Numerical Optimization on the Euclidean Group With Applications to Camera Calibration

Seungwoong Gwak, Junggon Kim, and Frank Chongwoo Park, *Member, IEEE*

***Abstract*—We present the cyclic coordinate descent (CCD) algorithm for optimizing quadratic objective functions on $SE(3)$, and apply it to a class of robot sensor calibration problems. Exploiting the fact that $SE(3)$ is the semidirect product of $SO(3)$ and $\Re^3$, we show that by cyclically optimizing between these two spaces, global convergence can be assured under a mild set of assumptions. The CCD algorithm is also invariant with respect to choice of fixed reference frame (i.e., left invariant, as required by the principle of objectivity). Examples from camera calibration confirm the simplicity, efficiency, and robustness of the CCD algorithm on $SE(3)$, and its wide applicability to problems of practical interest in robotics.**

***Index Terms*—Camera calibration, cyclic coordinate descent (CCD), Euclidean group, optimization, rotation group, $SE(3)$, $SO(3)$.**

I. Introduction

As IS well known, the Euclidean group of rigid-body motions, denoted $SE(3)$ and referred to more simply as the Euclidean group, plays a central role in the mathematical description of the location of rigid objects. Aside from its more obvious uses in the description of a robot's kinematics, the Euclidean group is the natural setting in which to express and solve a wide variety of problems, ranging from workpiece localization and camera calibration to trajectory generation and mechanism design.

In the context of the above applications, one frequently encounters situations that require the solution of an optimization problem on the Euclidean group. For example, Li *et al.* [8] have formulated the workpiece localization problem as one of minimizing a function of the form $\eta(g, x) = \sum_{i=1}^{n} \|g^{-1}y_i - x_i\|^2$, where $g \in SE(3)$ and $x \in \Re^{3n}$, and each $y_i \in \Re^3$. The problem of robot sensor calibration, in which the objective is to determine the relative location of a sensor mounted on a robot end-effector, can naturally be framed as one of minimizing $\eta(X) = \|AX - XB\|^2$, where A, B, $X \in SE(3)$, with A, B constructed from measurements [14], [17]. The camera calibration problem, which we address in more detail later in this paper, can also be represented as the optimization of an objective function of the form $\eta(X, y) = \sum_{i=1}^{m} \mathrm{Tr}\{XN_iX^TQ_i(y)\}$, where $X \in SE(3)$ and $y \in \Re^k$, with N_i and Q_i symmetric matrices constructed from measurements.

In principle, one can choose a set of local coordinates for $SE(3)$, and apply any number of unconstrained vector space optimization algorithms. One could, for example, parameterize the rotation group $SO(3)$ using, e.g., Euler angles or exponential coordinates, and optimize the objective function with respect to the chosen parameters. Alternatively, there is both a well-developed theory and an abundance of numerical algorithms for the constrained optimization of functions defined on vector spaces. In the case of $SO(3)$, for example, one could view the optimization as taking place in the vector space of 3×3 real matrices, and impose the equality constraints $R^TR = I$ and $\det(R) = 1$ for any $R \in \Re^{3\times3}$.

One disadvantage with the local coordinate-based approach is that local coordinates typically cannot cover the entire manifold with a single coordinate chart. Changes in local coordinates are required when the iteration ventures near boundaries of the local chart. Also, particular choices of local coordinates can introduce large nonlinearities that degrade algorithm performance. A general feature of constrained vector space optimization algorithms is that imposing algebraic constraints on the optimization parameters, without taking into account the geometric structure of the problem, can often complicate the optimization procedure. For example, the iterations may produce points that only approximately satisfy the constraints, requiring corrective procedures after each iteration.

In this paper, we suggest an alternative approach that takes advantage of the geometric structure of the Euclidean group $SE(3)$, and, in the absence of other constraints on the optimization parameters, recasts the problem as one of unconstrained minimization. More generally, in the case of Lie groups like $SE(3)$, it is possible to exploit the geometric and algebraic structure of the underlying space, and obtain natural extensions of various unconstrained vector space optimization algorithms. In certain cases, and we show this to be true for the class of problems listed above, the geometric approach leads to algorithms with faster and more robust convergence properties.

Previous literature on the optimization of functions defined on curved spaces has focused primarily on compact semisimple Lie groups like $SO(n)$. Smith [18] outlines a general mathematical framework for generalizing standard optimization algorithms like steepest descent, conjugate gradient, and Newton's method to general Riemannian manifolds and Lie groups. The above-referenced paper makes clear that such geometric algorithms are most effective when geodesics in the underlying space can be computed relatively easily, which happens to be the case for compact semisimple Lie groups. Park

Manuscript received February 27, 2001; revised December 29, 2001. This paper was recommended for publication by Associate Editor D. Pai and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported in part by the National Research Laboratory for CAD/CAM, and by the BK21 Program in Mechanical Engineering, both at Seoul National University, Seoul, Korea.

S. Gwak and F. C. Park are with the School of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, Korea.

J. Kim is with the Intelligent Mechatronics Group, Hyundai Heavy Industries Research Laboratories, Mabookri 449–910, Korea.

Digital Object Identifier 10.1109/TRA.2002.807530

et al. [16] have independently developed a set of geometric optimization algorithms on $SO(3)$ for solving the attitude determination problem using the global positioning system. Ma [11] also develops a set of geometric optimization algorithms on $SO(3) \times S^2$ for solving a variety of problems arising in computer vision.

Motivation for the above works can be traced in large part to the seminal work of Brockett [4], [5], who studied gradient flows for objective functions on $SO(n)$ of the form $\mathrm{Tr}\{RNR^TQ\}$, where N and Q are given symmetric $n \times n$ matrices. He shows that a variety of continuous and combinatorial problems (e.g., the eigenvalue problem, linear programming, sorting) can be posed as the minimization of such an objective function. Li [8], Bloch *et al.* [3], Helmke and Moore [6], and others (see, e.g., [4] and the references cited therein) further examine the geometry of these and other related optimization problems on non-Euclidean spaces. In the case when N and Q are positive definite, Bayard [1] derives a closed-form analytic solution for the global minimizer and maximizer.

In this paper, we focus specifically on quadratic objective functions on the Euclidean group $SE(3)$. Unlike $SO(n)$, because $SE(n)$ is not compact, and because natural (or bi-invariant) Riemannian metrics do not exist on $SE(n)$, a number of important subtleties arise when attempting to develop a corresponding set of geometric optimization algorithms. For our purposes, perhaps the most important difference is that the one-parameter subgroups on $SE(n)$ are no longer geodesics. The result is that the computation of gradients and Hessians on $SE(n)$ becomes more complicated; the straightforward extension of the geometric optimization algorithms defined on $SO(n)$ to the Euclidean group becomes much less appealing. Worse yet, if one abandons the computation of geodesics on $SE(n)$ in favor of the more easily computed one-parameter subgroups, then, in most cases, the algorithms fail to converge.

Instead, in this paper we develop a cyclic coordinate descent (CCD) algorithm for optimizing quadratic functions on $SE(3)$. Exploiting the fact that $SE(3)$ is the semidirect product of $SO(3)$ and $\Re^3$, we show that by cyclically optimizing between these two spaces, global convergence (in the sense of converging to a point that satisfies the first-order necessary conditions) can be guaranteed under a mild set of assumptions. In this way, one can also take advantage of the many efficient geometric optimization algorithms that have already been developed for $SO(3)$.

The paper is organized as follows. In Section II we review the necessary geometric background on the Euclidean group. The CCD algorithm is presented in Section III, together with a proof of global convergence and some examples. In Section IV, we present a detailed case study of the intrinsic camera calibration problem, and show that the $SE(3)$ geometric optimization algorithm proposed in the paper offers a unified, simple, and efficient means of solving the camera calibration problem.

