^T H (\boldsymbol{\theta} - \boldsymbol{\theta}_0)$, i.e., $\boldsymbol{\theta}$ is required to be close to a constant vector $\boldsymbol{\theta}_0$. Here H is a positive semidefinite matrix, denoted by $H \succeq 0$, whose entries represent the weights assigned to the elements of $\boldsymbol{\theta}$.

With the above assumption, the energy function (1a) now takes the following form:

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

where $J \triangleq \left[J^T(\mathbf{x}_1), \dots, J^T(\mathbf{x}_m)\right]^T$, $\mathbf{y} \triangleq \left[\mathbf{y}_1^T, \dots, \mathbf{y}_n^T\right]^T$, and $\mathbf{z} \triangleq \left[\|\mathbf{y}_1\|_2^2, \dots, \|\mathbf{y}_n\|_2^2\right]^T$. The operator 'diag' denotes converting a vector into a diagonal matrix. I_d denotes the

d-dimensional identity matrix and $\otimes$ denotes the Kronecker product.

We consider the following constraints on P:

$$P\mathbf{1}_n \leq \mathbf{1}_m, \quad \mathbf{1}_m^T P \leq \mathbf{1}_n^T, \quad \mathbf{1}_m^T P \mathbf{1}_n = s, \quad P \geq 0 \quad (3)$$

i.e., the matching is one-to-one and the number of matches is $s \leq \min(m,n)$, a constant positive integer. The constraints in (3) satisfy the total unimodularity property [16, 19], which implies that the vertices of the polytope (i.e., bounded polyhedron) determined by (3) have integer valued coordinates. By also taking into account the fact that $0 \leq p_{ij} \leq 1$, the coordinates of the vertices are actually $0, 1$ valued.

Given P, E is apparently a convex quadratic function of $\boldsymbol{\theta}$. Therefore, the optimal solution $\hat{\boldsymbol{\theta}}$ of E can be obtained in close form by letting $\frac{dE}{d\boldsymbol{\theta}} = 0$. The result is:

$$\hat{\boldsymbol{\theta}} = (J^T(\text{diag}(P\mathbf{1}_n)\otimes I_d)J+H)^{-1}\left[J^T(P\otimes I_d)\mathbf{y} + H\boldsymbol{\theta}_0\right] \quad (4)$$

By substituting the optimal $\hat{\boldsymbol{\theta}}$ back into (2), the variable $\boldsymbol{\theta}$ is eliminated and we arrive at an energy function which has only one variable P [1],

$$\begin{aligned} E(P) = {} & \mathbf{1}_m^T P\mathbf{z} - [\mathbf{y}^T(P\otimes I_d)^T J + \boldsymbol{\theta}_0^T H^T] \\ & \cdot (J^T(\text{diag}(P\mathbf{1}_n)\otimes I_d)J + H)^{-1}[J^T(P\otimes I_d)\mathbf{y} + H\boldsymbol{\theta}_0] \end{aligned} \quad (5)$$

To facilitate the optimization of E, P needs first to be vectorized. Let us define the vectorization of a matrix as the concatenation of its rows [2], denoted by $\text{vec}(\cdot)$. Let $\mathbf{p} = \text{vec}(P)$. To obtain a new form of E which has fewer nonlinear terms, we need some denotations. Let

$$J^T(P\otimes I_d)\mathbf{y} = A\mathbf{p},$$

$$\text{vec}(J^T(\text{diag}(P\mathbf{1}_n)\otimes I_d)J) = \text{vec}(J_2^T((P\mathbf{1}_n)\otimes I_t)) = B\mathbf{p},$$

where t denotes the dimensionality of $\boldsymbol{\theta}$ and $J_2 \triangleq \left[J(\mathbf{x}_1)^T J(\mathbf{x}_1), \ldots, J(\mathbf{x}_m)^T J(\mathbf{x}_m)\right]^T$. Based on the fact $\text{vec}(KLM) = (K\otimes M^T)\text{vec}(L)$, we have

$$A = (J^T \otimes \mathbf{y}^T)W_d^{m,n},$$

$$B = (J_2^T \otimes I_t)W_t^{m,1}(I_m \otimes \mathbf{1}_n^T),$$

where the $mnd \times mn$ matrix $W_d^{m,n} \triangleq I_m \otimes \left[I_n\otimes(\mathbf{e}_d^1)^T, \ldots, I_n\otimes(\mathbf{e}_d^d)^T\right]^T$ satisfies $\text{vec}(C_{m,n}\otimes I_d) = W_d^{m,n}\text{vec}(C_{m,n})$. Here $\mathbf{e}_d^i$ denotes the d-dimensional column vector with the i-th entry being 1 and all other entries being 0. $W_d^{m,n}$ is a large but sparse matrix and can be implemented using the function *speye* in Matlab.

