Sparse Second-Order Cone Programming for 3D Reconstruction

Hyun-Jung Lee, SangWook Lee, Yongduek Seo
{whitetobi,slee,yndk}@sogang.ac.kr
Department of Media Technology, Sogang University, Korea

Abstract

This paper presents how to minimize the second-order cone programming problem occurring in the 3D reconstruction of multiple views. The L_∞-norm minimization is done by a series of the minimization of the maximum infeasibility. Since the problem has many inequality constraints, we have to adopt methods of the interior point algorithm, in which the inequalities are sequentially approximated by log-barrier functions. An initial feasible solution is found easily by the construction of the problem. Actual computing is done by an iterative Newton-style update. When we apply the interior point method to the problem of reconstructing the structure and motion, every Newton update requires to solve a very large system of linear equations. We show that the sparse bundle-adjustment technique can be utilized in the same way during the Newton update, and therefore we obtain a very efficient computation.

1 Introduction

In geometric vision problems, the L_∞ norm minimization method has been issued because it can compute the global optimum instead of a local one [1, 2]. Many studies introduce the L_∞ applications: one-dimensional cameras in robotics application[3], motion computation [4], dealing with outliers [5, 6], increasing the computational efficiency [7, 8], and tracking a deformable surface [9], and so on. In addition, the L_∞ method is found to solve a more general problem dealing with the rotation parameters together with the branch-and-bound technique. Hartley and Kahl [10] solves the two view motion problem under the sense of global optimality. On the other hand, [11] or [12] try to solve a very large scale structure and motion problem. Such as [10], in the near future, a multi-view reconstruction problem has a globally optimal solution by extending.

An advantage of the L_∞ formulations is that any numerical method will result in the same global minimum in principle since the objective function is convex. Therefore, we note that the (quasi-) convexity and simplicity in implementation are the key to the fast growth and expansion of the L_∞ method.

This paper studies a *feasibility test algorithm* with the bisection method given in Algorithm 1. Minimizing any L_∞ error norm using the bisection method has been worked since the paper of Kahl [2]. The feasibility test is the core problem to minimize the maximum residual error (γ in Algorithm 1), but most of the previous works have used some tools such as SeDuMi, and have not considered it enough. An efficient algorithm for the feasibility problem will result in high speed computation of the L_∞ optimization. To reduce the time consuming and computationally demanding, this paper focuses on the feasibility problem for the structure and motion (SAM) under the assumption of known rotation.

Algorithm 1 Bisection method to minimize L_∞ norm

Input: initial upper(U)/lower(L) bounds, tolerance $\epsilon > 0$.

repeat
    $\gamma := (L + U)/2$
    Solve the feasibility problem (12)
    if *feasible* **then** $U := \gamma$ **else** $L := \gamma$
until $U - L \leq \epsilon$

In this paper, we first present how to formulate the feasibility problem of SAM. We show a few computational methods for the feasibility problem and choose the method that minimizes the maximum infeasibility, and provide its mathematical formulation. In fact, the minimization of the maximum infeasibility is a part of the so-called phase-I methods since it is a preliminary stage to check the feasibility constraints of a given convex problem; if the problem is feasible, it finds a feasible solution [13].

Actually, we can minimize the maximum infeasibility numerically with various algorithms. We focus on two methods in this paper: the barrier method and the primal-dual potential reduction interior-point method. Indeed, Newton updating schemes starting from an initial solution are the foundation of two methods. We show that the Newton update in the barrier method has all the same sparse structure as the bundle-adjustment [14, 15]. We also present how to find an initial solution for the barrier method. As mentioned [13], the primal-dual potential reduction method outperforms the barrier method in various areas. However, we introduce an additional linear inequality constraint which destroys the sparse structure compared with the barrier method to get an initial solution easily. We show that the matrix inversion lemma is applicable, allowing us to use all the same sparse matrix technique. In order to implement the application of the two method, it is not necessary to compute an initial point since it has been studying in many geometric vision problems [14]. Therefore, we can use the good quality initial solution to start the optimization. This is another advantage when we use the L_∞ optimization.