II. Geometry of the Euclidean Group

Recall that the *special orthogonal group* $SO(n)$ is defined by

$$SO(n) = \{R \,|\, R \in GL(n, \Re),\ RR^T = I,\ \det R = +1\} \quad (1)$$

where *the general linear group* $GL(n, \Re)$ is the group consisting of $n \times n$ nonsingular real matrices. The Lie algebra of $SO(n)$, denoted $so(n)$, may be identified with the real $n \times n$ skew-symmetric matrices; for $n = 3$ they assume the form

$$[w] = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \quad (2)$$

where $w = (w_1\ w_2\ w_3)^T \in \Re^3$. In this paper, we use the square bracket $[\omega]$ to denote the skew-symmetric matrix representation of a vector $\omega \in \Re^3$, and a superscript arrow $\vec{\Omega}$ to denote the vector representation of a skew-symmetric matrix Ω.

The *special Euclidean group* $SE(n)$ is given by

$$SE(n) = \left\{ \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \,\middle|\, R \in SO(n),\ p \in \Re^n \right\}. \quad (3)$$

The Lie algebra of $SE(3)$, denoted $se(3)$, can be identified with the 4×4 matrices of the form

$$\xi = \begin{bmatrix} [w] & v \\ 0 & 0 \end{bmatrix} \qquad w,\ v \in \Re^3. \quad (4)$$

For both $so(3)$ and $se(3)$ the Lie bracket is given by the matrix commutator $[A,\ B] = AB - BA$.

We now consider the construction of minimal geodesics on $SO(n)$ and $SE(n)$. In the standard construction of a distance metric on a Riemannian manifold, the distance between two points on a Riemannian manifold is defined as the length of the shortest curve connecting the two points; here, curve length is measured in the usual way with respect to the Riemannian metric. On $SO(3)$ with the natural metric, the shortest curves, or minimal geodesics, are of the form

$$R(t) = R_0 e^{\Omega t} \quad (5)$$

where $R_0 \in SO(3)$ is the initial point, $R_1 \in SO(3)$ is the final point, and $\Omega = \log(R_0^T R_1) \in so(3)$. Here, the curve $R(t)$ is defined such that $R(0) = R_0$ and $R(1) = R_1$.

In [9], it is shown that left invariance is sufficient to guarantee that the principle of objectivity is satisfied. Since $SE(3)$ does not admit a bi-invariant Riemannian metric, it is reasonable to look for minimal geodesics with respect to left-invariant metrics; [2] contains a discussion of Riemannian metrics in $SE(3)$ and their implications for motion interpolation. In [15], it is shown that the minimal geodesics on $SE(3)$ with respect to the standard left-invariant Riemannian metric defined by $ds^2 = d\omega^2 + dv^2$, have the form

$$X(t) = (R_0 e^{\Omega t},\ p_0 + (p_1 - p_0)t) \quad (6)$$

for the end points $X(t_0) = (R_0,\ p_0)$ and $X(t_1) = (R_1,\ p_1)$. That is, the minimal geodesic on $SE(3)$ consists of the union of the respective geodesics on $SO(3)$ and $\Re^3$.

III. CCD on $SE(n)$

We begin with a brief discussion of the standard CCD algorithm in $\Re^n$. Given an objective function $L(x)$, $x \in \Re^n$, the CCD method optimizes $L(x)$ by sequentially minimizing

with respect to each of the components x_i; that is, $L(x)$ is first minimized with respect to x_1 while keeping the remaining x_i's fixed, followed by x_2, x_3, and so forth (variations of this method are collectively referred to as coordinate descent methods). Although, in general, the CCD method has poorer convergence properties than steepest descent, it has the advantage that complete gradient information for $L(x)$ is not required in determining the descent direction, and that it is unaffected by (diagonal) scale factor changes in the coordinates. On the other hand, the CCD algorithm (and coordinate descent methods in general) is affected by orthogonal transformations of coordinates. The exact opposite is true for methods like steepest descent or Newton's method, which are invariant to orthogonal coordinate transformations but affected by diagonal scale factor changes.

One of the attractive features of the minimal geodesic-based approach to optimization on $SO(n)$—indeed, what makes this class of algorithms of practical use—is the relatively straightforward means by which the minimal geodesics can be computed, as the one-parameter subgroups, or matrix exponentials. Unfortunately, the one-parameter subgroups on $SE(n)$ are not minimal geodesics. The necessary computations in determining the minimal geodesic in the direction of the gradient (or, in the case of Newton's method, the gradient transformed by the inverse Hessian) become much more involved, to the point where geometric descent algorithms begin to lose their appeal. Although it is tempting to simply replace the minimal geodesics on $SE(n)$ by the one-parameter subgroups, the algorithms in this case usually fail to converge.

Rather than attempting to perform line searches over the minimal geodesics on $SE(n)$, the strategy we adopt is to perform CCD over $SO(n)$ and $\Re^n$. In this section, we first provide a brief review of the steepest descent and Newton's methods on $SO(3)$, followed by a convergence analysis of CCD on $SE(n)$. We then present examples of the CCD algorithm applied to various quadratic objective functions on $SE(3)$.

A. Numerical Optimization on $SO(3)$

In this subsection, we review both steepest descent and Newton's methods as first presented in [16] for quadratic objective functions on $SO(3)$ of the form $J(R) = \operatorname{tr}(RNR^TQ - 2RW)$. Although most of the objective functions that we shall encounter are of the more general form $J(R) = \sum_{i=1}^{m} \operatorname{tr}(RN_iR^TQ_i - 2RW_i)$, the extension of the algorithms below to this case should be straightforward. For the simpler form of objective function $J(R) = \operatorname{tr}(RNR^TQ)$, where $R \in SO(n)$, and $N = Y^TY$ and $Q = XX^T$ are both positive definite $n \times n$ matrices, Bayard [1] offers closed-form analytic formulas for the global maximizer and minimizer. Letting $N = P_n\Lambda_nP_n^T$ and $Q = P_q\Lambda_qP_q^T$ be orthogonal decompositions with the diagonal entries of P_n and P_q arranged in decreasing order, he shows that the global maximizer and global minimizer are given by $R_{\max} = P_qP_n^T$ and $P_qJP_n^T$, where $J \in \Re^{n\times n}$ is the reverse identity with ones along the opposite diagonal.

The procedure for steepest descent can be summarized as follows.

1) Expand the objective function $J(R)$ defined on $SO(3)$ about $R_k \in SO(3)$ in terms of the matrix exponential $R = R_ke^{\Omega}$, $\Omega \in so(3)$, as

$$J(R) = J(R_k) + \operatorname{tr}\{2(NR_k^TQR_k - WR_k)\Omega\} + \cdots.$$

2) Calculate the gradient

$$\Omega_k = NR_k^TQR_k - (NR_k^TQR_k)^T - WR_k + (WR_k)^T.$$

3) Perform a line search along the minimal geodesic indicated by the gradient direction

$$t_k = \arg\min_{t\in\Re} J(R_ke^{\Omega_k t})$$
$$R_{k+1} = R_ke^{\Omega_k t_k}.$$

4) Go to step 2) until the convergence conditions are satisfied.

In general, the steepest descent method converges more slowly than Newton or quasi-Newton methods, particularly if the Hessian of the objective function is ill conditioned (i.e., the level sets of the function resemble a long narrow valley). In [16], methods for rapidly computing a rough stepsize that guarantee a decrease in the objective function are given; empirical simulations indicate a significant improvement in convergence over an exact line search.

Since Newton's method uses both the gradient and Hessian in determining the line search direction, the objective function in this case needs to be expanded to second order. The algorithm for Newton's method on $SO(3)$ to second order is as follows.