With the above preparation, E can be written in terms of vector $\mathbf{p}$ as:

$$\begin{aligned} E(\mathbf{p}) = & -(\mathbf{p}^T A^T + \boldsymbol{\theta}_0^T H)(\text{mat}(B\mathbf{p}) + H)^{-1}(A\mathbf{p} + H\boldsymbol{\theta}_0) \\ & + (\mathbf{1}_m^T \otimes \mathbf{z}^T)\mathbf{p} \end{aligned} \quad (6)$$

Here the operator 'mat' denotes reconstructing a symmetric matrix from a vector which is the result of applying 'vec' to a symmetric matrix. Therefore, operator 'mat' can be seen as the inverse of operator 'vec' applied to symmetric matrices and its meaning will be clear from the context.

A is of full rank t, which equals to the number of its rows. B is rank deficient. Let B_2 be the matrix whose rows is formed by the *nonzero unique* rows of B. Examples of B_2 will be presented in Sec. 6. Assume that the number of rows in B_2 is l. Let the QR factorization of $\left[B_2^T, A^T\right]$ be $QR = \left[B_2^T, A^T\right]$, where R is an upper triangular matrix and the columns of Q are orthogonal unity vectors. In light of the form of E in (6), we can see that the nonlinear part E_C of E (i.e., the first term in (6)) is solely determined by variable

$$\mathbf{u} \triangleq R^{-T}\begin{bmatrix} B_2 \\ A \end{bmatrix}\mathbf{p} = Q^T\mathbf{p} \quad (7)$$

The specific form of E_C in terms of variable $\mathbf{u}$ is:

$$\begin{aligned} E_C(\mathbf{u}) = & -((\mathbf{u}^T R)_{l+1,\ldots,l+t} + \boldsymbol{\theta}_0^T H) \\ & \cdot (\text{mat}((R^T\mathbf{u})_{1,\ldots,l}) + H)^{-1}((R^T\mathbf{u})_{l+1,\ldots,l+t} + H\boldsymbol{\theta}_0) \end{aligned} \quad (8)$$

where $(R^T\mathbf{u})_{a,\ldots,b}$ denotes the vector formed by the a-th to b-th elements of vector $R^T\mathbf{u}$. Here we abuse the use of 'mat' so that $\text{mat}(B_2\mathbf{p}) = \text{mat}(B\mathbf{p})$, i.e., a matrix reconstructed from the nonzero unique elements of a vector should equal to the matrix reconstructed directly from the vector. The meaning will be clear from the context. The dimensionality of $\mathbf{u}$ is $l+t$, which is much smaller than that of $\mathbf{p}$ and also independent of the cardinalities of the two point sets. This is the key why our algorithm is applicable to large scale problems and scale well with problem size.

Proposition 1 *If the entries of a symmetric invertible matrix $G(\mathbf{p})$ and a vector $\mathbf{f}(\mathbf{p})$ are affine functions of $\mathbf{p}$ and $G(\mathbf{p}) \succeq 0$ for $\mathbf{p} \in \mathscr{D}$ where $\mathscr{D}$ is a convex set, then $\mathbf{f}(\mathbf{p})^T G(\mathbf{p})^{-1}\mathbf{f}(\mathbf{p})$ is convex.*

Please refer to Appendix A for the proof. This proposition shows that for $\mathbf{p} \in D$, E is a concave function of $\mathbf{p}$ since $P\mathbf{1} \geq \mathbf{0}$ and consequently $J^T(\text{diag}(P\mathbf{1})\otimes I_d)J \succeq 0$, where D denotes the feasible region of $\mathbf{p}$, as determined by (3). Analogous to [14], we have the following proposition:

Proposition 2 *There exists a minimum binary solution of E under the constraints defined in* (3).

[1] We omit the constant $\boldsymbol{\theta}_0^T H\boldsymbol{\theta}_0$ since it doesn't affect the optimization.

[2] This is different from the conventional definition.

Proof It's known that the minimum solution of a concave function over a polytope can be taken at one of its vertices. The proposition follows by combining this result with the total unimodularity of D as stated previously.

5. Optimization

Our analysis in the previous section indicates that the nonlinear part of E is determined by a low dimensional variable $\mathbf{u}$ and is concave for $\mathbf{p} \in D$. Therefore it's natural to use the normal simplicial algorithm [10], a BnB algorithm specifically designed for concave functions, to optimize E.

5.1. Initial bounding region

In the normal simplicial algorithm, simplexes are used to construct the convex envelopes of a concave function. Therefore the initial bounding region should be chosen as a simplex or a collection of simplexes.

