Sixth Indian Conference on Computer Vision, Graphics & Image Processing

A Scalable Projective Bundle Adjustment Algorithm using the L_∞ Norm *

Kaushik Mitra and Rama Chellappa
Dept. of Electrical and Computer Engineering
University of Maryland, College Park, MD
{kmitra, rama}@umiacs.umd.edu

Abstract

The traditional bundle adjustment algorithm for structure from motion problem has a computational complexity of $O((m+n)^3)$ per iteration and memory requirement of $O(mn(m+n))$, where m is the number of cameras and n is the number of structure points. The sparse version of bundle adjustment has a computational complexity of $O(m^3+mn)$ per iteration and memory requirement of $O(mn)$. Here we propose an algorithm that has a computational complexity of $O(mn(\sqrt{m}+\sqrt{n}))$ per iteration and memory requirement of $O(max(m,n))$. The proposed algorithm is based on minimizing the L_∞ norm of reprojection error. It alternately estimates the camera and structure parameters, thus reducing the potentially large scale optimization problem to many small scale subproblems each of which is a quasi-convex optimization problem and hence can be solved globally. Experiments using synthetic and real data show that the proposed algorithm gives good performance in terms of minimizing the reprojection error and also has a good convergence rate.

1 Introduction

Bundle adjustment is the procedure for refining a visual reconstruction to produce optimal 3D structure and camera parameters. Explicitly stated it is the problem of estimating structure and camera parameters given an initial estimate and the observed image data. An appropriate cost function to minimize for this problem is some norm of the reprojection error [14]. Reprojection error is the error between the reprojected image points, obtained from the current estimates of camera and structure parameters, and the observed image points.

The L_2 norm of reprojection error is the commonly used cost function. A lot of work has been done on bundle adjustment based on the L_2 norm. Triggs et al [14] provides a good review on this approach. One reason for this choice of the norm is that the cost function becomes a differentiable function of parameters and this allows the use of gradient and Hessian-based optimization methods. Second order methods like Gauss-Newton and Levenberg-Marquardt(LM) algorithms are the preferred methods for bundle adjustment as they have some nice convergence properties near local minimum. The LM algorithm has a computational complexity of $O((m+n)^3)$ per iteration and memory requirement of $O(mn(m+n))$, where m is the number of cameras and n is the number of structure points. However, for the structure from motion problem there exists a sparse version of the LM algorithm that takes advantage of the fact that there are no direct interactions among the cameras. They interact only through the structure. The same is true for the structure points. Using this fact, a computational complexity of $O(m^3+mn)$ per iteration and memory requirement of $O(mn)$ can be obtained. Appendix 6 of [2] provides a good review on this. Computational complexity and memory requirement are very important issues especially when solving problems that involve hundreds of frames and points. This motivates us to look for an algorithm which has lower computational complexity and memory requirement.

Another problem in minimizing the L_2 norm of reprojection error is that it is a highly non-linear and non-convex function of the camera and structure parameters. Even the simpler problem of estimating the structure parameters given the camera parameters and the corresponding image points, known as the triangulation problem, is a nonlinear and non-convex optimization problem. The corresponding cost function might have multiple minima and finding the global minimum is a very difficult problem. The same is true for the problem of estimating the camera parameters given the structure parameters and the corresponding image points. This problem is known as the resection problem.

Another norm that is geometrically[1] and statistically meaningful is the L_∞ norm. Minimizing the L_∞ norm is

*This work was prepared through collaborative participation in the Advanced Decision Architectures Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0009.

[1]For a discussion on geometrical significance, please see [5].

978-0-7695-3476-3/08 $25.00 © 2008 IEEE
DOI 10.1109/ICVGIP.2008.51

the same as minimax estimation in statistics. Apart from geometrical and statistical significance, the L_∞ norm of reprojection error has a nice analytical form for some problems. Specifically for the triangulation and resection problems, it is a quasi-convex function of the unknown parameters. A quasi-convex function has the property that any local minimum is also a global minimum. The global minimum can be obtained using a bisection algorithm ([3],[5]). Further, each step of the bisection algorithm can be solved by checking for the feasibility of a second order cone programming problem(SOCP) [3],[5], for which efficient software packages such as SeDuMi [11] are readily available.

The availability of efficient means of finding the global solution for the triangulation and resection problems in the L_∞ norm prompted us to look for bundle adjustment algorithms using the same norm. Joint structure and camera parameter estimation in the L_∞ norm is not a quasi-convex optimization problem because of the non-linear coupling of structure and camera parameters. However if we fix one of the unknowns, say structure, and optimize over the camera parameters then we are back to the original problem of L_∞ resection. The next step would be to fix the camera parameters and optimize over the structure. These two steps should be iterated till convergence. This algorithm is an instance of alternation algorithms and more specifically of resection-intersection algorithms in bundle adjustment [14]. In this discussion, intersection and triangulation are synonymous.