1) Expand $J(R)$ to second order near $R_k \in SO(3)$ using the matrix exponential $R = R_ke^{\Omega t}$, $\Omega \in so(3)$

$$\begin{aligned} J(R) = J(R_k) &+ \operatorname{tr}\{2(NR_k^TQR_k - WR_k)\Omega\} \\ &+ \tfrac{1}{2}\operatorname{tr}\{2(NR_k^TQR_k - WR_k)\Omega^2 \\ &\quad - 2R_k^TQR_k\Omega N\Omega\} + \cdots. \end{aligned}$$

2) Calculate the following temporary variables:

$$\begin{aligned} T &= NR_k^TQR_k + (NR_k^TQR_k)^T - WR_k - (WR_k)^T \\ M &= -2R_k^TQR_k \\ S &= 2\{(NR_k^TQR_k)^T - NR_k^TQR_k + (WR_k)^T - WR_k\}. \end{aligned}$$

3) Calculate the Hessian

$$\begin{aligned} H = MN + NM &- \operatorname{tr}(M)N - \operatorname{tr}(N)M \\ &+ (\operatorname{tr}(M)\operatorname{tr}(N) - \operatorname{tr}(MN))I + T - \operatorname{tr}(T)I. \end{aligned}$$

4) Find R_{k+1} that minimizes the second-order quadratic approximation to $J(R)$

$$\Omega_k = \left[H^{-1}\vec{S}\right]$$
$$t_k = \arg\min_{t\in\Re} J(R_ke^{\Omega_k t})$$
$$R_{k+1} = R_ke^{\Omega_k t_k}.$$

Recall $\vec{S}$ is the three-dimensional (3-D) vector representation of the skew-symmetric matrix S, while $[\cdot]$ is the 3×3 skew-symmetric matrix representation of a 3-D vector.

5) Go to step 2) until the convergence conditions are satisfied.

B. CCD on $SE(n)$

In this subsection, we perform a global convergence analysis of the CCD algorithm on $SE(n)$; our approach is based on Luenberger's [10] presentation of the framework originally laid out by Zangwill [20] for the global convergence analysis of algorithms. We first review the general case of X being a metric space.

Definition 1: An *algorithm* $\mathbf{A}$ is a mapping defined on X that assigns to every point $\mathbf{x} \in \mathbf{X}$ a subset of X.

An algorithm $\mathbf{A}$ generates a sequence of points as follows. Given $\mathbf{x_k} \in \mathbf{X}$, the algorithm yields $\mathbf{A}(\mathbf{x_k})$, which is a subset of X; in our case $\mathbf{A}(\mathbf{x_k})$ will typically be a point, $\mathbf{x_{k+1}}$.

Definition 2: Given an algorithm $\mathbf{A}$ on X, a continuous real-valued function Z: $X \to \Re$, and a subset $\Gamma \subset X$, Z is said to be a *descent function* for Γ and $\mathbf{A}$ if it satisfies the following.

1) If $\mathbf{x} \notin \mathbf{\Gamma}$ and $\mathbf{y} \in \mathbf{A}(\mathbf{x})$, then $Z(\mathbf{y}) < \mathbf{Z}(\mathbf{x})$.
2) If $\mathbf{x} \in \mathbf{\Gamma}$ and $\mathbf{y} \in \mathbf{A}(\mathbf{x})$, then $Z(\mathbf{y}) \leq \mathbf{Z}(\mathbf{x})$.

In this case, Γ is said to be a *solution set* for $\mathbf{A}$ and Z.

For example, in the standard problem of minimizing $L(\mathbf{x})$ subject to $\mathbf{x} \in \mathbf{X}$, we let Γ be the set of minimizing points, and define an algorithm $\mathbf{A}$ on X in such a way that L decreases at each step and thereby serves as a descent function.

Definition 3: A point-to-set mapping $\mathbf{A}$ from X to Y is said to be *closed* at $\mathbf{x} \in \mathbf{X}$ if the assumptions

1) $\mathbf{x_k} \to \mathbf{x}$, $\mathbf{x_k} \in \mathbf{X}$;
2) $\mathbf{y_k} \to \mathbf{y}$, $\mathbf{y_k} \in \mathbf{A}(\mathbf{x_k})$

imply $\mathbf{y} \in \mathbf{A}(\mathbf{x})$. The point-to-set map $\mathbf{A}$ is said to be *closed* on X if it is closed at each point of X.

It can be shown that given two point-to-set mappings $\mathbf{A}$: $\mathbf{X} \to \mathbf{Y}$ and $\mathbf{B}$: $\mathbf{Y} \to \mathbf{Z}$, if $\mathbf{A}$ is closed at $\mathbf{x}$, $\mathbf{B}$ is closed on $\mathbf{A}(\mathbf{x})$, and Y is compact, then the composite map $\mathbf{C} = \mathbf{B} \circ \mathbf{A}$: $X \to Z$ is closed at $\mathbf{x}$. With the above preliminaries, the Global Convergence Theorem can now be stated.

Theorem 1 (Global Convergence Theorem): Let $\mathbf{A}$ be an algorithm on X, and suppose that, given $\mathbf{x_0}$, the sequence $\{\mathbf{x}\}_{\mathbf{k=0}}^{\infty}$ is generated, satisfying $\mathbf{x_{k+1}} \in \mathbf{A}(\mathbf{x_k})$. Let a solution set $\Gamma \in X$ be given, and suppose

1) all points $\mathbf{x_k}$ are contained in a compact set $S \in X$;
2) there is a continuous function Z on X such that
2a) if $\mathbf{x} \notin \mathbf{\Gamma}$, then $Z(\mathbf{y}) < \mathbf{Z}(\mathbf{x})$ for all $\mathbf{y} \in \mathbf{A}(\mathbf{x})$;
2b) if $\mathbf{x} \in \mathbf{\Gamma}$, then $Z(\mathbf{y}) \leq \mathbf{Z}(\mathbf{x})$ for all $\mathbf{y} \in \mathbf{A}(\mathbf{x})$;
3) the mapping $\mathbf{A}$ is closed at points outside Γ.

Then the limit of any convergent subsequence of $\{\mathbf{x}\}_\mathbf{k}$ is a solution.

Proof: See [10]. ∎

An important corollary of the Global Convergence Theorem is that if, under the conditions of the theorem, Γ consists of a single point $\overline{\mathbf{x}}$, then the sequence $\{\mathbf{x_k}\}$ converges to $\overline{\mathbf{x}}$.

With the above preliminaries, we now consider the objective function L: $SE(n) \to \Re$, denoted $L(R, p)$. We first define the following mappings.

Definition 4: 1) $\mathbf{C^1}$: $\mathbf{SE(n)} \to \mathbf{SE(n)} \times \mathbf{\Re^n}$ is a point-to-point mapping defined by

$$\mathbf{C^1}(\mathbf{R}, \mathbf{p}) = (\mathbf{R}, \mathbf{p}, \mathbf{d}) \tag{7}$$

where d corresponds to the search direction along the $\Re^n$ component of $SE(n)$ at the point (R, p). For example, for steepest descent, $d = -\nabla_p L(R, p)$ where the subscript p denotes the gradient of the objective function L with respect to p.

2) $\mathbf{S^1}$ is a mapping defined on $SE(n) \times \Re^n$ that assigns to every point $(R, p, d) \in SE(n) \times \Re^n$ the following subset in $SE(n)$:

$$\mathbf{S^1}(\mathbf{R}, \mathbf{p}, \mathbf{d}) = \Big\{(R, p_+) \,|\, p_+ = p + \alpha d \text{ for some } \alpha \geq 0, \; L(R, p_+) = \min_{0 \leq \alpha < \infty} L(R, p + \alpha d)\Big\}. \tag{8}$$

$\mathbf{S^1}$ is a set-valued mapping, since in some cases, there may be many vectors p_+ yielding the minimum (with R fixed).