We use a collection of simplexes to bound the feasible region $U \triangleq \{\mathbf{u}|\mathbf{u} = Q^T\mathbf{p}, \mathbf{p} \in D\}$ as the resulting bounding could be more tight. The procedure is as follows. We first choose an interior point $\mathbf{v}_0$ of U. $\mathbf{v}_0$ can be chosen as $Q^T\frac{s}{mn}\mathbf{1}_{mn}$, which corresponds to the fuzziest point correspondence. We then construct a new coordinate system by shifting the origin of the coordinate system of $\mathbf{u}$ to $\mathbf{v}_0$, as illustrated in Fig. 1. We now construct each bounding simplex as the intersection of an orthant of the new coordinate system with a half space containing U, whose face, i.e., a plane, supports U and has a normal vector $\mathbf{h}$ chosen as the normalized mean of the orthant axes. Fig. 1 illustrates the bounding simplex construction process.

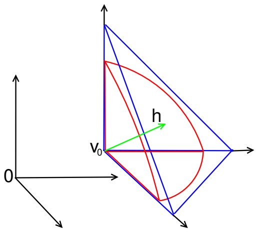

Figure 1. Red region: the intersection of U with an orthant of the new coordinate system. Blue region: the bounding simplex. The normal vector $\mathbf{h}$ of the supporting plane is chosen as the normalized mean of the orthant axes.

The distance from $\mathbf{v}_0$ to the supporting plane with normal $\mathbf{h}$ can be computed as:

$$\max\{\mathbf{h}^T(Q^T\mathbf{p} - \mathbf{v}_0)|\mathbf{p} \in D\}$$

This is a k-cardinality linear assignment problem which can be either directly solved [7] or transformed into a standard linear assignment problem [23] (we adopt the latter approach and choose the Jonker-Volgenant algorithm [12] for the resulting problem in this paper). The supporting plane with normal $\mathbf{h}$ can then be completely determined. In turn, the vertices of the bounding simplex can be recovered which has $\mathbf{v}_0$ as one of its vertices.

5.2. Choice of H

Although for feasible $\mathbf{u} \in U$, we have $\mathrm{mat}((R^T\mathbf{u})_{1,\dots,l}) \succeq 0$ and consequently E is concave according to Proposition 1, this may not be the case for infeasible $\mathbf{u}$, which can happen at one of the vertices of the bounding simplexes. Therefore, to ensure the concavity of E over all the bounding simplexes, we let the eigenvalues of H be large enough so that $\mathrm{mat}((R^T\mathbf{u})_{1,\dots,l}) + H \succeq 0$. The procedure is as follows. Assume that the eigenvalues of $\mathrm{mat}((R^T\mathbf{v}_i)_{1,\dots,l})$ constitute a vector $\boldsymbol{\lambda}_i$, where $\mathbf{v}_i$ is a vertex of one of the bounding simplexes. We choose a vector $\boldsymbol{\lambda}_0$ such that its j-th element $\lambda_0^j = \min\{\min_i \lambda_i^j, 0\}$. We then set $H = -\mathrm{diag}(\boldsymbol{\lambda}_0)$, and we now have $\mathrm{mat}((R^T\mathbf{v}_i)_{1,\dots,l}) + H \succeq 0$. Based on the convexity of the spectrahedra $\{\mathbf{u}|\mathrm{mat}((R^T\mathbf{u})_{1,\dots,l}) + H \succeq 0\}$, we have $\mathrm{mat}((R^T\mathbf{u})_{1,\dots,l}) + H \succeq 0$ for any $\mathbf{u}$ that belongs to the bounding simplexes. Based on Proposition 1, it follows that E is concave over the bounding simplexes.

5.3. Lower bounds

The convex envelope E_{CS} of the concave part $E_C(\mathbf{u})$ of E over a simplex $S = [\mathbf{v}_1, \dots, \mathbf{v}_{l+t+1}]$ is the unique affine function which coincides with E_C at the vertices $\mathbf{v}_1, \dots, \mathbf{v}_{l+t+1}$ [10], i.e., $E_{CS}(\mathbf{u}) = \sum_{i=1}^{l+t+1} \alpha_i E_C(\mathbf{v}_i)$ with $\mathbf{u} = \sum_{i=1}^{l+t+1} \alpha_i \mathbf{v}_i$, $\sum_{i=1}^{l+t+1} \alpha_i = 1$, $\alpha_i \geq 0$, $\forall i$. Based on this result, the lower bound of E for region $D \cap S$ can be obtained as the optimal value of the following linear program:

$$\begin{aligned} &\min_{\alpha_i, \mathbf{p}} \sum_{i=1}^{l+t+1} \alpha_i E_C(\mathbf{v}_i) + (\mathbf{1}_m^T \otimes \mathbf{z}^T)\mathbf{p} \\ &s.t. \sum_{i=1}^{l+t+1} \alpha_i \mathbf{v}_i = Q^T\mathbf{p}, \sum_{i=1}^{l+t+1} \alpha_i = 1, \alpha_i \geq 0, \forall i, \mathbf{p} \in D \end{aligned} \tag{9}$$

By tweaking this linear program, in Sec. 5.6, we will propose an alternative more efficient scheme to compute the lower bound.