The proposed algorithm, using the L_∞ norm, has two advantages. One, fixing one of the unknown set of parameters, say the structure parameters, the camera parameters estimation problem can be solved for each camera separately, which effectively reduces the high-dimensional problem to many low-dimensional subproblems. The same is true when the camera parameters are fixed and the structure parameters are estimated. Hence a high-dimensional parameter estimation problem transforms into many low-dimensional subproblems. The second advantage is that the subproblems that we have to solve are all quasi-convex optimization problems whose global minimum can be efficiently found.

Our algorithm is a projective bundle adjustment algorithm and so the triangulation and resection subproblems have to be solved in the projective space. In [3] and [5] the L_∞ triangulation problem is solved in the Euclidean/affine space where the optimization is done over the convex region 'in front of' all the cameras that the point is visible in. This region is well defined in Euclidean/affine space but not in the projective space. The search for this region results in an increase in the computational cost for the projective triangulation problem [4]. The same is true for the projective resection problem. We have avoided these computations by initializing our algorithm using a quasi-affine reconstruction. Quasi-affine reconstruction can be easily obtained from an initial projective reconstruction by solving a linear programming problem.

There are resection-intersection algorithms based on the L_2 norm. The algorithms proposed by Chen et al [1] and Mahamud et al [9] are some examples of this. These algorithms have almost the same computational complexity as our algorithm but they are based on minimizing algebraic errors, which are approximations of the actual L_2 reprojection error. These approximations make them susceptible to non-physical solutions [9]. Also in our experiments we have found that though the algebraic cost based algorithms converge in the algebraic cost, they may not converge in the L_2 reprojection error cost.

The organization of the paper is as follows: In section 1.1, we provide some necessary background on solving the triangulation and resection problems in the L_∞ framework. In section 2, we discuss the proposed L_∞ bundle adjustment algorithm. In section 3, we compare the computational complexity and memory requirements of our algorithm with the L_2 bundle adjustment algorithm and the L_2 based resection-intersection algorithms. Section 4, experimentally evaluates our algorithm for convergence, computational complexity, robustness to noise with appropriate comparison to other algorithms.

1.1 Background: geometric reconstruction problems using L_∞ norm

Here we give a very brief overview of the problem of minimizing the L_∞ norm for Euclidean/affine triangulation and resection problems. For more details see Kahl [3] and Ke et. al [5]. We begin with the definition of a quasi-convex function since the triangulation and resection cost function reduces to this form.

Definition 1. A function f(X) is a quasi-convex function if its sublevel sets are convex.

1.1.1 Triangulation

Let $P_i = (a_i^T; b_i^T; c_i^T), i = 1, 2, ..., M$ be the projection matrices for M cameras and $(u_i, v_i), i = 1, 2, ..., M$ be the images of the unknown 3D point X in these M cameras. The problem is to estimate X given P_i and (u_i, v_i). Let $\tilde{X}$ be the homogeneous coordinate of X i.e, $\tilde{X} = (X; 1)$. Then the reprojected image point, in camera i, is given by $\left(\frac{a_i^T\tilde{X}}{c_i^T\tilde{X}}, \frac{b_i^T\tilde{X}}{c_i^T\tilde{X}}\right)$ and the L_2 norm of reprojection error function is given by:

$$\begin{aligned} f_i(X) &= \left\| \left(\frac{a_i^T\tilde{X}}{c_i^T\tilde{X}} - u_i, \frac{b_i^T\tilde{X}}{c_i^T\tilde{X}} - v_i \right) \right\|_2 \\ &= \left\| \left(\frac{a_i^T\tilde{X} - u_i c_i^T\tilde{X}}{c_i^T\tilde{X}}, \frac{b_i^T\tilde{X} - v_i c_i^T\tilde{X}}{c_i^T\tilde{X}} \right) \right\|_2 \end{aligned} \tag{1}$$

In Euclidean triangulation, the fact that X is in front of camera i is expressed by: $c_i^T \tilde{X} > 0$ (also known as cheirality constraint). With this constraint, we have:

$$\begin{aligned} f_i(X) &= \frac{\left|\left|a_i^T \tilde{X} - u_i c_i^T \tilde{X}, b_i^T \tilde{X} - v_i c_i^T \tilde{X}\right|\right|_2}{c_i^T \tilde{X}} \\ &= \frac{p_i(X)}{q_i(X)} \end{aligned} \tag{2}$$

$p_i(X)$ is a convex function because it is a composition of a convex function (norm) and an affine function. $q_i(X)$ is an affine function. Functions of the form of $f_i(X)$ are quasi-convex (for proof, see [3],[5]).

The L_∞ norm of reprojection error is

$$F_\infty(X) = \max_i f_i(X) \tag{3}$$

which is again a quasi-convex function, as point-wise maximum of quasi-convex functions is also quasi-convex.(for proof, see [3],[5]).

Minimization of the quasi-convex function F_∞ can be done using a bisection algorithm in the range of F_∞ ([3], [5]). One step in the bisection algorithm involves solving the following feasibility problem:

$$\text{find} \quad X \quad \text{s.t.} \quad X \in S_\alpha \tag{4}$$

where S_α is the alpha sub-level set of $F_\infty(X)$ with the cheirality constraint.