3) $\mathbf{C^2}$: $\mathbf{SE(n)} \to \mathbf{SE(n)} \times \mathbf{so(n)}$ is a point-to-point mapping defined by

$$\mathbf{C^2}(\mathbf{R}, \mathbf{p}) = (\mathbf{R}, \mathbf{p}, \mathbf{\Omega}) \tag{9}$$

where Ω corresponds to the search direction along the $SO(n)$ component at the point (R, p), expressed as an element of $so(n)$ by left translation (see the examples below).

4) $\mathbf{S^2}$ is a mapping defined on $SE(n) \times so(n)$ that assigns to each point $(R, p, \Omega) \in SE(n) \times so(n)$ the following set in $SE(n)$:

$$\mathbf{S^2}(\mathbf{R}, \mathbf{p}, \mathbf{\Omega}) = \Big\{(R_+, p) \,|\, R_+ = Re^{\Omega\alpha} \text{ for some } 0 \leq \alpha \leq 2\pi, \; L(R_+, p) = \min_{0 \leq \alpha \leq 2\pi} L(Re^{\Omega\alpha}, p)\Big\}. \tag{10}$$

Again, $\mathbf{S^2}$ is a set-valued mapping, since in some cases, there may be many values of R_+ yielding the minimum (with p fixed).

Theorem 2 [Global Convergence of CCD on $SE(n)$]: Let $SE(n)$ have a left-invariant metric space structure, and consider a differentiable objective function $L(R, p)$ on $SE(n)$. Define the solution set

$$\Gamma = \{(R, p) \,|\, \nabla_R L = \nabla_p L = 0\} \tag{11}$$

where ∇_R denotes the gradient of L (regarding p as fixed) with respect to the natural Riemannian metric on $SO(n)$, and $\nabla_p L = \partial L / \partial p$ (regarding R as fixed). Let $\mathbf{A}$ be an algorithm on $SE(n)$ defined as the composition of four maps $\mathbf{A} = \mathbf{S^2} \circ \mathbf{C^2} \circ \mathbf{S^1} \circ \mathbf{C^1}$, where $\mathbf{C^1}$ and $\mathbf{C^2}$ are assumed continuous, and $\mathbf{S^1}$ and $\mathbf{S^2}$ are assumed to yield unique minimum points. We also assume that $\mathbf{A}$ is restricted to points on a compact set of $SE(n)$. Then the algorithm $\mathbf{A}$ is globally convergent.

Proof: Both $\mathbf{S^1}$ and $\mathbf{S^2}$, which are just line searches over the doubly infinite line (as opposed to the semi-infinite line), are clearly closed, and since points are restricted to a compact set, the composite map $\mathbf{A}$ is also closed. The objective function $L(R, p)$ also serves as a continuous descent function for $\mathbf{A}$ with respect to Γ; this is because a search along either $\Re^n$ or $SO(n)$ must either yield a decrease or, by the uniqueness assumption, it cannot change position. Therefore, at a point not in the solution set Γ, either $\nabla_R L$ or $\nabla_p L$ must be nonzero; a search along the corresponding direction must yield a decrease.

The theorem now follows by invoking the general Global Convergence Theorem. ■

Remark 1: As long as we do not consider objective functions with local minima at infinity, the assumption that points are restricted to a compact set is a mild one; the objective functions of interest in this paper are quadratic, and are derived from physical problems with real solutions. In fact, most of the objective functions considered in this paper are coercive, i.e., $L(X)$ approaches infinity as $\|X\| \to \infty$, $X \in SE(3)$, thereby satisfying the compactness requirement. Also, for our algorithms, $\mathbf{C^1}$ and $\mathbf{C^2}$ will be determined by the gradient and Hessian, and can, therefore, both be regarded as continuous mappings.

Remark 2: Supposing $x = (x^1, x^2, x^3)$ are local coordinates for $SO(3)$, the gradient $\nabla_R L$ above can be expressed in local coordinates by $\nabla_R L = G^{-1}(x)(\partial L/\partial x)^T$, where $G(x) = (g_{ij})$ is the 3×3 matrix representation of the Riemannian metric on $SO(3)$. Explicit formulas for calculating the gradient (with respect to the bi-invariant metric) $\nabla_R L$, in terms of both Euler angle and exponential coordinates, are given in [13]. Because the objective functions in this paper are all quadratic, the gradients assume a particularly simple form, and can be calculated without resorting to the general formula as we show below.

From the construction, it should be clear that both steepest descent and Newton's methods produce search directions that decrease the objective function, thereby satisfying one of the requirements for global convergence. In principle, any optimization method on $SO(n)$ and $\Re^n$ that produces descent directions in a continuous fashion, such as appropriate generalizations of the various vector space quasi-Newton methods, can be used in the CCD algorithm.

Example 1: In this first example, for an arbitrary matrix $A \in \Re^{4\times 4}$, we want to find $X \in SE(3)$ that is the closest point to A in some sense. One means of formulating this as an optimization problem on $SE(3)$ is

$$\min_{X\in SE(3)} J(X) = \|X - A\|^2$$

where the Frobenius norm defined by $\|A\|^2 = \operatorname{tr}\{AA^T\}$ is used. The objective function can be simplified to

$$J(X) = \operatorname{tr}\{XX^T - 2XA^T + AA^T\}$$

which is a special case of $\operatorname{tr}\{XNX^TQ - 2XW\}$ that arises in the workpiece localization problem [17]. We ignore the constant AA^T and expanding in terms of the R and p components of $SE(3)$. Specifically, let

$$X = \begin{pmatrix} R & p \\ 0 & 1 \end{pmatrix}, \qquad A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

where $R \in SO(3)$, $p \in \Re^3$, $A_{11} \in \Re^{3\times 3}$, A_{12}, $A_{21}^T \in \Re^{3\times 1}$, $A_{22} \in \Re$. The objective function can then be simplified to

$$\min_{R\in SO(3),\, p\in\Re^3} J(R, p) = -2\operatorname{tr}\{RA_{11}^T\} + p^Tp - 2A_{12}^Tp.$$

Applying CCD on $SE(3)$, the objective function is first minimized about R regarding p as constant, followed by a minimization about p regarding R as constant. These two steps are repeated until convergence occurs.

Minimization on $SO(3)$: The steepest descent algorithm for minimizing about R is given by

$$\begin{aligned} \Omega_k &= \tfrac{1}{2}\left(R_kA_{11}^T - A_{11}R_k^T\right) \\ t_k &= \arg\min_{t\in\Re} J(R_ke^{\Omega_k t}) \\ R_{k+1} &= R_ke^{\Omega_k t_k}. \end{aligned}$$

Minimization on $\Re^3$: We now minimize about p considering R as constant, which was optimized in the previous step. For this special case, R and p are decoupled; from the first-order necessary conditions $(\partial J/\partial p)(R, p) = 0$, the optimal p is $p = A_{12}$.