5.4. Division of a simplex

Since the BnB algorithm is used for optimization, during the branching phase, a chosen simplex needs to be subdivided into several smaller simplexes. We adopt the following simple strategy to divide a simplex. For a chosen simplex, the longest edge is bisected. This results in two sub-simplexes. It has been proved that such a subdivision scheme leads to a BnB algorithm that converges [10].

5.5. The normal simplicial BnB algorithm

We now describe the algorithm for minimizing E. During initialization, a set of simplexes whose union contains the feasible region D is computed. Then in each iteration, the simplex yielding the lowest lower bound among all the simplexes is further subdivided so as to improve the lower bound of E for D. Meanwhile, the upper bound is updated by evaluating E with solutions of the linear programs used to compute the lower bounds. The pseudo-code of the algorithm is summarized in Algorithm 1.

Algorithm 1: The normal simplicial algorithm for minimizing E

```
Initialization
Select tolerance error $\epsilon > 0$.
Find a collection of simplexes $\{S_i\}$ such that
$U \subset \cup_i S_i$ according to Sec. 5.1. Choose $\boldsymbol{\lambda}_0$ according
to Sec. 5.2 and set $H = -\mathrm{diag}(\boldsymbol{\lambda}_0)$. Set
$\mathscr{M}_1 = \mathscr{N}_1 = \{S_i\}$, where $\mathscr{M}_1$ denotes the collection
of all rectangles and $\mathscr{N}_1$ denotes the collection of
active rectangles.
for $k = 1, 2, \dots$ do
    For each simplex $S \in \mathscr{N}_k$, solve the linear
    program (9) to obtain a basic optimal solution
    $\boldsymbol{\omega}(S)$ and the optimal value $\beta(S)$. $\beta(S)$ is the
    lower bound of $E$ for region $D \cap S$.
    Let $\mathbf{p}^k$ equal to the best among all feasible
    solutions so far encountered: $\mathbf{p}^{k-1}$ and all
    $\boldsymbol{\omega}(S), S \in \mathscr{N}_k$. Delete all simplexes $S \in \mathscr{M}_k$ such
    that $\beta(S) \geq E(\mathbf{p}^k) - \epsilon$. Let $\mathscr{R}_k$ be the remaining
    collection of simplexes.
    If $\mathscr{R}_k = \emptyset$, terminate: $\mathbf{p}^k$ is the global $\epsilon$-minimal
    solution. Otherwise, go to the next step.
    Select the simplex to be divided:
    $S_k \in \arg\min\{\beta(S) | S \in \mathscr{R}_k\}$.
    Divide $S_k$ according to Sec. 5.4 to get two
    sub-simplexes $S_{k1}$ and $S_{k2}$.
    Let $\mathscr{N}_{k+1} = \{S_{k1}, S_{k2}\}$ and
    $\mathscr{M}_{k+1} = (\mathscr{R}_k \backslash \{S_k\}) \cup \mathscr{N}_{k+1}$.
end
```

Our algorithm is an instance of the BnB technique, hence its worst case time complexity is exponential. Nevertheless, this problem is much alleviated by the low dimension $t + l$ of the space of $\mathbf{u}$, to which the BnB algorithm is applied.

5.6. A new efficient lower bounding scheme

Our algorithm is an instance of the BnB technique, and it consists of three subproblems: branching, finding the upper bound and finding the lower bound. It's obvious that the lower bounding problem (9) costs much more than the other two subproblems since it is a generic linear program, for which there is no efficient algorithm. In this subsection, we propose an alternative lower bounding scheme which is more efficient to solve while at the same time retaining the tightness of the original lower bounding problem.

To achieve the above goal, a natural idea is to drop the inequality constraints $\alpha_i \geq 0, \forall i$ in (9), and then there are only linear equality constraints on α_i. Hence α_i can be eliminated via algebraic substitution and we arrive at the following equivalent problem:

$$\begin{aligned}\min E_S(p) = {} & ([E_C(\mathbf{v}_1), \dots, E_C(\mathbf{v}_{l+t})] - E_C(\mathbf{v}_{l+t+1})\mathbf{1}_{l+t}^T) \\ & \cdot ([\mathbf{v}_1, \dots, \mathbf{v}_{l+t}] - \mathbf{v}_{l+t+1}\mathbf{1}_{l+t}^T)^{-1}(Q^T\mathbf{p} - \mathbf{v}_{l+t+1}) \\ & + E_C(\mathbf{v}_{l+t+1}) + (\mathbf{1}_m^T \otimes \mathbf{z}^T)\mathbf{p} \\ s.t. \quad & \mathbf{p} \in D \end{aligned} \tag{10}$$

Problem (10) is a k-cardinality linear assignment problem which can be very efficiently solved by the combinatorial optimization algorithms mentioned in Sec. 5.1. Note that simplex S will not degenerate throughout the BnB iterations and therefore matrix $[\mathbf{v}_1, \dots, \mathbf{v}_{l+t}] - \mathbf{v}_{l+t+1}\mathbf{1}_{l+t}^T$ is always invertible. We have the following proposition.

