Available online at www.isr-publications.com/jnsa
J. Nonlinear Sci. Appl., 10 (2017), 5540–5551
Research Article

ISSN: 2008-1898

Journal of Nonlinear Sciences and Applications

Journal Homepage: www.tjnsa.com - www.isr-publications.com/jnsa

Minimizing the object space error for pose estimation: towards the most efficient algorithm

Yingwei Han[a], Yong Xia[b], Ping Li[a,*]

[a] School of Economics and Management, Beihang University, Beijing, 100191, China.

[b] School of Mathematics and System Sciences, Beihang University, 100191, China.

Communicated by R. Saadati

Abstract

In this paper, we present an efficient branch-and-bound algorithm to globally minimize the object space error for the camera pose estimation. The key idea is to reformulate the pose estimation model using the optimal Lagrangian multipliers. Numerical simulation results show that our algorithm usually terminates in the first iteration and finds an ϵ-suboptimal solution. Furthermore, the efficiency of our algorithm is demonstrated by a comprehensive numerical comparison with two well-known heuristics. We also demonstrate the computational power of our algorithm by comparing it with the state-of-the-art global optimization package BARON.

Keywords: Pose estimation, PnP, robotics, branch-and-bound, Lagrangian dual.
2010 MSC: 90C26, 90C30, 65D19.

1. Introduction

Pose estimation, also known as the Perspective-n-Point problem (PnP), is to estimate the pose of the camera based on the given 3D reference points and their associated 2D images [12]. It is one of the important problems in computer vision, photogrammetry and robotics.

In general, solution methods for solving pose estimation can be divided into the following three groups:

The first group is composed of the iterative local search methods ([15, 20, 21, 24]). The orthogonal iteration (OI) algorithm [15] may be the most efficient. The basic idea of OI is to minimize the object space error by alternatively minimizing the estimation of the rotation matrix and the translation vector. Starting from a proper initialization, the OI algorithm often fast converges to a high-accuracy global minimizer. But if it is poorly initialized, OI could get trapped in a local minimizer.

The second group is made up of the iterative global optimization methods. Agarwal et al. [1] proposed a branch-and-bound algorithm to solve the triangulation and camera pose estimation, where the objective function is fractional. The lower bounding approach is to solve the second-order cone programming (SOCP) relaxation, by noting that a single fraction t/s bounded with s and t can be rewritten as an SOCP

*Corresponding author
Email addresses: hanyingwei@buaa.edu.cn (Yingwei Han), yxia@buaa.edu.cn (Yong Xia), liping124@buaa.edu.cn (Ping Li)

doi:10.22436/jnsa.010.10.34

Received 2017-05-16

[26]. This algorithm was further employed in [18] to minimize the image space error, where the rotation matrix was parameterized by quaternion. Hartley and Kahl [9, 10] developed a branch-and-bound method to minimize the ℓ_∞ norm of the tangent of the angle error, based on SOCP relaxation. Though providing a solution of proven global optimality, the branch-and-bound methods are of limited application in practice because of their high computational complexity. For example, the average running time reported in [10] is 1.5 minutes for 10 reference points.

The third group consists of the non-iterative methods ([2, 7, 11, 14, 17, 19, 23, 28]). The pose estimation problem is first reformulated to a single (large) equation system. Then the system is approximately solved in order to gain speed. Recently, Schweighofer and Pinz [23] proposed the semidefinite relaxation (SDR) approach by lifting the quaternion model of the pose estimation problem. Moreover, to perform better, the standard SDR was refined to a heuristic version with two well-chosen parameters, see [23]. In practice, SDR often gives a solution close to the global minimizer, even for small number of points and large noise. The limitation is that the accuracy of the solution obtained by SDR is lower than that of OI.

To our knowledge, the branch-and-bound algorithm for minimizing the object space error (which is the same cost function as in OI and SDR) has not been studied in literature. Suppose now we directly employ the branch-and-bound algorithm developed in [1] to minimize the object space error, at first we have to introduce much more additional variables to linearize the cost function, which certainly is far from efficient. In this paper, we observe that the object space error is already a convex quadratic function. It motivates us to develop a new branch-and-bound method based on quadratic programming (QP) relaxation. To improve the efficiency, we establish a tighter Lagrangian reformulation of the quadratic object space error. Surprisingly, the numerical simulation results show that the new branch-and-bound algorithm usually terminates in the first iteration and returns an ϵ-suboptimal solution. Finally, a comprehensive numerical comparison demonstrates that the new branch-and-bound algorithm outperforms OI, SDR and the global optimization package BARON.

The remainder of the paper is organized as follows. In Section 2, we reformulate the pose estimation problem based on Lagrangian dual theory. Section 3 presents the new branch-and-bound algorithm in detail. In Section 4, we first numerically compare the efficiency of the branch-and-bound algorithm for solving four different reformulations. Then we compare the accuracy of the solutions obtained by the branch-and-bound algorithm, OI and SDR, respectively. Finally, we compare the efficiency of our new branch-and-bound algorithm and the commercial global optimization package BARON. Concluding remarks are given in Section 5.