Example 2: We now consider the more general case of two given arbitrary matrices A, $B \in \Re^{4\times 4}$. The objective is to find $X \in SE(3)$ that minimizes the following objective function:

$$\min_{X\in SE(3)} J(X) = \|AX - XB\|^2$$

where the Frobenius norm defined by $\|A\|^2 = \operatorname{tr}\{AA^T\}$ is again used. In the event that A, B are assumed to be elements of $SE(3)$, the above objective function arises in the problem of calibrating wrist-mounted robotic sensors [14], [17], [21]. The above objective function can be rewritten as follows:

$$\begin{aligned} J(X) = \operatorname{tr}\{&AXX^TA^T - AXB^TX^T \\ &- XBX^TA^T + XBB^TX^T\}. \end{aligned}$$

Since each term is of the form $\operatorname{tr}\{XAX^TB^T\}$, the necessary gradients and Hessian of the objective function can be obtained by repeating the procedure for the following simplified objective function:

$$J(X) = \operatorname{tr}\{XAX^TB^T\}.$$

Let X, A, and B be partitioned in the same manner as in the previous example. The general matrix relation $\operatorname{tr}(ABC) = \operatorname{tr}(CAB) = \operatorname{tr}(BCA)$ for any arbitrary matrices A, B, C in which the products make sense is also useful. Ignoring the constant term, the objective function simplifies to

$$\begin{aligned} J(R, p) = \operatorname{tr}\{&RA_{11}R^TB_{11} \\ &+ R(A_{21}(p^TB_{11} + B_{12}^T) + A_{12}(p^TB_{11} + B_{21}^T))\} \\ &+ A_{22}(p^TB_{11}p + p^T(B_{12} + B_{21})). \end{aligned}$$

Minimization on $SO(3)$: The objective function is first minimized with respect to R regarding p as constant. After ignoring the constant terms, the objective function can be rewritten as follows:

$$J(R) = \operatorname{tr}\{RA_{11}R^TB_{11} - 2RC\}$$

where $C = -(1/2)(A_{21}(p^TB_{11} + B_{12}^T) + A_{12}(p^TB_{11} + B_{21}^T))$ is constant. The corresponding steepest descent algorithm for minimizing with respect to R is

$$\begin{aligned} \Omega_k &= A_{11}R_k^TB_{11}R_k - R_k^TB_{11}R_kA_{11} - CR_k + R_k^TC^T \\ t_k &= \arg\min_{t\in\Re} J(R_ke^{\Omega_k t}) \\ R_{k+1} &= R_ke^{\Omega_k t_k}. \end{aligned}$$

Newton's method on $SO(3)$ can also be applied; the corresponding algorithm is (see [16])

$$\begin{aligned}
T &= A_{11}R_k^T B_{11}R_k + (A_{11}R_k^T B_{11}R_k)^T - CR_k - R_k^T C^T \\
M &= -2R_k^T B_{11}R_k \\
S &= \sum_{i=1}^{m} 2\{(A_{11}R_k^T B_{11}R_k)^T - A_{11}R_k^T B_{11}R_k \\
&\qquad + R_k^T C^T - CR_k\} \\
H &= \sum_{i=1}^{m} \{B_{11}A_{11} + A_{11}B_{11} - \mathrm{tr}(B_{11})A_{11} - \mathrm{tr}(A_{11})B_{11} \\
&\quad + (\mathrm{tr}(B_{11})\mathrm{tr}(A_{11}) - \mathrm{tr}(B_{11}A_{11}))I + \mathrm{tr}(T)I - T\} \\
\Omega_k &= \left[H^{-1}\vec{S}\right] \\
t_k &= \arg\min_{t\in\Re} J(R_k e^{\Omega_k t}) \\
R_{k+1} &= R_k e^{\Omega_k t_k}.
\end{aligned}$$

Minimization on $\Re^3$: We now minimize about p considering R as constant. Set the optimal value of R from the previous step to be R_+. The objective function can then be rewritten as follows:

$$\begin{aligned}
J(p) &= A_{22}p^T B_{11}p \\
&\quad + p^T\{A_{22}(B_{12} + B_{21}) + B_{11}R_+(A_{12} + A_{21})\}.
\end{aligned}$$

If $A_{22} > 0$ and $B_{11} > 0$, or $A_{22} < 0$ and $B_{11} < 0$, then there exists a p which minimizes the objective function. From the first-order necessary conditions

$$\begin{aligned}
\frac{\partial J(R_+, p)}{\partial p} &= 2A_{22}B_{11}p + \{A_{22}(B_{12} + B_{21}) \\
&\quad + B_{11}R_+(A_{12} + A_{21})\} = 0
\end{aligned}$$

from which the optimal p is evaluated to be

$$p = -\frac{1}{2}\left\{B_{11}^{-1}(B_{12} + B_{21}) + \frac{1}{A_{22}}R_+(A_{12} + A_{21})\right\}.$$

Simulation Results: We now present simulation results for minimizing the objective function using the CCD algorithm on $SE(3)$. We use both the steepest descent algorithm and Newton's method on $SO(3)$ to optimize with respect to R. Alternatively, for benchmarking purposes, we parameterize the objective function using exponential coordinates on $SO(3)$, and minimize the function using the standard steepest descent algorithm and Newton's method in Euclidean space. In this case, the objective function can be regarded as the following function defined in Euclidean space $\Re^6$ with $R = e^{[w]}$:

$$\begin{aligned}
J(R, p) &= \mathrm{tr}\left\{e^{[w]}A_{11}e^{-[w]}B_{11}\right. \\
&\quad \left. + e^{[w]}(A_{21}(p^T B_{11} + B_{12}^T) + A_{12}(p^T B_{11} + B_{21}^T))\right\} \\
&\quad + A_{22}(p^T B_{11}p + p^T(B_{12} + B_{21})).
\end{aligned}$$

As with other local coordinate parameterizations of $SO(3)$, for this particular parameterization, obtaining analytic expressions for the gradient and Hessian of the objective function becomes very difficult; we resort to numerical approximations of these quantities.

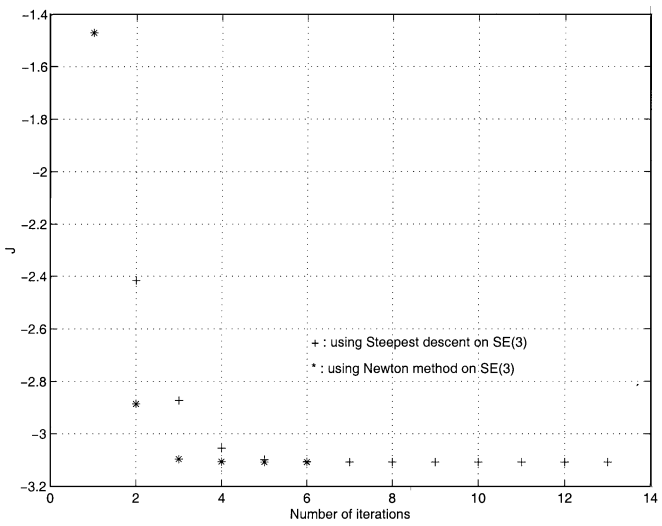


Fig. 1. Convergence of the objective function using coordinate descent.

TABLE I
SIMULATION RESULTS FOR OBJECTIVE FUNCTION ON $SE(3)$

	Steepest descent		Newton's method	
	$SE(3)$	$\Re^6$	$SE(3)$	$\Re^6$
Time (sec)	0.2103	0.8031	0.1032	0.5488
Iterations	23	38	8	5

The simulations are performed using MATLAB, on a Pentium 700 MHz PC with 128 MB of memory. The iteration is stopped at the kth step when the convergence criterion $||R_{k-1} - R_k|| + ||p_{k-1} - p_k|| < 10^{-4}$ is satisfied. The values for A and B used in this simulation are given by algorithms, while Fig. 1 shows the average convergence behavior.

$$A = \begin{pmatrix} 0.950 & 0.486 & 0.457 & 0.952 \\ 0.231 & 0.891 & 0.019 & 0.231 \\ 0.607 & 0.762 & 0.821 & 0.607 \\ 0.486 & 0.891 & 0.762 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 0 & 0 & 0.922 \\ 0 & 1 & 0 & 0.738 \\ 0 & 0 & 1 & 0.176 \\ 0.445 & 0.615 & 0.792 & 1 \end{pmatrix}.$$

Table I shows the average elapsed time to convergence and the number of iterations for the respective algorithms.

We note that the speed of the algorithms defined on $SE(3)$ are faster than those of the algorithms defined on $\Re^6$. In the case of steepest descent, the number of iterations for the CCD algorithm on $SE(3)$ is much smaller than that of the algorithm on $\Re^6$. Although the opposite is true in the case of Newton's method, Newton's method on $\Re^6$ is slower than the corresponding on CCD algorithm on $SE(3)$, due to the computational burden of numerically approximating the gradient and Hessian of the objective function in $\Re^6$.