Proposition 3 *The optimal value of problem* (10) *is a lower bound of E for region $D \cap S$.*

Proof Problem (10) is a relaxed version of (9) by dropping the constraints $\alpha_i \geq 0, \forall i$. Therefore the optimal value of (10) will not be greater than that of (9), whereas solving (9) yields a lower bound of E for region $D \cap S$.

What remains is to show that the lower bound computed via (10) is tight enough. In iteration k of our algorithm, the lowest one among all the lower bounds corresponding to simplexes in $\mathscr{M}_k$ is chosen as the lower bound of E for the feasible region D. Hence only the lowest lower bound determines the quality of a bounding scheme. W.l.o.g, let us assume that S is the simplex yielding the lowest lower bound by using (10) to compute the lower bound of E for region D. It's apparent that there are two possibilities for the location of the optimal solution $\mathbf{p}^*$ of problem (10): $\mathbf{p}^* \in \{\mathbf{p} | E_S(\mathbf{p}) \leq E(\mathbf{p})\}$ or $\mathbf{p}^* \in \{\mathbf{p} | E_S(\mathbf{p}) > E(\mathbf{p})\}$, as illustrated in Fig. 2. Note that the latter case is impossible since in this case, $E_S(\mathbf{p}^*)$ will be strictly larger than the minimum value of E over D, violating the assumption that $E_S(\mathbf{p}^*)$ is a lower bound of E for region D. Hence it can only happen that $\mathbf{p}^* \in \{\mathbf{p} | E_S(\mathbf{p}) \leq E(\mathbf{p})\}$. From Fig. 2, we can see that $\{\mathbf{p} | E_S(\mathbf{p}) \leq E(\mathbf{p})\}$ is an ellipse-like region containing and inscribing the simplex S. Therefore, if the length of the longest edge of S shrinks to zero, the lower bounds computed via (9) and via (10) will be close to each other. This is the case with our algorithm since the simplex yielding the lowest lower bound is selected to be further subdivided in each BnB iteration and thus it becomes smaller and smaller with the increase of iteration. Hence problem (10) can serve as a good candidate for computing the lower bound of E for D in our algorithm.

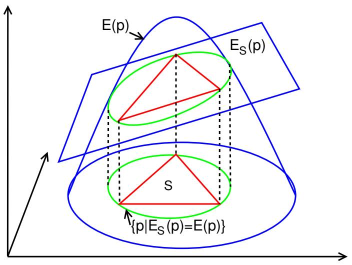

Figure 2. Ellipse-like closed curve $\{\mathbf{p}|E_S(\mathbf{p}) = E(\mathbf{p})\}$ separates the solution space into two disjoint regions. Simplex S is contained in the region $\{\mathbf{p}|E_S(\mathbf{p}) \leq E(\mathbf{p})\}$.

6. Experimental results

We implement our method under the Matlab 8.1 environment and compare it with RPM [6], CPD [17] and MG [11], whose source codes are freely available. These methods represent state-of-the-arts, only utilize the point position information for matching, and are capable of handling outliers in both point sets. Affine transformation is used for RPM. Rigid transformation is used for CPD and MG (other types of transformations are found to be far less robust for the types of experiments conducted in this paper). We set $\epsilon = 0.1$ for our method (the data point set is assumed to be unit sized). All the methods are run on a PC with 2.4 GHz CPU and 8G RAM.

6.1. Estimation of parameter s

The proposed algorithm needs to know s, the number of matches, in advance. This number, however, is unknown in practice. We propose the following simple method to estimate s.

By using some feature descriptor (shape context [2] is used in this paper), we could get the cost $d_{i,j}$ of matching two points $i \in \mathscr{X}$ and $j \in \mathscr{Y}$. We then compute the matching cost for each model point as $d_i^{\mathscr{X}} = \min_j d_{i,j}$. Then, we use the k-means clustering algorithm to separates the set $\{d_i^{\mathscr{X}}\}$ into 2 clusters and choose the cluster with smaller mean value (which more likely corresponds to the inliers in $\mathscr{X}$). Let $s_{\mathscr{X}}$ be the cardinality of this cluster. By applying the same procedure to the data point set, we get another cardinality $s_{\mathscr{Y}}$. Finally, we set $s = \min(s_{\mathscr{X}}, s_{\mathscr{Y}})$.

To validate if the proposed estimation method of s is effective, let's compare the point matching performance by using the estimated s and the ground truth s. The experiment settings are described in Sec. 6.2. Fig. 3 shows the values of the ground truth s and the corresponding values of estimated s. We can see that the estimated s is in general smaller than the ground truth s. In Fig. 4, we compare the matching errors in different experiments by using the ground truth s and the estimated s. We can see that our method's performance does not degrade much (particularly for similarity transformation) by using the estimated s in comparison with using the ground truth s.