2. Formulations and relaxations

2.1. Problem formulation

Given a set of 3D reference points p_i, $i = 1, 2, \cdots, n$, $(n \geqslant 3)$ in the object coordinate system and the associated normalized 2D image projections v_i in the camera coordinate system, we minimize the following object space error [22]

$$\min_{R\in S(3),t} \left\{ E(R,t) = \sum_{i=1}^{n} \|(I-\widehat{V}_i)(Rp_i+t)\|^2 \right\}, \tag{2.1}$$

where $S(3)$ is the set of 3×3 orthogonal matrices, I is the 3×3 identity matrix, $\|\cdot\|$ is the standard ℓ_2-norm, and

$$\widehat{V}_i = \frac{\widehat{v}_i\widehat{v}_i^T}{\widehat{v}_i^T\widehat{v}_i},$$

is the line-of-sight projection matrix along $\widehat{v}_i = (u_i, v_i, 1)^T$. Since (2.1) is an unconstrained quadratic program in terms of t, by setting the partial gradient of (2.1) with respect to t equal to zero

$$\frac{\partial}{\partial t}E(R,t) = \sum_{i=1}^{n} 2\{(I-\widehat{V}_i)(Rp_i+t)\} = 0,$$

we can get the optimal translation vector [15, 23]:

$$t_{\mathrm{opt}} = -\left(\sum_{i=1}^{n} Q_i\right)^{-1} \sum_{i=1}^{n} (Q_i R p_i), \tag{2.2}$$

where

$$Q_i = (I - \widehat{V}_i)^T (I - \widehat{V}_i) = I - \widehat{V}_i.$$

As in [23], define the following operators for the 3D vector p and 3×3 matrix R, respectively,

$$C(p) = \begin{bmatrix} p^T & 0_{1\times 3} & 0_{1\times 3} \\ 0_{1\times 3} & p^T & 0_{1\times 3} \\ 0_{1\times 3} & 0_{1\times 3} & p^T \end{bmatrix}, \quad r(R) = \begin{bmatrix} r_1^T \\ r_2^T \\ r_3^T \end{bmatrix},$$

where $0_{1\times 3}$ is a zero matrix of size 1×3 and $R = \left[r_1^T \; r_2^T \; r_3^T\right]^T$. Now we can rewrite (2.2) as

$$t_{\mathrm{opt}} = T_{3\times 9} \cdot r, \tag{2.3}$$

where

$$T_{3\times 9} = -\left(\sum_{i=1}^{n} Q_i\right)^{-1} \sum_{i=1}^{n} (Q_i C(p_i)).$$

Substituting (2.3) into (2.1) and rearranging the formulation yields the following simple model

$$\min_{R \in S(3)} \left\{ f_1(R) := r(R)^T M r(R) \right\}, \tag{2.4}$$

where

$$M = \sum_{i=1}^{n} \left((C(p_i) + T_{3\times 9})^T \, Q_i \, (C(p_i) + T_{3\times 9}) \right).$$

2.2. *Problem relaxations and reformulations*

It is easy to verify that

$$R \in S(3) \iff R^T R = I \iff R R^T = I.$$

Then (2.4) has the following three quadratic constrained quadratic programming (QCQP) reformulations

$$\min_{R^T R = I} r(R)^T M r(R), \tag{2.5}$$

$$\min_{R R^T = I} r(R)^T M r(R), \tag{2.6}$$

$$\min_{R^T R = I, R R^T = I} r(R)^T M r(R), \tag{2.7}$$

where the idea to add two redundant constraints in (2.7) is not new, see for example, [3, 27].

For QCQP, Lagrangian dual often provides a high-quality lower bound for the primal problem. We first present the Lagrangian dual of (2.7). Let S, T be two symmetric matrices of size 3×3, respectively. The Lagrangian function of (2.7) is

$$\begin{aligned} L(r(R), S, T) &= r(R)^T M r(R) - \mathrm{tr}((R^T R - I)S) - \mathrm{tr}((R R^T - I)T) \\ &= r(R)^T (M - I \otimes S - T \otimes I) r(R) + \mathrm{tr}(S + T), \end{aligned}$$

where $\mathrm{tr}(A)$ denotes the trace of the matrix A (i.e., the sum of all the diagonal entries of A), $A \otimes B$ denotes the Kronecker product of A and B, i.e., $A \otimes B = [A_{ij}B]$. Then the dual function reads

$$\begin{aligned} d(S,T) &= \min_{r(R)} L(r(R),S,T) \\ &= \begin{cases} \operatorname{tr}(S+T), & \text{if } M - I\otimes S - T\otimes I \succeq 0, \\ -\infty, & \text{otherwise,} \end{cases} \end{aligned}$$

where $A \succeq 0$ denotes that A is positive semidefinite. Now, the Lagrangian dual problem is

$$\max_{S=S^T,T=T^T} \{d(S,T)\} = \max_{M-I\otimes S-T\otimes I\succeq 0, S=S^T, T=T^T} \operatorname{tr}(S+T). \tag{2.8}$$

We similarly write the Lagrangian dual problems of (2.5) and (2.6) as follows:

$$\max_{M-I\otimes S\succeq 0, S=S^T} \operatorname{tr}(S), \tag{2.9}$$

$$\max_{M-T\otimes I\succeq 0, T=T^T} \operatorname{tr}(T). \tag{2.10}$$

The above three dual problems are all semidefinite programming (SDP) problems. They can be globally solved by the publicly available optimization tools SeDuMi [25]. Denote the optimization solutions to (2.8), (2.9) and (2.10) by (S^*, T^*), S^{**} and T^{**}, respectively. Then we have the following results. The proofs can be found in Appendices A–D, respectively.