IV. Camera Calibration

As another application of the CCD algorithm on the Euclidean group, in this section, we address the camera calibration problem. Camera calibration is the process of determining the camera's geometric and optical characteristics (the intrinsic parameters) and the 3-D position and orientation of the camera frame relative to a certain world coordinate system (the extrinsic parameters) [11], [19].

One of the first works to address this problem was [19], which solved the problem by breaking it into four steps. In the first step, the position and orientation of the camera are calibrated, while in the remaining steps, the optimal intrinsic parameters are sequentially obtained in an *ad hoc* fashion. The solutions are obtained by solving a set of constraint equations associated with a known monoview coplanar set of points rather than by optimization. Ma [11] addressed the problem of camera self-calibration using only information from image measurements, without any *a priori* knowledge about the observed scene. In this approach, an uncalibrated camera is considered as a calibrated camera in a space with an unknown metric, and camera calibration is regarded as the process of recovering this unknown metric.

In this paper, we follow Tsai's [19] original problem formulation. Observe that there is some variation in mathematically formulating the camera calibration problem due to different choices of intrinsic parameters. In our approach, the camera calibration problem is framed in what we believe is its most natural form: as an optimization problem on $SE(3) \times \Re^4$, in which all the parameters to be determined are treated uniformly. With the given data set $(n_i = \Re^3, q_i = \Re^2)_{i=1,\ldots,m}$, the objective function can be defined succinctly as follows:

$$\min_{X \in SE(3),\, y \in \Re^4} J(X, y) = \sum_{i=1}^{m} \operatorname{tr}\{X N_i X^T Q_i(y)\} \qquad (12)$$

where

$$N_i = \begin{pmatrix} n_i \\ 1 \end{pmatrix} \begin{pmatrix} n_i^T & 1 \end{pmatrix} = \begin{bmatrix} n_i n_i^T & n_i \\ n_i^T & 1 \end{bmatrix}$$

$$Q_i(y) = \begin{bmatrix} 1 & 0 & -\phi_1 & 0 \\ 0 & 1 & -\phi_2 & 0 \\ -\phi_1 & -\phi_2 & \phi^T\phi & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\phi(y, q_i) = \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} = \frac{1 + y_2 \tilde{q}_i^T S^2 \tilde{q}_i}{y_1} S\tilde{q}_i$$

$$\tilde{q}_i = q_i - c$$

$$c = \begin{pmatrix} c_x \\ c_y \end{pmatrix}$$

$$S = \begin{bmatrix} \dfrac{d_x}{y_3} & 0 \\ 0 & \dfrac{d_y}{y_4} \end{bmatrix}$$

$$y = (f, k_c, s_x, s_y)^T.$$

The physical meaning of the above camera parameters are as follows (see Fig. 2):

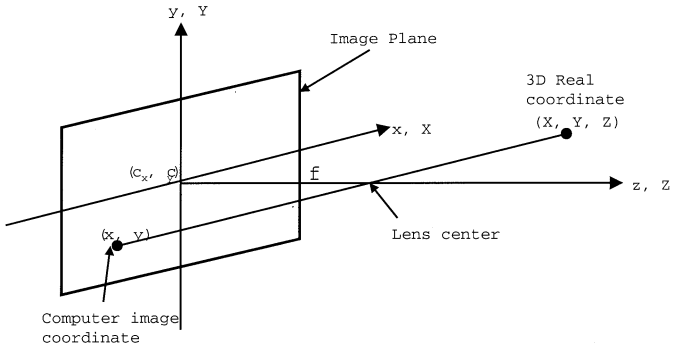


Fig. 2. Basic model of the imaging process.

n_i:	3-D camera coordinates
q_i:	computer image coordinates
$\tilde{q}_i$:	real image coordinates
c_x, c_y:	computer image coordinates for the image plane
d_x, d_y:	center to center distance between adjacent sensor elements in x, y direction
f:	effective focal length
k_c:	lens distortion coefficient
s_x, s_y:	uncertainty scale factor for x, y.

The objective function is similar to that of *Example 2* in the previous section, but with the additional variables $y \in \Re^4$ that represent the intrinsic parameters of the camera. The CCD optimization algorithm on $SE(3)$ therefore needs to be modified accordingly. Rather than minimizing with respect to the three unknowns, $R \in SO(3)$, $p \in \Re^3$, and $y \in \Re^4$ sequentially using CCD, we optimize p and y in the same step. The objective function can be rewritten in the following form:

$$\begin{aligned} J(R, p, y) &= \sum_{i=1}^{m} \operatorname{tr}\left\{ \begin{pmatrix} R & p \\ 0 & 1 \end{pmatrix} \begin{pmatrix} n_i n_i^T & n_i \\ n_i^T & 1 \end{pmatrix} \right. \\ &\qquad \left. \begin{pmatrix} R^T & 0 \\ p^T & 1 \end{pmatrix} \begin{pmatrix} \hat{Q}_i(y) & 0 \\ 0 & 0 \end{pmatrix} \right\} \\ &= \sum_{i=1}^{m} \operatorname{tr}\{R n_i n_i^T R^T \hat{Q}_i(y) + 2 R n_i p^T \hat{Q}_i(y)\} \\ &\quad + p^T \hat{Q}_i(y) p. \end{aligned}$$

The camera calibration problem can, therefore, be reduced to the following optimization problem:

$$\min_{R \in SO(3),\, (p,\, y) \in \Re^7} \sum_{i=1}^{m} \operatorname{tr}\{R n_i n_i^T R^T \hat{Q}_i(y) + 2 R n_i p^T \hat{Q}_i(y)\} + p^T \hat{Q}_i(y) p$$

where $\hat{Q}_i(y) \in \Re^{3\times 3}$ is symmetric.

1) Minimization on $SO(3)$: To minimize the objective function on $SO(3)$, p and y are regarded as constants. Ignoring constant terms, the objective function becomes

$$\min_{R \in SO(3)} J(R) = \sum_{i=1}^{m} \operatorname{tr}\{R \hat{N}_i R^T \hat{Q}_i - 2 R \hat{W}_i\}$$

where $\hat{N}_i = n_i n_i^T$, $\hat{W}_i = -n_i p^T \hat{Q}_i$ are constant. We apply Newton's method to perform the optimization with respect to R

$$\begin{cases} T_i = \hat{N}_i R_k^T \hat{Q}_i R_k + (\hat{N}_i R_k^T \hat{Q}_i R_k)^T - \hat{W}_i R_k - R_k^T \hat{W}_i^T \\ M_i = -2 R_k^T \hat{Q}_i R_k \\ S = \sum_{i=1}^{m} 2\{(\hat{N}_i R_k^T \hat{Q}_i R_k)^T - \hat{N}_i R_k^T \hat{Q}_i R_k + R_k^T \hat{W}_i^T - \hat{W}_i R_k\} \\ H = \sum_{i=1}^{m} \{\hat{M}_i \hat{N}_i + \hat{N}_i \hat{M}_i - (\operatorname{tr} \hat{M}_i)\hat{N}_i - (\operatorname{tr} \hat{N}_i)\hat{M}_i \\ \qquad + (\operatorname{tr} \hat{M}_i \operatorname{tr} \hat{N}_i - \operatorname{tr} \hat{M}_i \hat{N}_i) I + (\operatorname{tr} T_i) I - T_i\} \\ \Omega_k = \left[H^{-1} \vec{S} \right] \\ t_k = \arg\min_{t \in \Re} J(R_k e^{\Omega_k t}) \\ R_{k+1} = R_k e^{\Omega_k t_k}. \end{cases}$$