We present the feasibility test algorithm for SAM and some preliminary definitions in section 2. In section 3, we discuss the sparse structure for the case of the second-order cone programming. In section 4, we show how to adopt the primal-dual algorithm instead of the barrier method without destroying the sparse structure. Some experimental results are given in Section 5. We bring to a conclusion int the last section.

2 SAM and the Feasibility Problem

Let $[u_{ki1}, u_{ki2}]^\top$ be an image measurement of the i-th point $\mathbf{X}_i$ in 3D through the k-th camera $\mathtt{P}_k = [\mathtt{R}_k|\mathbf{t}_k]$, the residual vector is defined by

$$\mathbf{e}_{ik} = \left[u_{ik1} - \frac{\mathbf{r}_{k1}^\top \mathbf{X}_i + t_{k1}}{\mathbf{r}_{k3}^\top \mathbf{X}_i + t_{k3}} \,,\; u_{ik2} - \frac{\mathbf{r}_{k2}^\top \mathbf{X}_i + t_{k2}}{\mathbf{r}_{k3}^\top \mathbf{X}_i + t_{k3}} \right]^\top , \quad (1)$$

where $\mathbf{r}_{kn}^\top$ is the n-th row vector of $\mathtt{R}_k$, and t_{kn} the n-th component of the vector $\mathbf{t}_k$. N and K are the numbers of 3D points and cameras, respectively. M ($M \le NK$) is the total number of image measurements, and the index set I_M is the set of the (i,k) pairs ($|I_M| = M$).

Before we precede further, the gauge (coordinate system) must be chosen. We select the same gauge as [2]; the last 3D point is set to $[0,0,0]$, and the last component of the translation $t_{K3} = 1$, that is, $\mathbf{t}_K = [t_{K1}, t_{K2}, 1]^\top$. The number of total parameters is then $P = 3(N-1) + 3K - 1$. θ is a column vector of all the unknown parameters

$$\theta = [\mathbf{X}_1, \ldots, \mathbf{X}_{N-1}, \mathbf{t}_1, \ldots, \mathbf{t}_{K-1}, t_{K1}, t_{K2}]^\top, \quad (2)$$

the residual vector $\mathbf{e}_{ik}$ can be written as

$$\mathbf{e}_{ik} = \left[\frac{\mathbf{a}_{ik1}^\top \theta + b_{ik1}}{\mathbf{c}_{ik}^\top \theta + d_{ik}}, \quad \frac{\mathbf{a}_{ik2}^\top \theta + b_{ik2}}{\mathbf{c}_{ik}^\top \theta + d_{ik}} \right]^\top , \quad (3)$$

where $\mathbf{a}_{ikn}, b_{ikn}, n = 1, 2, \mathbf{c}_{ik}$, and d_{ik} are all coefficient vectors and scalars composed of u_{ikn} and the elements of $\mathtt{R}_k$. Note that $\mathbf{a}_{ikn}$ has only five non-zero elements:

$$\mathbf{a}_{ikn}^\top(3i-2:3i) = u_{ikn}\mathbf{r}_{k3}^\top - \mathbf{r}_{kj}^\top \quad (4)$$

$$\mathbf{a}_{ikn}^\top(3(N-1)+3k) = u_{ikn} \quad (5)$$

$$\mathbf{a}_{ikn}^\top(3(N-1)+3k-n) = -1 \quad (6)$$

That is, $\mathbf{a}_{ikn}^\top(3i-2:3i)$ has three coefficients for $\mathbf{X}_i$, $\mathbf{a}_{ikn}^\top(3(N-1)+3k-n)$ has one for t_{kn}, and $\mathbf{a}_{ikn}^\top(3(N-1)+3k)$ has one for t_{k3}. Similarly, non-zero elements in the vector $\mathbf{c}_{ik}$ are:

$$\mathbf{c}_{ik}^\top(3i-2:3i) = \mathbf{r}_{k3}^\top \quad (7)$$

$$\mathbf{c}_{ik}^\top(3(N-1)+3k) = 1 \quad (8)$$

As we know in [7], the L_2 norm $\|\mathbf{e}_{ik}\|_2$ is a quasi-convex function and also is a pseudo-convex function. Its L_∞ norm $\|\mathbf{e}_{ik}\|_\infty$ is also a quasi-convex function because L_1 norm of each of the two functions

$$\left| \frac{\mathbf{a}_{ik1}^\top \theta + b_{ik1}}{\mathbf{c}_{ik}^\top \theta + d_{ik}} \right| \text{ and } \left| \frac{\mathbf{a}_{ik2}^\top \theta + b_{ik2}}{\mathbf{c}_{ik}^\top \theta + d_{ik}} \right| \quad (9)$$

is of quasi-convex [8, 10]. Given a positive constant γ representing the maximum residual allowable, the solution space of θ is given by the intersection of all the constraints:

$$\|\mathbf{e}_{ik}\| \le \gamma \,, \quad \forall ik \in I_M \,. \quad (10)$$

Therefore, the L_∞ error minimization problem is given by:

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & \|\mathbf{e}_{ik}\| \le \gamma, \; \forall ik \in I_M \end{array} \quad (11)$$

To find the optimal solution, we can use the bisection method presented in Algorithm 1, in which the following feasibility test problem is solved repeatedly

$$\begin{array}{ll} \text{find} & \theta \\ \text{subj. to} & \|\mathtt{A}_{ik}\theta + \mathbf{b}_{ik}\| \le \gamma(\mathbf{c}_{ik}^\top \theta + d_{ik}), \forall ik \in I_M \end{array} \quad (12)$$

where $\mathtt{A}_{ik} = [\mathbf{a}_{ik1}, \mathbf{a}_{ik2}]^\top$, a $2 \times P$ matrix, and $\mathbf{b}_i = [b_{ik1}, b_{ik2}]^\top$, a 2 dimensional vector. The feasibility test problem can be computed by a few computation methods. The one we consider in this paper is a method of minimizing the maximum infeasibilities. An auxiliary variable s is introduced to find a strictly feasible solution of the inequalities or determine that none exists:

$$\begin{array}{ll} \min & s \\ \text{s.t.} & \|\mathtt{A}_{ik}\theta + \mathbf{b}_{ik}\| \le \gamma(\mathbf{c}_{ik}{}^\top \theta + d_{ik}) + s, \; \forall ik \in I_M \end{array} \quad (13)$$

Note that the variable s represents a bound on the maximum infeasibility of the inequalities, and the goal of this problem is to drive s below zero. If $s \le 0$ after the minimization, the problem is feasible, and a solution θ may be retrieved. Otherwise, $s > 0$ then the constraint set is infeasible, having an empty intersection.

We can re-write the problem in the standard form of convex optimization using augmented coefficient matrices. Let us construct the $2 \times (P+1)$ matrix $\tilde{\mathtt{A}}_{ik} = [\mathtt{A}_{ik}|\mathbf{0}]$ by augmenting the matrix with a column of zeros, and $P+1$ dimensional vectors $\tilde{\mathbf{c}} = [\gamma \mathbf{c}^\top, 1]^\top$ and $\tilde{\theta} = [\theta^\top, s]^\top$. Then we have

$$\begin{array}{ll} \min & \mathbf{f}^\top \tilde{\theta} \\ \text{s.t.} & \|\tilde{\mathtt{A}}_{ik}\tilde{\theta} + \mathbf{b}_{ik}\| \le \tilde{\mathbf{c}}_{ik}^\top \tilde{\theta} + \gamma d_{ik}, \\ & \tilde{\mathbf{c}}_{ik}^\top \tilde{\theta} + d_{ik} \ge 0 \end{array} \quad (14)$$

where $\mathbf{f} = [\mathbf{0}, 1]^\top$ is a $(P+1)$ vector in which all components are zero except for the last one. Note that the problem is a second-order cone programing if the L_2 norm function is adopted; it is a linear programming if L_∞ residual (9) is used.