Proposition 2.1. *The three positive semidefinite matrices* $M - I\otimes S^* - T^*\otimes I$, $M - I\otimes S^{**}$ *and* $M - T^{**}\otimes I$ *are all singular.*

Proof. Denote by $\lambda_{\min}$ the minimal eigenvalue of $M - I\otimes S^* - T^*\otimes I$. Suppose $M - I\otimes S^* - T^*\otimes I$ is nonsingular. Then $\lambda_{\min} > 0$. Define $\widetilde{S} = S^* + \lambda_{\min}\cdot I$. We have

$$M - I\otimes \widetilde{S} - T^*\otimes I = (M - I\otimes S^* - T^*\otimes I) + \lambda_{\min} I\otimes I \succeq 0,$$

which implies that $(\widetilde{S}, T^*)$ remains feasible in (2.8). Since

$$\operatorname{tr}(\widetilde{S} + T^*) = \operatorname{tr}(S^* + T^*) + 3\lambda_{\min} > \operatorname{tr}(S^* + T^*),$$

we obtain a contradiction. The singularity of $M - I\otimes S^{**}$ and $M - T^{**}\otimes I$ can be similarly proved. □

Proposition 2.2.

$$\begin{aligned} \min_{R\in S(3)} f_1(R) &\geqslant \operatorname{tr}(S^* + T^*) \geqslant \max\{\operatorname{tr}(S^{**}), \operatorname{tr}(T^{**})\} \\ &\geqslant \min\{\operatorname{tr}(S^{**}), \operatorname{tr}(T^{**})\} \geqslant 0. \end{aligned}$$

Proof. The first inequality is due to the weak duality theory. The second inequality follows from the fact both $(S^{**}, 0_{3\times 3})$ and $(0_{3\times 3}, T^{**})$ are feasible solutions of (2.8). The last inequality holds since M is positive semidefinite and hence $0_{3\times 3}$ is feasible in both (2.9) and (2.10). □

Theorem 2.3. *The pose estimation problem* (2.4) *has the following three reformulations:*

$$\min_{R\in S(3)} \left\{ f_2(R) := r(R)^T (M - I\otimes S^{**}) r(R) + \operatorname{tr}(S^{**}) \right\}, \tag{2.11}$$

$$\min_{R\in S(3)} \left\{ f_3(R) := r(R)^T (M - T^{**}\otimes I) r(R) + \operatorname{tr}(T^{**}) \right\}, \tag{2.12}$$

$$\min_{R\in S(3)} \left\{ f_4(R) := r(R)^T (M - I\otimes S^* - T^*\otimes I) r(R) + \operatorname{tr}(S^* + T^*) \right\}. \tag{2.13}$$

Proof. For any $R \in S(3)$ and any symmetric matrices S and T, it holds that $\mathrm{tr}((R^TR - I)S) = 0$ and $\mathrm{tr}((RR^T - I)T) = 0$. It follows that

$$\begin{aligned} r(R)^T M r(R) &= r(R)^T M r(R) - \mathrm{tr}((R^TR - I)S) - \mathrm{tr}((RR^T - I)T) \\ &= r(R)^T(M - I \otimes S - T \otimes I)r(R) + \mathrm{tr}(S + T). \end{aligned}$$

The proof is complete by setting (S, T) as $(S^{**}, 0_{3\times3})$, $(0_{3\times3}, T^{**})$ and (S^*, T^*), respectively. □

Proposition 2.4. *Let $A \leqslant (<)B$ denote that $B - A$ is componentwise nonnegative (positive). For any given two 3×3 matrices $L < 0_{3\times3}$ and $U > 0_{3\times3}$, we have*

$$\min_{L\leqslant R\leqslant U} f_1(R) = 0, \tag{2.14}$$

$$\min_{L\leqslant R\leqslant U} f_2(R) = \mathrm{tr}(S^{**}), \tag{2.15}$$

$$\min_{L\leqslant R\leqslant U} f_3(R) = \mathrm{tr}(T^{**}), \tag{2.16}$$

$$\min_{L\leqslant R\leqslant U} f_4(R) = \mathrm{tr}(S^* + T^*), \tag{2.17}$$

where the minimums are attained at some scaled eigenvectors corresponding to the minimal eigenvalues of the Hessian matrices of f_1, f_2, f_3 and f_4, respectively.

Proof. The equality (2.14) holds since $M \succeq 0$ and $L < 0_{3\times3} < U$. Now we show (2.17). Since

$$M - I \otimes S^* - T^* \otimes I \succeq 0,$$

we have

$$r^T(M - I \otimes S^* - T^* \otimes I)r \geqslant 0, \quad \forall r.$$

According to Proposition 2.1, the minimal eigenvalue of $M - I \otimes S^* - T^* \otimes I$ is zero. Denote by $y \neq 0$ the corresponding normalized eigenvector. Then for any positive scalar $t > 0$, we have

$$(ty)^T(M - I \otimes S^* - T^* \otimes I)(ty) = 0.$$

Let $R(t)$ be such that $r(R(t)) = ty$. For sufficient small t, we have $L \leqslant R(t) \leqslant U$. Therefore, $R(t)$ solves

$$\min_{L\leqslant R\leqslant U} f_4(R).$$

The other equalities (2.15)-(2.16) are similarly proved. □

Propositions 2.2 and 2.4 imply that (2.13) is the tightest reformulation in view of relaxation.