For comparison purposes, the steepest descent method is also used

$$\begin{cases} \Omega_k = \sum_{i=1}^{m} \{\hat{N}_i R_k^T \hat{Q}_i R_k - R_k^T \hat{Q}_i R_k \hat{N}_i - \hat{W}_i R_k + R_k^T \hat{W}_i^T\} \\ t_k = \arg\min_{t \in \Re} J(R_k e^{\Omega_k t}) \\ R_{k+1} = R_k e^{\Omega_k t_k}. \end{cases}$$

2) Minimization on $\Re^7$: We now minimize the objective function in the Euclidean space $\Re^7$ by fixing R to the constant value R_+ obtained in the previous step. The objective function can be rewritten as

$$\min_{(p,\, y) \in \Re^7} \sum_{i=1}^{m} \{p^T \hat{Q}_i(y) p + 2 p^T \hat{Q}_i(y) R_+ n_i + n_i^T R_+^T \hat{Q}_i(y) R_+ n_i\}.$$

We can construct the following steepest descent algorithm using the gradient $((\partial J/\partial p), (\partial J/\partial y))$:

$$\begin{cases} \dfrac{\partial J}{\partial p} = \sum_{i=1}^{m} 2\{p^T \hat{Q}_i(y) + n_i^T R_+^T \hat{Q}_i(y)\} \\ \dfrac{\partial J}{\partial y} = \left(\dfrac{\partial J}{\partial y_1}, \dfrac{\partial J}{\partial y_2}, \dfrac{\partial J}{\partial y_3}, \dfrac{\partial J}{\partial y_4} \right) \\ t_k = \arg\min_{t \in \Re} J\left(p_k - \dfrac{\partial J}{\partial p} t,\, y_k - \dfrac{\partial J}{\partial y} t \right) \\ (p_{k+1},\, y_{k+1}) = \left(p_k - \dfrac{\partial J}{\partial p} t_k,\, y_k - \dfrac{\partial J}{\partial y} t_k \right) \end{cases}$$

where

$$\frac{\partial J}{\partial y_k} = \sum_{i=1}^{m} \left\{ p^T \frac{\partial \hat{Q}_i(y)}{\partial y_k} p + 2 p^T \frac{\partial \hat{Q}_i(y)}{\partial y_k} R_+ n_i + n_i^T R_+^T \frac{\partial \hat{Q}_i(y)}{\partial y_k} R_+ n_i \right\}$$

TABLE II
SIMULATION RESULTS FOR OBJECTIVE FUNCTION IN $SE(3) \times \Re^4$

	Steepest descent		Newton's method	
	$SE(3) \times \Re^4$	$\Re^{10}$	$SE(3) \times \Re^4$	$\Re^{10}$
$m = 4$: time (s)	1.595	3.201	1.441	2.289
$m = 4$: # iter.	41	50	39	23
$m = 10$: time (s)	1.705	4.583	1.113	n/a
$m = 10$: # iter.	12	18	4	n/a
$m = 100$: time (s)	10.158	15.603	7.991	n/a
$m = 100$: # iter.	8	6	8	n/a

$$\frac{\partial \hat{Q}_i(y)}{\partial y_k} = \begin{bmatrix} 0 & 0 & -\dfrac{\partial \phi_1}{\partial y_k} \\ 0 & 0 & -\dfrac{\partial \phi_2}{\partial y_k} \\ -\dfrac{\partial \phi_1}{\partial y_k} & -\dfrac{\partial \phi_2}{\partial y_k} & 2\phi^T \dfrac{\partial \phi}{\partial y_k} \end{bmatrix}$$

$$\frac{\partial \phi}{\partial y_k} = \begin{pmatrix} \dfrac{\partial \phi_1}{\partial y_k} \\ \dfrac{\partial \phi_2}{\partial y_k} \end{pmatrix}$$

$$\frac{\partial \phi}{\partial y_1} = -\frac{1 + y_2 \tilde{q}_i^T S^2 \tilde{q}_i}{y_1^2} S \tilde{q}_i$$

$$\frac{\partial \phi}{\partial y_2} = \frac{\tilde{q}_i^T S^2 \tilde{q}_i}{y_1} S \tilde{q}_i$$

$$\frac{\partial \phi}{\partial y_3} = \frac{2 y_2 \tilde{q}_i^T S^2 \frac{\partial S}{\partial y_3} \tilde{q}_i}{y_1} S \tilde{q}_i + \frac{1 + y_2 \tilde{q}_i^T S^2 \tilde{q}_i}{y_1} \frac{\partial S}{\partial y_3} \tilde{q}_i$$

$$\frac{\partial \phi}{\partial y_4} = \frac{2 y_2 \tilde{q}_i^T S^2 \frac{\partial S}{\partial y_4} \tilde{q}_i}{y_1} S \tilde{q}_i + \frac{1 + y_2 \tilde{q}_i^T S^2 \tilde{q}_i}{y_1} \frac{\partial S}{\partial y_4} \tilde{q}_i$$

$$\frac{\partial S}{\partial y_3} = \begin{bmatrix} -\dfrac{d_x}{y_3^2} & 0 \\ 0 & 0 \end{bmatrix}$$

$$\frac{\partial S}{\partial y_4} = \begin{bmatrix} 0 & 0 \\ 0 & -\dfrac{d_x}{y_4^2} \end{bmatrix}.$$

As seen from above, the equations for y are quite complex compared with those for R and p.

3) Simulation Results: We now present simulation results for the objective function using the modified CCD algorithm on $SE(3) \times \Re^4$. Alternatively, for comparison purposes, we minimize the function using the steepest descent and Newton's methods in Euclidean space $\Re^{10}$, by parameterizing R with the exponential coordinates $R = e^{[w]}$. As before, we must resort to numerically calculating the gradient and Hessian of the objective function, which will degrade convergence.

The simulations are performed in MATLAB under the same conditions and convergence criteria as in the previous example. We perform the simulations for a range of data points: 4, 10, and 100 data points. For each set we perform ten trials, in which each trial consists of an application of the four optimization algorithms. The starting points are arbitrarily chosen and varied for each of the ten trials. For each algorithm, we compute the average elapsed time and number of iterations over the ten trials. Table II displays the simulation results, while Fig. 3 shows the average convergence behavior.

Like the results obtained for *Example 2*, the speed of the algorithms defined on $SE(3) \times \Re^4$ are faster than those of the

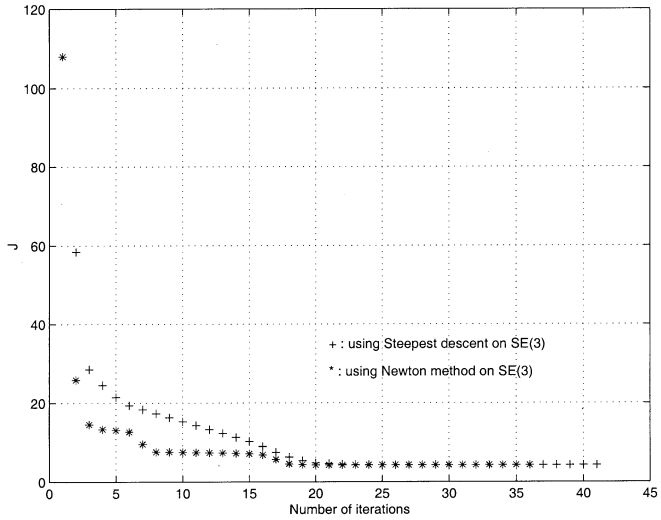


Fig. 3. Quadratic convergence of the objective function using CCD.

algorithms defined on $\Re^{10}$ by approximately a factor of two. Newton's method on $\Re^{10}$ failed to converge, most likely due to numerical stability in evaluating the 10×10 inverse Hessian. Note that the computation time increases with the number of data points; this is due to the number of summation terms in the objective function, which is equal to the number of data points.