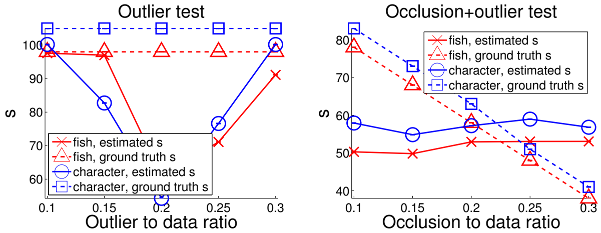

Figure 3. Estimated s and ground truth s for the tests described in Sec. 6.2.

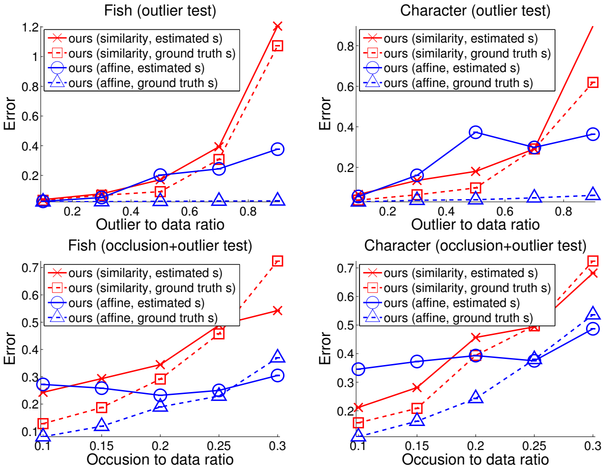

Figure 4. Matching errors by our method with estimated s and ground truth s for the tests described in Sec. 6.2.

In all our experiments in Sec. 6.2, we use the estimated s as the input to our algorithm.

6.2. Experiments on benchmark dataset

We use the benchmark Chui-Rangarajan synthesized datasets [6] to evaluate the 4 competing methods. More experiments on 3D synthetic data can be found in the supplementary materials. Examples of the data and model point sets are shown in Fig. 5. We consider the 2D similarity and affine transformations between two point sets. For 2D similarity transformation, we let $\boldsymbol{\theta} = \left[\theta_1, \ldots, \theta_4\right]^T$. Here $[\theta_3, \theta_4]^T$ are translation parameters and $\theta_1 = r\cos(\phi)$ and $\theta_2 = r\sin(\phi)$, where r represents scale and ϕ is rotation angle. Then we have $J(\mathbf{x}_i) = \begin{bmatrix} x_i^1 & -x_i^2 & 1 & 0 \\ x_i^2 & x_i^1 & 0 & 1 \end{bmatrix}$. It can be verified that the rows of $B_2 = B([1,3,4,11],:)$ constitute the nonzero unique rows of B.

For 2D affine transformation, we let $\boldsymbol{\theta} = \left[\theta_1, \ldots, \theta_6\right]^T$ with $[\theta_1, \ldots, \theta_4]^T$ being the linear part of the transformation and $[\theta_5, \theta_6]^T$ being the translation. Then we have

$J(\mathbf{x}_i) = \begin{bmatrix} x_i^1 & x_i^2 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i^1 & x_i^2 & 0 & 1 \end{bmatrix}$. It can be verified that the rows of $B_2 = B([1, 2, 5, 8, 11, 29], :)$ constitute the nonzero unique rows of B.

Two categories of tests are used to evaluate the performances of different methods against outliers. 1) **Outlier test**. Equal number of normally distributed random outliers are added to different sides of the prototype shape to generate the two point sets, respectively, as illustrated in columns 2, 3 of Fig. 5. 2) **Occlusion + Outlier test**. First, equal degree of occlusions are applied to the prototype shape to generate the two point sets, respectively. We simulate occlusion by first finding the shortest Hamiltonian circle of the prototype point set (via solving a traveling salesman problem) and then retaining a segment of the circle starting at a random point. Then, a fixed number of normally distributed random outliers (outlier to data ratio is fixed to 0.5) are added to different sides of the two point sets, as illustrated in columns 4, 5 of Fig. 5. For all the above tests, a moderate amount of nonrigid deformation is applied to the data point set. Two shapes [6], a fish and a character, as shown in the left column of Fig. 5, are used as the prototype shapes, respectively.

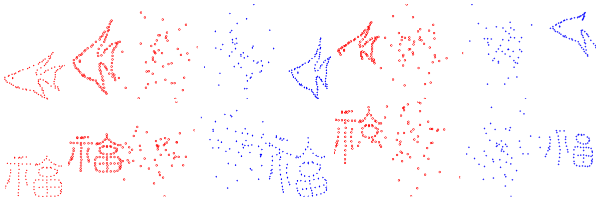

Figure 5. Left column: the prototype shapes. For the rest columns: examples of model and data point sets in the outlier (columns 2, 3) and occlusion + outlier (columns 4, 5) tests.