3. A new branch-and-bound method

The branch-and-bound algorithm plays a great role in globally minimizing the nonconvex problems, see for example, [4]. It terminates with a certificate proving that the obtained solution is ϵ-suboptimal, by iteratively updating the upper and lower bounds on the optimal objective value. However, in general, the worst-case complexity of the branch-and-bound method grows exponentially with the problem size. For this purpose, we rewrite the pose estimation problem (2.4) as

$$\min_{(\alpha,\beta,\gamma)\in[0,2\pi]\times[0,\pi]\times[0,2\pi]} f_1(R(\alpha, \beta, \gamma)), \tag{3.1}$$

by observing there is a one-to-one mapping between the rotation matrices and the Euler angles

$$R(\alpha, \beta, \gamma) = \begin{bmatrix} R_{11} & R_{12} & \sin(\alpha)\sin(\beta) \\ R_{21} & R_{22} & -\cos(\alpha)\sin(\beta) \\ R_{31} & R_{32} & \cos(\beta) \end{bmatrix}, \tag{3.2}$$

where $R_{11} = \cos(\alpha)\cos(\gamma) - \cos(\beta)\sin(\alpha)\sin(\gamma)$, $R_{12} = -\cos(\beta)\cos(\gamma)\sin(\alpha) - \cos(\alpha)\sin(\gamma)$, $R_{21} =$

$\cos(\gamma)\sin(\alpha)+\cos(\alpha)\cos(\beta)\sin(\gamma)$, $R_{22} = \cos(\alpha)\cos(\beta)\cos(\gamma)-\sin(\alpha)\sin(\gamma)$, $R_{31} = \sin(\beta)\sin(\gamma)$ and $R_{32} = \cos(\gamma)\sin(\beta)$.

Denote by Q_0 the feasible region of (3.1), which is a cuboid. For any subset $Q \subseteq Q_0$, denote by $f_{lb}(Q)$ and $f_{ub}(Q)$ the lower and upper bounds of the objective function over Q, respectively. The following general branch-and-bound algorithm presented in [5] is employed to solve (3.1):

Algorithm 3.1.

s.0 Set $\epsilon > 0$. Initialize $k = 0$, $S_0 = \{Q_0\}$, $L_0 = f_{lb}(Q_0)$ and $U_0 = f_{ub}(Q_0)$.

s.1 If $U_k - L_k < \epsilon$, stop and return an ϵ-suboptimal solution R^* such that $f(R^*) = U_k$. Otherwise, goto s.2.

s.2 (Branching) Select $Q \in S_k$ such that $f_{lb}(Q) = L_k$ and then split Q along one of its longest edges into Q_l and Q_r. More precisely, suppose $Q = [\underline{q}_1, \overline{q}_1] \times [\underline{q}_2, \overline{q}_2] \times [\underline{q}_3, \overline{q}_3]$. Let $j = \arg\max_{i=1,2,3}(\overline{q}_i - \underline{q}_i)$. $q_j = (\underline{q}_j + \overline{q}_j)/2$. If $j = 1$, $Q_l = [\underline{q}_j, q_j] \times [\underline{q}_2, \overline{q}_2] \times [\underline{q}_3, \overline{q}_3]$, $Q_r = [q_j, \overline{q}_j] \times [\underline{q}_2, \overline{q}_2] \times [\underline{q}_3, \overline{q}_3]$. When $j = 2, 3$, Q_l and Q_r are similarly defined. Let $S_{k+1} = S_k \bigcup Q_l \bigcup Q_r \setminus Q$. Goto s.3.

s.3 (Bounding) Compute $f_{ub}(Q_l)$ and $f_{ub}(Q_r)$. Update the upper bound

$$U_{k+1} = \min\{U_k, f_{ub}(Q_l), f_{ub}(Q_r)\},$$

and the lower bound $L_{k+1} = \min_{Q\in S_{k+1}} f_{lb}(Q)$. Update the candidate optimal solution R^* as the feasible solution corresponding to U_{k+1}. Prune $\{Q : f_{lb}(Q) > U_{k+1}\}$ from S_{k+1}. Let $k := k+1$ and goto s.1.

Now, we discuss in detail the estimation of lower and upper bounds, $f_{lb}(Q)$ and $f_{ub}(Q)$, which are critical for the efficiency of the branch-and-bound algorithm.

Suppose the cuboid $Q = [\underline{q}_1, \overline{q}_1] \times [\underline{q}_2, \overline{q}_2] \times [\underline{q}_3, \overline{q}_3]$. It follows from the representation (3.2) that we can easily calculate the element-wise lower and upper bounds of R, denoted by L and U, respectively. For example, if $\max(\overline{q}_2, \overline{q}_3) \leqslant \pi/2$, $L(3,1) = \sin(\underline{q}_2)\sin(\underline{q}_3)$ and $U(3,1) = \sin(\overline{q}_2)\sin(\overline{q}_3)$. Define

$$R(Q) = \{R(\alpha,\beta,\gamma) : \ (\alpha,\beta,\gamma) \in Q\}.$$

Then, according to (3.2), we have

$$R(Q) = \{R \in S(3) : \ L \leqslant R \leqslant U\}. \tag{3.3}$$

Removing the constraint $R \in S(3)$ yields a lower relaxation of (3.1)

$$f_{lb}(Q) := \min_{L\leqslant R\leqslant U} \left\{f_1(R) = r(R)^T M r(R)\right\}, \tag{3.4}$$

which is a box-constrained convex quadratic programming (QP) problem and hence globally solved in polynomial time. In particular, at the root note (i.e., $Q = Q_0$), all the entries of L are -1 and $U = -L$. According to Proposition 2.4, we immediately have $f_{lb}(Q_0) = 0$ without any need to solve (3.4).