It should also be emphasized that what is not reflected in the table is the extreme sensitivity of the local coordinate-based (i.e., $\Re^{10}$) optimization procedures when the iteration ventures near coordinate singularities. The results shown represent, in some sense, a best-case scenario for the local coordinate-based methods. In general, when using local coordinate-based methods, one must always check to see whether the iteration has ventured sufficiently close to a coordinate patch boundary, and if so, transform to another set of local coordinates. In the case of the exponential coordinate parameterization of $SO(3)$, for example, one ensures that $|Tr(R)+1|$ always exceeds some specified threshold ϵ. Such methods are not only cumbersome from a programming standpoint, but also significantly degrade performance, possibly failing to converge in the worst case. The CCD algorithm (and geometric methods in general), on the other hand, does not suffer from any such singularities.

V. Conclusion

In this paper, we have presented the CCD algorithm as an effective means of optimizing quadratic objective functions on $SE(n)$ that arise in a wide range of applications in robotics. By exploiting the fact that $SE(n)$ is the semidirect product of $SO(n)$ and $\Re^n$, we have shown that by cyclically optimizing between these two spaces, global convergence can be guaranteed under a set of mild assumptions. Optimization in $SO(n)$ can furthermore be performed using any of a number of efficient geometric algorithms.

The superior performance of geometric optimization algorithms on compact Lie groups like $SO(n)$ has been amply demonstrated in the literature. Unlike $SO(n)$, however, $SE(n)$ does not possess a bi-invariant Riemannian metric, and as such, the one-parameter subgroups no longer constitute minimal geodesics on $SE(n)$. The straightforward extension of geometric optimization algorithms to $SE(n)$ now becomes much less appealing, due to the difficulty in evaluating the search directions. Instead, the CCD algorithm allows one to take advantage of the extant geometric optimization algorithms on $SO(n)$—computation of gradients and Hessians on $SO(n)$ and $\Re^n$ is much simpler than on $SE(n)$—and provides conditions under which global convergence can be guaranteed.

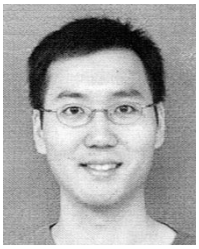
As with Newton's method in $\Re^n$, we expect that employing quasi-Newton methods on $SO(n)$ that do not require explicit calculation of the Hessian should improve performance even further. The algorithm is, moreover, invariant with respect to choice of fixed reference frame, as it should be to satisfy the principle of objectivity. Also, because they do not rely on local coordinates for $SE(n)$, the geometric algorithms of the type considered in the paper are also immune to any problems arising from local coordinate singularities. When done right, local coordinate-based methods involve a significant amount of bookkeeping and coordinate changes that will degrade their performance even more *vis-à-vis* geometric methods.

The examples considered in this paper confirm the algorithmic simplicity, efficiency, and robustness of the CCD algorithm on $SE(n)$. In particular, for the camera calibration problem treated in Section IV, in which the objective function is defined on $SE(3)\times\Re^4$, the corresponding CCD algorithm treats the parameters to be determined in a uniform and consistent manner. This is in contrast to previous algebraic approaches, in which a set of constraint equations derived from physical principles, together with the assumption that certain intrinsic parameter errors are negligible compared to other errors, are solved for the unknown parameters in an *ad hoc* sequence, leading to extremely complicated procedures whose results may depend on the order in which the unknown parameters are solved.

References

[1] D. S. Bayard, "An optimization result with application to optimal spacecraft reaction wheel orientation design," in *Proc. American Control Conf.*, Arlington, VA, June 25–27, 2001.
[2] C. Belta and V. Kumar, "New metrics for rigid body motion interpolation," in *Proc. Ball 2000 Symp.*, Cambridge, U.K., July 2000.
[3] A. M. Bloch and P. Crouch, "Nonholonomic control systems on Riemannian manifolds," *SIAM J. Control*, vol. 37, no. 1, pp. 126–148, 1995.
[4] R. W. Brockett, "Least squares matching problems," *Linear Algebra and Its Applications*, vol. 122-124, pp. 761–777, 1989.
[5] ——, "Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems," *Linear Algebra and Its Applications*, vol. 146, pp. 79–91, 1991.
[6] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*. New York: Springer-Verlag, 1994.
[7] J. Kim, "Numerical optimization on the rotation group," M.S. thesis, School of Mech. and Aerosp. Eng., Seoul Nat. Univ., Seoul, Korea, 1998.
[8] Z. Li, J. Gou, and Y. Chu, "Geometric algorithms for workpiece localization," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 864–878, Dec. 1998.
[9] Q. Lin and J. W. Burdick, "Objective and frame-invariant kinematic metric function for rigid bodies," *Int. J. Robot. Res.*, vol. 19, no. 6, pp. 612–625, June 2000.
[10] D. G. Luenberger, *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1989.
[11] Y. Ma, "A differential geometric approach to computer vision and its application in control," Ph.D. dissertation, Dept. Elec. Eng. and Comp. Sci., Univ. Calif., Berkeley, 2000.
[12] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC, 1994.

[13] F. C. Park, "Optimal kinematic design of mechanisms," Ph.D. dissertation, Div. Appl. Sci., Harvard Univ., Cambridge, MA, June 1991.
[14] F. C. Park and B. J. Martin, "Robot sensor calibration: Solving $AX = XB$ on the Euclidean group," *IEEE Trans. Robot. Automat.*, vol. 10, pp. 717–721, Oct. 1994.
[15] F. C. Park, "Distance metrics on the rigid-body motions with applications to mechanism design," *ASME J. Mech. Des.*, vol. 117, pp. 48–54, Mar. 1995.
[16] F. C. Park, J. Kim, and C. Kee, "Geometric descent algorithms for attitude determination using the global positioning system," *AIAA J. Guid., Control, Dyn.*, vol. 23, no. 1, pp. 26–33, 2000.
[17] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $AX = XB$," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 16–29, Feb. 1989.
[18] S. T. Smith, "Optimization techniques on Riemannian manifolds," in *Proc. Fields Inst. Workshop on Hamiltonian and Gradient Flows, Algorithms, and Control*, 1994.
[19] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3-D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robot. Automat.*, vol. RA-3, pp. 323–344, Aug. 1987.
[20] W. I. Zangwill, *Nonlinear Programming: A Unified Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1969.
[21] H. Zhuang, "A note on hand–eye calibration," *Int. J. Robot. Res.*, vol. 16, no. 5, pp. 725–727, Oct. 1997.



Seungwoong Gwak received the B.S. and M.S. degrees in mechanical engineering from Seoul National University, Seoul, Korea in 1999 and 2001, respectively.

His research interests are in geometric optimization and their applications in robotics. He is currently completing his military service in the armed forces of the Republic of Korea.

Junggon Kim received the B.S. and M.S. degrees in mechanical design and production engineering from Seoul National University, Seoul, Korea in 1996 and 1998, respectively.

He is currently pursuing the Ph.D. degree in the area of multibody systems modeling and simulation. His research interests are in global positioning system (GPS)-based attitude determination, robot motion optimization, and object-oriented multibody systems modeling and simulation. He is currently with the Intelligent Mechatronics Group, Hyundai Heavy Industries Research Laboratories, Seoul, Korea.

Frank Chongwoo Park (M'89) received the B.S. degree in electrical engineering and computer science from Massachusetts Institute of Technology, Cambridge, in 1985, and the S.M. and Ph.D. degrees in applied mathematics from Harvard University, Cambridge, MA in 1991.

From 1991 to 1995, he was an Assistant Professor of Mechanical and Aerospace Engineering at the University of California, Irvine. Since 1995, he has been with the School of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Korea, where he is currently an Associate Professor. He has held visiting positions at the IBM T. J. Watson Research Center (1988) and the Courant Institute of Mathematical Sciences (2001–2002). His research interests are in the broad areas of mathematical systems theory, robotics, and related areas of applied mathematics.