3 Second-order cone programming

When we use L_2 norm in (14), we are come up with an SOCP. In this case, the barrier method adopts the *generalized logarithm* ψ of degree two for the log-barrier function

$$\psi(\mathbf{x}) = \log\left(x_3^2 - x_1^2 - x_2^2\right) \quad (15)$$

whose domain is defined by the second-order cone

$$\mathbf{dom}(\mathbf{x}) = \{\mathbf{x} \in \mathbb{R}^3 | \sqrt{x_1^2 + x_2^2} < x_3\} \,. \quad (16)$$

The barrier method minimizes the following unconstrained objective function while iteratively increasing t as before:

$$\min \quad E := t\mathbf{f}^\top \tilde{\theta} - \sum_{ik} \log(e_{ik3}^2 - e_{ik1}^2 - e_{ik2}^2) \quad (17)$$

where $\mathbf{e}_{ik} = [e_{ik1}, e_{ik2}, e_{ik3}]^\top$ is defined by

$$\mathbf{e}_{ik} = [\tilde{\mathbf{a}}_{ik1}^\top \tilde{\theta} + b_{ik1}, \tilde{\mathbf{a}}_{ik2}^\top \tilde{\theta} + b_{ik2}, \tilde{\mathbf{c}}_{ik}^\top \tilde{\theta} + \gamma d_{ik}]^\top \tag{18}$$

Now, let us define $\bar{\mathtt{A}}_{ik}$ of size $3 \times (P+1)$ as:

$$\bar{\mathtt{A}}_{ik} = [\tilde{\mathbf{a}}_{ik1}, \tilde{\mathbf{a}}_{ik2}, \tilde{\mathbf{c}}_{ik}]^\top \ . \tag{19}$$

The objective in (17)is then minimized by a Newton's method, the hessian is now given by

$$\mathtt{H}_{\mathrm{SOCP}} = \sum_{ik} \bar{\mathtt{A}}_{ik}^\top \mathtt{H}_{ik} \bar{\mathtt{A}}_{ik} \tag{20}$$

where $\mathtt{H}_{ik} = \nabla^2 \psi(\mathbf{e}_{ik})$ is of size 3×3. Examining the shape of the hessian matrix, we meet a sparse structure again:

Result 1 *The sparse structure of the hessian (20) is the same as the one developed for the multiview bundle-adjustment in [15, 14]. We have the additional weighting matrices $\mathtt{H}_{ik}$, contrary to [15, 14], but they do not change the sparse shape of the hessians $\mathtt{H}_{\mathrm{SOCP}}$, respectively.*

3.1 Initial solution

A solution $\tilde{\theta}^{(0)}$ can be obtained easily:

$$\theta^{(0)} = \mathbf{0} \text{ and } s^{(0)} = \max\{\|b_{ik}\|_2 - d_{ik}\} + c, \ c > 0. \tag{21}$$

3.2 Stopping criterion

In [13], the stopping criterion for this SOCP is $2M/t < \epsilon$ from the theory of convex optimization of the second-order cone. 2, M, and ϵ represent the degree of the generalized logarithm ψ, the number of inequalities, and the tolerance gap between the actual optimum and the true optimum, respectively.

4 Primal-dual interior-point method

In this section, we introduce a more efficient method for solving our SOCP. It is the primal-dual potential reduction interior-point method of Nesterov and Nemirovsky[16]. The method is used in [17] to solve a variety of application problems as the numerical computation method of SOCP. In general, primal-dual interior-point methods can exhibit better than linear convergence. Therefore, they are often more efficient than the barrier method. This paper presents efficiency of the primal-dual interior-point methods compared with the barrier method in various areas [13]. Our goal in this section is to show that the sparse structure can still be retained by using the matrix inversion lemma.