The upper bound $f_{ub}(Q)$ is set as $f_1(R^*)$, where $R^* \in R(Q)$ is obtained by some heuristic. Solving (3.4), we obtain a solution matrix, denoted by $\widetilde{R}$. If $\widetilde{R} \in S(3)$, then according to (3.3), we have $\widetilde{R} \in R(Q)$ and then set $R^* = \widetilde{R}$, $f_{ub}(Q) = f_1(\widetilde{R})$. Otherwise, let $\widetilde{R}^*$ be the closest point to $\widetilde{R}$ in $S(3)$, i.e.,

$$\widetilde{R}^* = \arg\min_{R\in S(3)} \left\{\|R-\widetilde{R}\|_F^2 = \mathrm{tr}\left((R-\widetilde{R})^T(R-\widetilde{R})\right) = \mathrm{tr}(I) - 2\mathrm{tr}(\widetilde{R}^T R) + \mathrm{tr}(\widetilde{R}^T\widetilde{R})\right\},$$

where $\|\cdot\|_F$ is the Frobenius norm, or equivalently,

$$\widetilde{R}^* = \arg\max_{R\in S(3)} \mathrm{tr}(\widetilde{R}^T R). \tag{3.5}$$

Let $\widetilde{R} = USV^T$ be the singular value decomposition (SVD) of $\widetilde{R}$, where U and V are orthogonal matrices and S is a diagonal matrix. As first given in [6], the solution of (3.5) is

$$\widetilde{R}^* = UV^T.$$

Suppose $\widetilde{R}^* \notin R(Q)$. Define $R^* = \arg\min_{R\in R(Q)} \|R - \widetilde{R}\|_F$. Then $\|R^* - \widetilde{R}\|_F > \|\widetilde{R}^* - \widetilde{R}\|_F$ and hence

$$\begin{aligned} |f_1(\widetilde{R}^*) - f_1(R^*)| &\leqslant L_p\|\widetilde{R}^* - R^*\|_F \\ &\leqslant L_p(\|\widetilde{R}^* - \widetilde{R}\|_F + \|\widetilde{R} - R^*\|_F) \\ &< 2L_p\|\widetilde{R} - R^*\|_F \\ &\leqslant 2L_p\|U - L\|_F, \end{aligned}$$

where L_p is the Lipschitz constant of $f_1(R)$. Therefore, to ensure the finite termination of the branch-and-bound algorithm, we can always set $f_{ub}(Q) = f_1(\widetilde{R}^*)$ whatever $\widetilde{R}^* \in R(Q)$.

Finally, we notice that the above branch-and-bound algorithm for solving (2.4) can be similarly employed to solve (2.11), (2.12), (2.13). Denote these four algorithms by B&B1, B&B2, B&B3 and B&B4, respectively.

4. Experiments

In order to produce the data, we use a virtual calibrated camera having a focal length 800 and a principal point at $(320, 240)$. The random 3D reference points are within the rectangular box defined by $[-2, 2] \times [-2, 2] \times [4, 8]$ in the object space, and the corresponding 2D projections lie inside the 640×480 image plane, with additional Gaussian noise.

4.1. Numerical comparison among different reformulations

In this subsection, we compare the performance of B&B1, B&B2, B&B3 and B&B4. We independently run 200 simulations using MATLAB R2009b on a computer with a 2.66 GHz Intel Core 2 Duo processor and 4GB RAM. The number of the reference points varies from 6 to 100. The standard deviation of Gaussian noise is fixed at $\sigma = 5$. The semidefinite programming problems are solved using the SeDuMi software package [25]. The convex quadratic programming problems are solved using the compatible function 'quadprog' in the Matlab optimization toolbox.

We report in Table 1 the numerical results where the parameter $\epsilon = 0.01$ is fixed. The first column is to distinguish the four different algorithms. The second column gives the number of reference points. We report in Column 3-6 the average running time in seconds (where the modeling time is omitted since we just focus on the comparison among the four branch-and-bound algorithms), the average (sub)optimal function values, the average number of iterations and the total number of times that the algorithm terminates in the first iteration within 200 simulations, respectively. From Table 1, we see the B&B4 algorithm highly outperforms the other three algorithms.

We report in Table 2 the performance of B&B4 with different stop criterions (i.e., ϵ varies from 10^{-6} to 10^{-3}) and different Gaussian noises (i.e., σ varies from 1 to 15). For each σ and every fixed number of reference points (n), we independently run 500 simulations. Surprisingly, it is observed that, with a high probability, B&B4 terminates in the first iteration. Moreover, according to Table 2, this probability seems to be an increasing function of n and a decreasing function in terms of both σ and ϵ.

4.2. Numerical comparison among B &B4, OI and SDR

Let R_{true} and t_{true} be the true camera rotation and the true translation, respectively. Then we define the relative errors of the estimated rotation R and the estimated translation t:

$$E_{rot} = \frac{\|R - R_{true}\|_F}{\|R\|_F}, \quad E_{tran} = \frac{\|t - t_{true}\|}{\|t\|}.$$

Table 1: Comparison among B&B1, B&B2, B&B3 and B&B4.