The average matching errors over 100 random trials by different methods in the 2 categories of tests are shown in Fig. 6, where error is defined as mean of the Euclidean distances between the warped ground truth model inliers and their corresponding data inliers. It can be seen that our method performs much better than other methods. This demonstrates our method's robustness to outliers. Examples of matching results are shown in Fig. 7.

The average running times of different methods are listed in Table 1. It is not surprising that our method has longer running time than other methods since our method is global optimal, while others are heuristics based. Our method runs much faster in the case of similarity transformation than affine transformation. This is because affine transformation has more parameters, thus requiring more time to converge.

7. Conclusion

We proposed a globally optimal point matching algorithm capable of handling outliers in both point sets. This algorithm is a non-trival extension of the concave optimization approach in [14] by relaxing the condition that each model point has a counterpart in data point set. We also proposed a novel lower bounding scheme to make our algorithm efficient. Our experimental results showed that the proposed method outperforms state-of-the-art methods in robustness to outliers with competitive time efficiency.

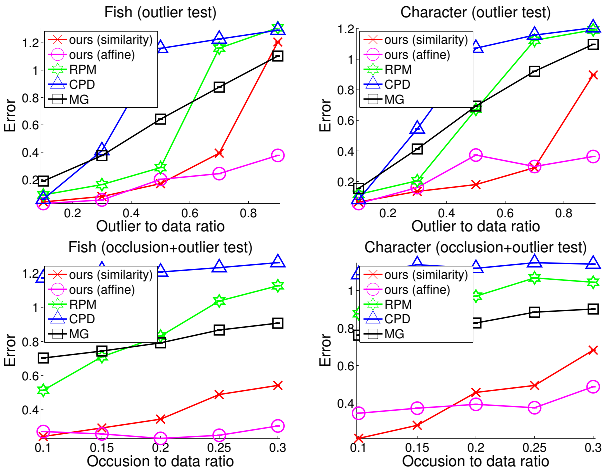

Figure 6. Average matching errors over 100 random trials by different methods for the 2 categories of tests.

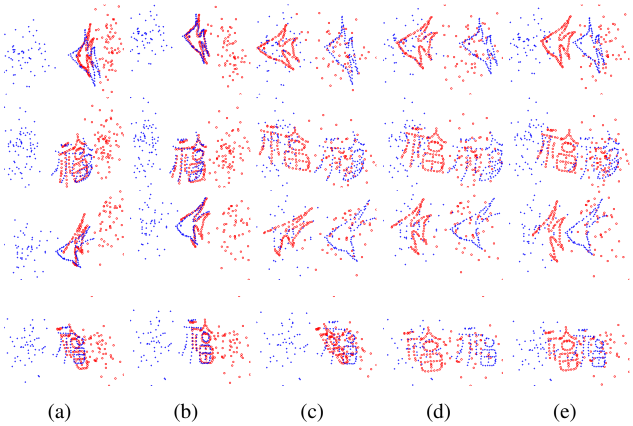

Figure 7. Examples of matching results by different methods in the outlier (top 2 rows) and occlusion + outlier (bottom 2 rows) tests. (a) Ours (similarity); (b) ours (affine); (c) RPM; (d) CPD; and (e) MG.

Table 1. Average running times of different methods (in seconds)

	Outliers	Occlusion+Outliers
ours (similarity)	3.0564	2.0102
ours (affine)	34.2416	18.1863
RPM	1.1405	0.9588
CPD	0.0963	0.0924
MG	0.2265	0.1982

A. Proof of Proposition 1

Since $\mathbf{f}(\mathbf{p})^T G(\mathbf{p})^{-1}\mathbf{f}(\mathbf{p})$ is a smooth function of $\mathbf{p}$, to prove it's convex, it suffices to show that its Hessian matrix is positive semidefinite.

By taking its partial derivative w.r.t. $\mathbf{p}$, we get the Jacobian vector of $\mathbf{f}(\mathbf{p})^T G(\mathbf{p})^{-1}\mathbf{f}(\mathbf{p})$ as

$$\mathbf{J}_{ac} = \begin{bmatrix} 2\mathbf{f}^T G^{-1} & -(\mathbf{f}^T G^{-1} \otimes \mathbf{f}^T G^{-1})G \end{bmatrix} \begin{bmatrix} \frac{d\,\mathbf{f}}{d\,\mathbf{p}} \\ \frac{d\,\mathrm{vech}(G)}{d\,\mathbf{p}} \end{bmatrix}$$

where the operator 'vech' denotes converting the upper or lower portion (including the diagonal line) of a symmetric matrix into a vector [8]. The constant matrix $G \triangleq \frac{d\,\mathrm{vec}(G)}{d\,\mathrm{vech}(G)}$. Matrices $\frac{d\,\mathbf{f}}{d\,\mathbf{p}}$ and $\frac{d\,\mathrm{vech}(G)}{d\,\mathbf{p}}$ are constant matrices since the entries of $\mathbf{f}(\mathbf{p})$ and $G(\mathbf{p})$ are affine functions of $\mathbf{p}$.