The primal-dual potential reduction method minimizes both of the primal and dual variables at the same time, compared to the barrier method given in previous sections which updates only the primal variable $\tilde{\theta}$ in minimizing the error function (17). The primal problem (14) is re-written below:

$$\begin{array}{ll} \min & \mathbf{f}^\top \tilde{\theta} \\ \text{s.t.} & \|\tilde{\mathtt{A}}_{ik} \tilde{\theta} + \mathbf{b}_{ik}\|_2 \leq \tilde{\mathbf{c}}_{ik}^\top \tilde{\theta} + \gamma d_{ik}, \ \forall ik \in I_M \end{array} \tag{22}$$

where the linear inequality is omitted since it is superfluous. The dual problem is given by

$$\begin{array}{ll} \max & -\sum_{ik} (\mathbf{b}_{ik}^\top \mathbf{z}_{ik} + \gamma d_{ik} w_{ik}) \\ \text{s.t.} & \sum_{ik} (\tilde{\mathtt{A}}_{ik}^\top \mathbf{z}_i + \tilde{\mathbf{c}}_{ik} w_{ik}) = \mathbf{f} \\ & \|z_i\|_2 \leq w_i, \quad \forall ik \in I_M \end{array} \tag{23}$$

The vectors $\mathbf{z}_{ik} \in \mathbb{R}^2$ and $w_i \in \mathbb{R}$ are the dual optimization variables. $\lambda_{ik}^\top = [\mathbf{z}_{ik}^\top, w_{ik}]$, and $\mathbf{z}$ and $\mathbf{w}$ be the whole set of $\mathbf{z}_{ik}$'s and w_{ik}'s, respectively.

The potential function (24) below of the primal-dual potential reduction method is minimized starting at initial primal and dual solutions $\tilde{\theta}^{(0)}, \mathbf{z}^{(0)}, \mathbf{w}^{(0)}$:

$$\begin{aligned} \varphi(\tilde{\theta}, \mathbf{z}, \mathbf{w}) = & (2M + \nu\sqrt{2M}) \log \eta \\ & + \sum_{ik} (\psi(\mathbf{e}_{ik}) + \psi(\lambda_{ik})) - 2N \log N \end{aligned} \tag{24}$$

where $\nu \geq 1$ is an algorithm parameter, and η is the duality gap the difference between the primal and dual objectives:

$$\eta(\tilde{\theta}, \mathbf{z}, \mathbf{w}) = \mathbf{f}^\top \tilde{\theta} + \sum_{ik} (\mathbf{b}_{ik}^\top \mathbf{z}_{ik} + \gamma d_{ik} w_{ik}) \ . \tag{25}$$

The theory declares that if $\varphi \to -\infty$ then $\eta \to 0$ and $(\tilde{\theta}, \mathbf{z}, \mathbf{w})$ approaches optimality [16].

4.1 Initial solutions

In (21), we already have shown how to find a initial solution $\tilde{\theta}^{(0)}$. An additional linear bound is included in the original problem To find a dual solutions, $\mathbf{z}^{(0)}$ and $\mathbf{w}^{(0)}$ without affecting the problem itself; this is called the big-M procedure. The dual of the modified problem can be computed such as a pair of dual solutions. The modified problem is as follows:

$$\begin{array}{ll} \min & \mathbf{f}^\top \tilde{\theta} \\ \text{s.t.} & \|\tilde{\mathtt{A}}_{ik} \tilde{\theta} + \mathbf{b}_{ik}\|_2 \leq \tilde{\mathbf{c}}_{ik}^\top \tilde{\theta} + \gamma d_{ik}, \ \forall ik \in I_M \\ & \sum_{ik} (\tilde{\mathbf{c}}_{ik} \tilde{\theta} + d_{ik}) \leq M_B \end{array} \tag{26}$$

and its dual is

$$\begin{array}{ll} \max & -\sum_{ik} (\mathbf{b}_{ik}^\top \mathbf{z}_{ik} + d_{ik}(w_{ik} - \beta)) - \beta M_B \\ \text{s.t.} & \sum_{ik} (\tilde{\mathtt{A}}_{ik}^\top \mathbf{z}_i + \tilde{\mathbf{c}}_{ik}(w_{ik} - \beta)) = \mathbf{f} \\ & \|z_i\|_2 \leq w_i, \quad \forall ik \in I_M \\ & \beta \geq 0 \end{array} \tag{27}$$

Note that the main problem is the same as the original one that constant M_B is large enough. In fact, while we optimize this problem, M_B is iteratively increased to keep the bound inactive. Let us put $B = M_B - \sum_{ik} (\tilde{\mathbf{c}}_{ik} \tilde{\theta} + d_{ik})$ and $\tilde{\mathbf{c}}_B = \sum_{ik} \tilde{\mathbf{c}}_{ik}$.