B&B	n	Time(s)	Avg. obj.	Avg. it.	#(it.= 1)
1	6	14.0318	0.0122	201.72	5
	8	15.4215	0.0183	238.36	2
	10	13.9087	0.0221	223.94	3
	20	12.8818	0.0512	213.845	0
	50	16.0813	0.1431	277.735	0
	100	28.0318	0.2918	388.245	0
2	6	1.5788	0.0127	23.305	53
	8	0.4625	0.0181	7.81	121
	10	0.1472	0.0213	3.09	173
	20	0.0108	0.0502	1	200
	50	0.0090	0.1425	1	200
	100	0.0086	0.2941	1	200
3	6	2.6326	0.0129	39.6	25
	8	1.6127	0.0190	26.85	90
	10	0.5957	0.0219	10.44	134
	20	0.0133	0.0502	1.03	197
	50	0.0095	0.1425	1	200
	100	0.0091	0.2914	1	200
4	6	0.0340	0.0100	1.275	189
	8	0.0127	0.0165	1	200
	10	0.0112	0.2070	1	200
	20	0.0094	0.0502	1	200
	50	0.0088	0.1425	1	200
	100	0.0086	0.2914	1	200

Based on E_{rot} and E_{tran}, we compare in this subsection the performance of the B&B4 algorithm with those of two well-known heuristics, OI [15] and SDR [23]. These three methods all target at minimizing the object space error (2.1). Though the branch-and-bound algorithm gives an ϵ-suboptimal solution of (2.1), it is interesting to notice that the object space error function is neither E_{rot} nor E_{tran}.

We independently run 500 simulations, where the number of the reference points varies from 6 to 50, the standard deviation of Gaussian noise varies from 0 to 10, and the parameter ϵ in the B&B4 algorithm is set as 0.001.

We plot in Figure 1 the average relative errors as functions of the noise, where the number of the reference points is fixed at 6. It is observed that the average relative rotation error E_{rot} corresponding to either B&B4 or SDR is smaller than that of OI. When the pixel errors are small (for example, the standard deviation of the noise is between 0 and 5), the accuracy of the solution obtained by B&B4 is slightly higher than that of SDR. But it is no longer true for large amounts of noise. This makes sense since the gap between the object space error and the rotation error E_{rot} becomes large. For the average relative translation error E_{tran}, we see that the performance of SDR and OI is less accurate than that of B&B4.

Figure 2 plots the average relative errors as functions of the noise where the number of the reference points is fixed at 20. Again, it is observed that the accuracy of the solution obtained by SDR is lower than that of either OI or B&B4 for small amounts of noise.

Figure 3 depicts the average relative errors as functions of n, the number of the reference points, at the same level of noise ($\sigma = 5$). We observe that for $n \geqslant 10$, B&B4 and OI have almost the same performance. And, the solution obtained by either B&B4 or OI is more accurate than that of SDR. On the other hand,

Table 2: The percentage of the number of times that B&B4 with different ϵ and σ terminates in the first iteration.

σ	n	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$
1	6	100	100	100	100
	7	100	100	100	100
	8	100	100	100	100
	9	100	100	100	100
	10	100	100	100	100
	20	100	100	100	100
	50	100	100	100	100
	100	100	100	100	100
5	6	99.2	98.8	98.8	98.8
	7	100	99.8	99.8	99.8
	8	100	100	100	100
	9	100	100	100	100
	10	100	100	100	100
	20	100	100	100	100
	50	100	100	100	100
	100	100	100	100	100
10	6	98.6	98	98	98
	7	99.4	99.4	99.4	99.4
	8	99.8	99.8	99.6	99.6
	9	100	100	100	100
	10	100	100	100	100
	20	100	100	100	100
	50	100	100	100	100
	100	100	100	100	100
15	6	95.4	93.8	93.8	93.8
	7	97.4	96.2	95.8	95.8
	8	98.8	98.8	98.8	98.6
	9	99.6	99.4	99.4	99.4
	10	100	99.8	99.8	99.8
	20	100	100	100	100
	50	100	100	100	100
	100	100	100	100	100

when $n < 10$, B&B4 slightly outperforms the best.

Figure 4 plots the average running time (in seconds) for the three algorithms, where the number of the reference points varies from 6 to 1000. Here we notice that the running time of B&B4 contains the modeling time (i.e., the time to solve SDP (2.8) so as to construct (2.13)), which is omitted in the third column of Table 1. When $n = 6$, OI performs the fastest. But the average running time of OI grows linearly when increasing the number of the reference points. For larger number of points, the average running time of SDR and B&B4 are almost the same constant (≈ 50 ms). Moreover, for $n \geqslant 400$, B&B4 runs a bit faster than SDR. This makes sense since the major computational cost of SDR and the first iteration of B&B4 are solving the corresponding semidefinite programming problems, where the sizes of the coefficient matrices of the constraints are 117×32 and 81×12, respectively.