By taking the partial derivative of $\mathbf{J}_{ac}$ w.r.t. $\mathbf{p}$, we get the Hessian matrix of $\mathbf{f}(\mathbf{p})^T G(\mathbf{p})^{-1}\mathbf{f}(\mathbf{p})$ as

$$\begin{aligned} H_{es} = 2 &\begin{bmatrix} (\frac{d\,\mathbf{f}}{d\,\mathbf{p}})^T & (\frac{d\,\mathrm{vech}(G)}{d\,\mathbf{p}})^T \end{bmatrix} \\ \cdot &\begin{bmatrix} G^{-1} & -(\mathbf{f}^T G^{-1} \otimes G^{-1})G \\ -G^T(G^{-1}\mathbf{f} \otimes G^{-1}) & G^T((G^{-1}\mathbf{f}\mathbf{f}^T G^{-1}) \otimes G^{-1})G \end{bmatrix} \\ \cdot &\begin{bmatrix} \frac{d\,\mathbf{f}}{d\,\mathbf{p}} \\ \frac{d\,\mathrm{vech}(G)}{d\,\mathbf{p}} \end{bmatrix} \end{aligned}$$

If $G \succeq 0$, then we have $G^{-1} \succeq 0$ and $G^{-1} = G^{-\frac{1}{2}} G^{-\frac{1}{2}}$. Substituting this result into H_{es}, we get

$$\begin{aligned} H_{es} =& 2 \begin{bmatrix} (\frac{d\,\mathbf{f}}{d\,\mathbf{p}})^T & (\frac{d\,\mathrm{vech}(G)}{d\,\mathbf{p}})^T \end{bmatrix} \begin{bmatrix} G^{-1/2} \\ -G^T(G^{-1}\mathbf{f} \otimes G^{-1/2}) \end{bmatrix} \\ & \cdot \begin{bmatrix} G^{-1/2} & -(\mathbf{f}^T G^{-1} \otimes G^{-1/2})G \end{bmatrix} \begin{bmatrix} \frac{d\,\mathbf{f}}{d\,\mathbf{p}} \\ \frac{d\,\mathrm{vech}(G)}{d\,\mathbf{p}} \end{bmatrix} \end{aligned}$$

i.e., H_{es} can be factored as the product of a matrix and its transpose. Therefore we have $H_{es} \succeq 0$. This implies that $\mathbf{f}(\mathbf{p})^T G(\mathbf{p})^{-1}\mathbf{f}(\mathbf{p})$ is convex.

Acknowledgements

This work was supported by Natural Science Foundation of Shanxi, China (grant No. 2012021015-2).

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. *TPAMI*, 35(7):1565–1576, 2013.

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

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

[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.

[5] T. Chen, B. C. Vemuri, A. Rangarajan, and S. J. Eisenschenk. Group-wise point-set registration using a novel cdf-based havrda-charvt divergence. *IJCV*, 86(1):111–124, 2010.

[6] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, 2003.

[7] M. Dell'Amicoa, A. Lodib, and S. Martello. Efficient algorithms and codes for k-cardinality assignment problems. *Discrete Applied Mathematics*, 110:25–40, 2001.

[8] P. L. Fackler. Notes on matrix calculus, 2005. http://www4.ncsu.edu/∼pfackler/MatCalc.pdf .

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

[10] R. Horst and H. Tuy. *Global Optimization, Deterministic Approaches*. Springer-Verlag, 1996.

[11] 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.

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

[13] H. Li and R. Hartley. The 3d-3d registration problem revisited. In *ICCV*, pages 1–8, 2007.

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

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

[16] J. Maciel and J. Costeira. A global solution to sparse correspondence problems. *TPAMI*, 25(2):187–199, 2003.

[17] A. Myronenko and X. Song. Point set registration: Coherent point drift. *TPAMI*, 32(12):2262–2275, 2010.

[18] C. Olsson, F. Kahl, and M. Oskarsson. Branch-and-bound methods for euclidean registration problems. *TPAMI*, 31(5):783–794, 2009.

[19] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover Publications, INC. Mineola. New York, 1998.

[20] 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.

[21] M. Sofka, G. Yang, and C. V. Stewart. Simultaneous covariance driven correspondence (cdc) and transformation estimation in the expectation maximization framework. In *CVPR*, pages 1–8, 2007.

[22] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *ECCV*, pages 558–569, 2004.

[23] A. Volgenant. Solving the k-cardinality assignment problem by transformation. *European Journal of Operational Research*, 157:322–331, 2004.