We first solve the linear constraint equation in (27) to find the initial dual solution after setting $v_{ik} = w_{ik} - \beta$; this yields $\mathbf{z}_{ik}^{(0)}$ and v_{ik}. Because the system is under-determined, a least-norm solution can be used. Then, from the differences $\delta_{ik} = \|\mathbf{z}_{ik}^{(0)}\|_2 - v_{ik}$, we can find a strictly feasible solution

$$\beta^{(0)} = \max\{\max\{\delta_{ik}\}, 0\} + c, \quad c > 0 \ , \tag{28}$$

and we have $w_{ik}^{(0)} = v_{ik} + \beta^{(0)}$ as a result.

4.2 Computing the search directions

Both of the primal and dual variables of the primal-dual potential reduction algorithm are updated via a Newton's method. The $\Delta\tilde{\theta}$ is updated by solving the following

$$\mathtt{H}_{\mathrm{pd}}\, \Delta\tilde{\theta} \quad = \quad -\mathbf{g} \tag{29}$$

where $\bar{\mathtt{A}}_{ik}$ is defined in (19), and $\mathbf{g}$ is given by

$$\mathbf{g} = \rho\mathbf{f} + \bar{\mathtt{A}}_{ik}^{\top}\nabla\psi(\mathbf{e}_{ik}) + \tilde{\mathbf{c}}_B/B \tag{30}$$

and the hessian $\mathtt{H}_{\mathrm{pd}}$ is given by

$$\mathtt{H}_{\mathrm{pd}} \quad = \quad \sum_{ik} \bar{\mathtt{A}}_{ik}^{\top}\mathtt{H}_{ik}\bar{\mathtt{A}}_{ik} + \frac{1}{B^2}\tilde{\mathbf{c}}_B\tilde{\mathbf{c}}_B^{\top} \tag{31}$$

$$\quad = \quad \mathtt{H}_{\mathrm{SOCP}} + \frac{1}{B^2}\tilde{\mathbf{c}}_B\tilde{\mathbf{c}}_B^{\top}\ . \tag{32}$$

The dual direction $\Delta\lambda_{ik}$ is then computed using $\Delta\tilde{\theta}$:

$$\Delta\lambda_{ik} = -\rho\lambda_{ik} - \nabla\psi(\mathbf{e}_{ik}) - \nabla^2\psi(\mathbf{e}_{ik})\bar{\mathtt{A}}_{ik}\Delta\tilde{\theta} \tag{33}$$

which gives $\Delta\mathbf{z}_{ik}$ and Δw_{ik}. The outline of the algorithm is thus as follows [17]:

Algorithm 2 Primal-dual potential reduction algorithm

Input: strictly feasible $(\tilde{\theta}, \mathbf{z}, \mathbf{w})$, tolerance $\epsilon > 0$.

1: **repeat**
2: Find primal and dual search directions by computing (32) and (33).
3: *Plane search.* Find $p, q \in \mathbb{R}$ that minimize $\psi(\tilde{\theta} + q\Delta\tilde{\theta}, \lambda_{ik} + q\Delta\lambda_{ik})$
4: *Update.* $\tilde{\theta} := \tilde{\theta} + p\Delta\tilde{\theta}$, $\lambda_i k := \lambda_{ik} + q\Delta\lambda_{ik}$.
5: **until** $\eta(\tilde{\theta}, \mathbf{z}, \mathbf{w}) \leq \epsilon$