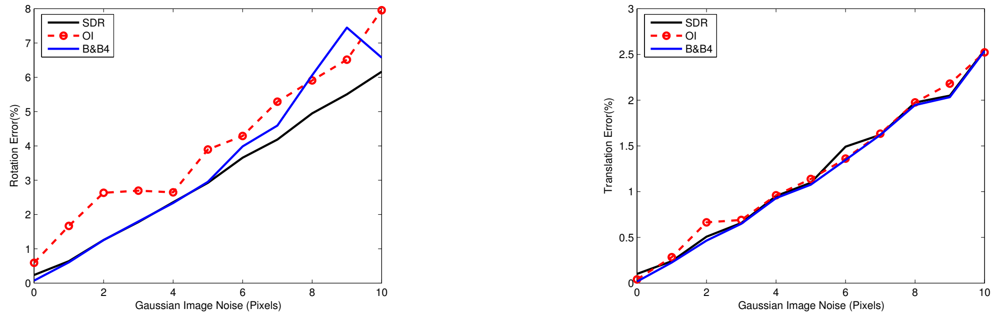

Figure 1: Average rotation and translation errors for different levels of Gaussian noise using 6 reference points.

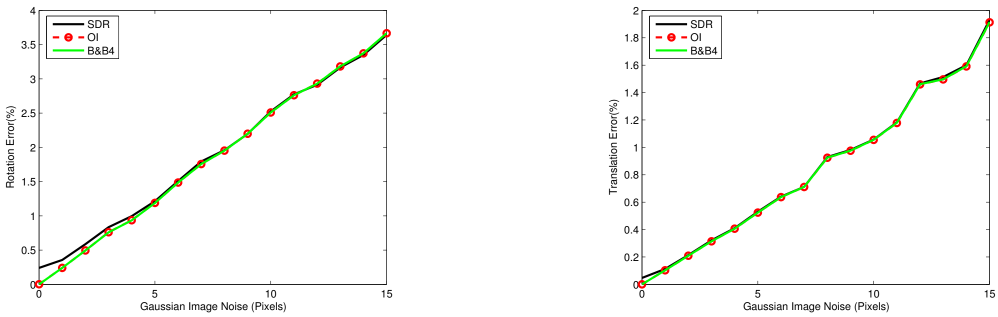

Figure 2: Average rotation and translation errors for different levels of Gaussian noise using 20 reference points.

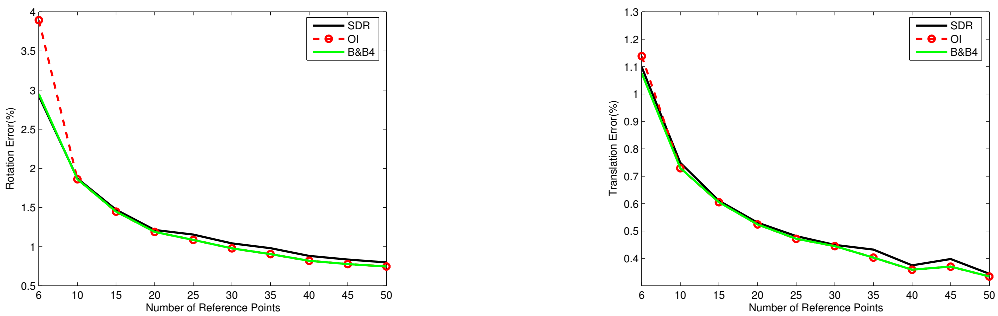

Figure 3: Average rotation and translation errors for different number of reference points using the noise fixed at $\sigma = 5$.

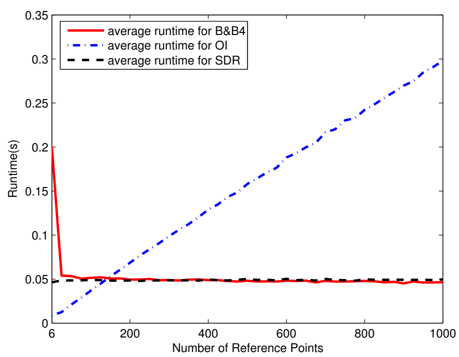

Figure 4: Average runtime for different number of reference points using the noise fixed at $\sigma = 5$.

4.3. Numerical comparison among B& B4 and BARON

We report in Table 3 the numerical results where the parameter $\epsilon = 10^{-5}$ and the standard deviation of Gaussian noise $\sigma = 5$ are fixed, the stopping tolerance parameter is set as 10^{-5} in BARON. The first column is to distinguish the two different algorithms. The second column gives the number of reference points. We report in Column 3-4 the average running time in seconds where the modeling time is contained and the average number of iterations, respectively. From Table 3, we can see that the B&B4 algorithm highly outperforms the commercial global optimization solver BARON.

Table 3: Comparison between B&B4 and BARON based on 200 simulations.

Algorithm	n	Time(s)	Avg. it.
B&B4	6	0.31	4.21
	8	0.09	1.07
	10	0.08	1.03
	20	0.06	1
	50	0.08	1
	100	0.08	1
BARON	6	5.58	640.83
	8	9.47	4160.09
	10	5.03	982.48
	20	1.43	53
	50	3.87	1168.43
	100	7.99	4687.02

5. Conclusions

In the area of pose estimation, the model of minimizing the object space error has been used in many heuristics including the well-known orthogonal iteration (OI) and the very recent semidefinite programming relaxation (SDR). In this paper, we first present three Lagrangian dual problems for this model. All can be formulated as semidefinite programs and hence can be efficiently solved. Using the optimal dual variables as Lagrangian multipliers, we propose three new quadratic constrained quadratic programming (QCQP) reformulations for the pose estimation model. Then, based on convex quadratic programming relaxation, we develop new branch-and-bound algorithms for these QCQP models. We show the branch-and-bound algorithm corresponding to the QCQP model having the tightest relaxation is the most efficient. According to the numerical simulation results, it usually terminates in the first iteration and returns an ϵ-suboptimal solution. Finally, a comprehensive numerical comparison demonstrates that the branch-and-bound algorithm outperforms OI, SDR and the global optimization package BARON. One of the future works is to theoretically study the proposed new models.