4.3 Sparsity in getting $\Delta\tilde{\theta}$

Note that the hessian $\mathtt{H}_{\mathrm{SOCP}}$ has the sparse structure but the last term $\tilde{\mathbf{c}}_B\tilde{\mathbf{c}}_B^{\top}$ has not in (32). A blind computation must deal with the symmetric positive definite but *non-sparse* matrix H_{pd} of size $(P+1)\times(P+1)$, which becomes unmanageable easily in our problem of multi-view SAM. Fortunately, exploiting the matrix inversion lemma allows us to utilize the sparse structure as before. The matrix inversion lemma for our case is as follows:

$$\left(\mathtt{H} + \frac{\tilde{\mathbf{c}}\tilde{\mathbf{c}}^{\top}}{B^2}\right)^{-1} = \mathtt{H}^{-1} - \mathtt{H}^{-1}\tilde{\mathbf{c}}\left(B^2 + \tilde{\mathbf{c}}^{\top}\mathtt{H}^{-1}\tilde{\mathbf{c}}\right)^{-1}\tilde{\mathbf{c}}^{\top}\mathtt{H}^{-1} \tag{34}$$

We can solve the following two linear equations efficiently using the sparse structure of $\mathtt{H}_{\mathrm{SOCP}}$:

$$\mathtt{H}_{\mathrm{SOCP}}\mathbf{u} \quad = \quad \mathbf{g} \tag{35}$$

$$\mathtt{H}_{\mathrm{SOCP}}\mathbf{v} \quad = \quad \tilde{\mathbf{c}}_B \tag{36}$$

from which we have the solution of (29):

$$\Delta\tilde{\theta} = \mathbf{u} - \frac{\mathbf{v}\tilde{\mathbf{c}}_B^{\top}\mathbf{u}}{B^2 + \tilde{\mathbf{c}}_B^{\top}\mathbf{v}}\ . \tag{37}$$

Note that in (37), it consists only of the vector inner products, a scalar division and some additive operations. In addition, solving the two equations (35) and (36) can be done simultaneously.

Result 2 *During the Newton update in the primal-dual potential reduction algorithm, the sparse structure still remains with the help of the matrix inversion lemma (34).*

5 Experiments

In this paper, we show only a very brief performance comparison between two optimizations - with and without exploiting the sparsity. Because the efficiency of the computation with and without the sparsity is in [15, 14].

We implemented the primal-dual potential reduction interior-point algorithm based on the C implementation of [17]. Computation of the sparse Cholesky decomposition was done using the package called CHOLMOD [18, 19, 20, 21, 22, 23].

First, we generated a synthetic data of 200 measurements (20 points in 10 views)and contaminated it with a Gaussian noise. Therefore, the number of parameters for the structure, motion, and the maximum infeasibility was 87; $3\times(20-1)$, $3\times 10-1$, and 1, respectively. We tested two optimizations. The optimization without sparse computation took 24.5 seconds; matrix size was 87×87, and solved for each Newton update. The optimization with sparse computation took 1.8 seconds; matrix size was 30×30, and each Newton update involved 57 inversions of 3×3 matrices. The computation exploiting the sparsity for this instance of input data was 13.6 times faster than the computation without the sparsity. In the next experiment, we increased the measurement data up to 750 (150 points in five views). We repeatedly tested 100 times using sets of randomly generated data. The optimization without sparse computation took 2,230seconds on average; whereas it took 20 seconds with sparse computation, showing 112 $(= 2,239/20)$ times faster execution time on average than the computation without the sparsity.

6 Conclusion

We showed to solve the feasibility test problem with two interior-point algorithms: the barrier method and the primal-dual potential reduction method. The interior-point algorithms were based on the iterations of a Newton update. It was required to deal with a very large system of linear equations for the Newton update to solve the problems of structure and motion using known rotation. We also explained the sparse computation technique. This technique is applicable to LP and SOCP of the L_∞ formulations. The formulation of the feasibility problem that minimizes the maximum infeasibility leads to the sparse structures. Because of the interior-point algorithms, we needed the sparse structure. The interior-point algorithm started from initial solutions that could be obtained easily by problem constructions. *Sparse* computation techniques appropriately devised for each of the problems were used to solve the system of equations, and the sparse structure is very much similar to the one developed for the bundle-adjustment.

In order to reduce the computation time of the bisection algorithm for the L_∞ optimization, it is necessary to incorporate our upgraded low-level computation technique and high-level techniques which was presented in [2] or [8]. These days, we are facing with new algorithms of dealing with rotation parameters under the framework of branch-and-bound [10]. A fast L_∞ optimizer specially designed for the structure and motion problem is necessary to develop a global optimization method with a reasonable computational speed for multiple view reconstruction. For this reason, we hope that our paper should be helpful for a globally optimal structure from motion.

7 Acknowledgement

This research is accomplished as the result of the research project for culture contents technology development supported by KOCCA.

This work was also supported by the IT R&D program of MKE/IITA. [2008-F-030-01, Development of Full 3D Reconstruction Technology for Broadcasting Communication Fusion]

References

[1] R. Hartley and F. Schaffalitzky. L_∞ minimization in geometric reconstruction problems. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[2] Fredrik Kahl. Multiple view geometry and the L_∞-norm. In *Proc. Int. Conf. on Computer Vision*, pages 1002–1009, Beijing, China, 2005.

[3] Kalle Aström, Olof Enquist, Carl Olsson, Fredrik Kahl, and Richard Hartley. An L-infinity approach to structure and motion problems for 1d-vision. In *ICCV'07*, 2007.

[4] K. Sim and R. Hartley. Recovering camera motion using L_∞ minimization. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006.

[5] Hongdong Li. A practical algorithm for L_∞ triangulation with outliers. In *IEEE Internatonal Conference on Computer Vision and Pattern Recognition*, 2007.

[6] K. Sim and R. Hartley. Removing outliers using the L_∞ norm. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006.

[7] Fredrik Kahl Carl Olsson, Anders P. Eriksson. Efficient optimization for L_∞ problems using pseudoconvexity. In *IEEE International Conference on Computer Vision*, 2007.

[8] Yongduek Seo and Richard I. Hartley. A fast method to minimize L_∞ error norm for geometric vision problems. In *IEEE International Conference on Computer Vision*, 2007.

[9] Mathieu Salzmann, Richard Hartley, and Pascal Fua. Convex optimization for deformable surface 3-d tracking. In *IEEE International Conference on Computer Vision*, 2007.

[10] Richard Hartley and Fredrik Kahl. Global optimization through searching rotation space and optimal estimation of the essential matrix. In *ICCV'07*, 2007.

[11] Kai Ni, Drew Steedly, and Frank Dellaert. Out-of-core bundle adjustment for large-scale 3d reconstruction. In *IEEE International Conference on Computer Vision*, 2007.

[12] Daniel Martinec and Tomas Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007.

[13] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Press, 2004.

[14] Richard I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Press, 2004.

[15] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice, LNCS 1883*. Springer-Verlag, 1999.

[16] Y.E. Nesterov and A.S. Nemirovski. *Interior-point polynomial methods in convex programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, 1994.

[17] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.

[18] Timothy A. Davis. *Direct methods for sparse linear systems*. SIAM, 2006.

[19] T. A. Davis and W. W. Hager. Dynamic supernodes in sparse Cholesky update/downdate and triangular solves. Technical Report TR-2006-004, CISE, University of Florida, 2006.

[20] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam. Algorithm 8xx: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. Technical Report TR-2006-005, CISE, University of Florida, 2006.

[21] T. A. Davis and W. W. Hager. Row modifications of a sparse cholesky factorization. *SIAM Journal on Matrix Analysis and Applications*, 26(3):621–639, 2005.

[22] T. A. Davis and W. W. Hager. Multiple-rank modifications of a sparse Cholesky factorization. *SIAM Journal on Matrix Analysis and Applications*, 22(4), 2001.

[23] T. A. Davis and W. W. Hager. Modifying a sparse Cholesky factorization. *SIAM Journal on Matrix Analysis and Applications*, 20(3):606–627, 1999.