Acknowledgment

This research was supported by National Natural Science Foundation of China under grants No. 71271015, 71571008 and No. 11471325, 11571029.

References

[1] S. Agarwal, M. K. Chandraker, F. Kahl, D. Kriegman, S. Belongie, *Practical global optimization for multiview geometry*, Comput. Vis. ECCV, Graz, Austria, May 7–13, Springer, Berlin, Heidelberg, (2006), 592–605. 1

[2] A. Ansar, K. Daniilidis, *Linear pose estimation from points or lines*, Comput. Vis. ECCV, Denmark, May 28–31, Springer, Berlin, Heidelberg, (2002), 282–296. 1

[3] K. Anstreicher, X. Chen, H. Wolkowicz, Y.-X. Yuan, *Strong duality for a trust-region type relaxation of the quadratic assignment problem*, Linear Algebra Appl., **301** (1999), 121–136. 2.2

[4] V. Balakrishnan, S. Boyd, S. Balemi, *Branch and bound algorithm for computing the minimum stability degree of parameterdependent linear systems*, Internat. J. Robust Nonlinear Control, **1** (1991), 295–317. 3

[5] H. P. Benson, *Using concave envelopes to globally solve the nonlinear sum of ratios problem*, Dedicated to Professor Reiner Horst on his 60th birthday, J. Global Optim., **22** (1936), 343–364. 3

[6] C. Eckart, G. Yong, *The approximation of one matrix by another of lower rank*, Psychometrika, **1** (1936), 211–218. 3

[7] P. D. Fiore, *Efficient linear solution of exterior orientation*, IEEE Trans. Pattern Anal. Mach. Intell., **23** (2001), 140–148. 1

[9] R. Hartley, F. Kahl, *Global Optimization through Searching Rotation Space and Optimal Estimation of the Essential Matrix*, International Conference on Computer Vision, (2007), 1–8. 1

[10] R. Hartley, F. Kahl, *Global optimization through rotation space search*, Int. J. Comput. Vis., **82** (2009), 64–79. 1

[12] R. Hartley, A. Zisserman, *Multiple view geometry in computer vision*, Second edition, With a foreword by Olivier Faugeras, Cambridge University Press, Cambridge, (2003). 1

[11] H. Hmam, J. Kim, *Optimal non-iterative pose estimation via convex relaxation*, Image Vis. Comput., **28** (2010), 1515–1523. 1

[14] S.-Q. Li, C. Xu, M. Xie, *A robust* $O(n)$ *solution to the perspective-n-point problem*, IEEE Trans. Pattern Anal. Mach. Intell., **34** (2012), 1444–1450. 1

[15] C.-P. Lu, G. D. Hager, E. Mjolsness, *Fast and globally convergent pose estimation from video images*, IEEE Trans. Pattern Anal. Mach. Intell., **22** (2000), 610–622. 1, 2.1, 4.2

[17] F. Moreno-Noguer, V. Lepetit, P. Fua, *Accurate non-iterative* $O(n)$ *solution to the PnP problem*, IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, (2007), 1–8. 1

[18] C. Olsson, F. Kahl, M. Oskarsson, *Optimal estimation of perspective camera pose*, 18th International Conference on Pattern Recognition, Hong Kong, China, **2** (2006), 5–8. 1

[19] L. Quan, Z.-D. Lan, *Linear n-point camera pose determination*, IEEE Trans. Pattern Anal. Mach. Intell, **21** (1999), 774–780. 1

[20] M. Sarkis, K. Diepold, *Camera-pose estimation via projective Newton optimization on the manifold*, IEEE Trans. Image Process., **21** (2012), 1729–1741. 1

[22] G. Schweighofer, A. Pinz, *Fast and Globally Convergent Structure and Motion Estimation for General Camera Models*, Proceedings of the 17th British Machine Vision Conference, (2006), 147–156. 2.1

[21] G. Schweighofer, A. Pinz, *Robust pose estimation from a planar target*, IEEE Trans. Pattern Anal. Mach. Intell., **28** (2006), 2024–2030. 1

[23] G. Schweighofer, A. Pinz, *Globally optimal* $O(n)$ *solution to the PnP problem for general camera models*, Proceedings of the 19th British Machine Vision Conference, (2008), 1–10. 1, 2.1, 2.1, 4.2

[25] J. F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., **11** (1990), 625–653. 2.2, 4.1

[24] O. Tahri, H. Araujo, Y. Mezouar, F. Chaumette, *Efficient iterative pose estimation using an invariant to rotations*, IEEE Trans. Cybern., **44** (2014), 199–207. 1

[26] M. Tawarmalani, N. V. Sahinidis, *Semidefinite relaxations of fractional programs via novel convexification techniques*, J. Global Optim., **20** (2001), 133–154. 1

[27] Y. Xia, *Global optimization of a class of nonconvex quadratically constrained quadratic programming problems*, Acta Math. Sin. (Engl. Ser.), **27** (2011), 1803–1812. 2.2

[28] X. S. Zhou, S. I. Roumeliotis, *Determining 3-D relative transformations for any combination of range and bearing measurements*, IEEE Trans. Robot., **29** (2013), 458–474. 1