For triangulation,

$$\begin{aligned} S_\alpha &= \{X | f_i(X) \leq \alpha, q_i(X) > 0, \forall i\} \\ &= \{X | p_i(X) - \alpha q_i(X) \leq 0, q_i(X) > 0, \forall i\} \end{aligned} \tag{5}$$

S_α is a convex set and hence we have to solve a convex feasibility problem. Moreover, since $p_i(X)$ is a L_2 norm, this problem is a second order cone programming problem which can be solved efficiently using software packages like Sedumi [11].

1.1.2 Resection

Here we are given N 3D points $X_i, i = 1, 2, ..., N$ and their corresponding image points $(u_i, v_i), i = 1, 2, ..., N$. The problem is to estimate the camera projection matrix P. Again, the L_∞ reprojection error is a quasi-convex function of the unknown camera parameters and the global minimum can be obtained as in the triangulation case [3],[5].

2 The L_∞ projective bundle adjustment algorithm

For L_∞ projective bundle adjustment, we propose an iterative algorithm based on the principle of resection-intersection. We partition the unknown structure and camera parameters into two separate sets and minimize the L_∞ norm of reprojection error over one set of parameters while keeping the other set fixed. In the resection step, the minimization is done over the camera parameters while keeping the structure parameters fixed and in the intersection step, the optimization is done over the structure parameters while keeping the camera parameters fixed. These resection-intersection steps are iterated many times till the algorithm converges.

The resection and intersection step of the proposed algorithm is still a high-dimensional optimization problem. In section 2.1, we show that how these two steps can be further simplified by solving a large number of small optimization problems. In section 2.2, we discuss the correct way to initialize our algorithm. Convergence is another important issue for any iterative method. In section 2.3, we show the convergence of our algorithm in the L_∞ norm of reprojection error.

2.1 Decoupling

Consider the intersection step of the algorithm, where camera parameters are fixed and minimization of the L_∞ norm of reprojection error is done over the structure parameters. Let $P^j, j = 1, 2, ..., M$ be the given projection matrices of M cameras and $X_i, i = 1, 2, ..., N$ be the N 3D points, which are to be estimated. Let f_i^j be the L_2 norm of reprojection error for the i-th 3D point imaged in the j-th camera.

The L_∞ norm of reprojection error is:

$$\begin{aligned} F_\infty(X_1, X_2, ..., X_N) &= \max_{i,j} f_i^j(X_1, X_2, \ldots, X_N) \\ &= \max_i \max_j f_i^j(X_i) \\ &= \max_i f_{\infty,i}(X_i) \end{aligned} \tag{6}$$

where,

$$f_{\infty,i}(X_i) = \max_j f_i^j(X_i) \tag{7}$$

Theorem 1. *The problem of minimizing the L_∞ norm of reprojection error over the joint structure $(X_1, X_2, \ldots, X_N)$ can be solved by minimizing, independently, the L_∞ norm of reprojection error corresponding to each 3D structure X_i, i.e. suppose $(\widehat{X_1}, \widehat{X_2}, \ldots, \widehat{X_N})$ solves the joint structure problem and $(\widetilde{X_1}, \widetilde{X_2}, \ldots, \widetilde{X_N})$ solves the independent problems of minimizing $f_{\infty,1}(X_1), f_{\infty,2}(X_2), \ldots, f_{\infty,N}(X_N)$ respectively, then*

$$F_\infty(\widehat{X_1}, \widehat{X_2}, \ldots, \widehat{X_N}) = F_\infty(\widetilde{X_1}, \widetilde{X_2}, \ldots, \widetilde{X_N}) \tag{8}$$

Proof. By definition (see equation 6),

$$F_\infty(\widetilde{X_1}, \widetilde{X_2}, ..., \widetilde{X_N}) = \max_i f_{\infty,i}(\widetilde{X_i})$$

$$F_\infty(\widehat{X_1}, \widehat{X_2}, \ldots, \widehat{X_N}) = \max_i f_{\infty,i}(\widehat{X_i})$$

Now, $\widetilde{X_i}$ solves $\min_{X_i} f_{\infty,i}(X_i)$ therefore,

$$f_{\infty,i}(\widetilde{X_i}) \leq f_{\infty,i}(X_i) \qquad \forall X_i$$

implies,

$$f_{\infty,i}(\widetilde{X_i}) \leq f_{\infty,i}(\widehat{X_i})$$

This is true for all i, therefore

$$\max_i f_{\infty,i}(\widetilde{X_i}) \leq \max_i f_{\infty,i}(\widehat{X_i})$$

i.e.,

$$F_\infty(\widetilde{X_1}, \widetilde{X_2}, \ldots, \widetilde{X_N}) \leq F_\infty(\widehat{X_1}, \widehat{X_2}, \ldots, \widehat{X_N})$$

but since

$$F_\infty(\widehat{X_1}, \widehat{X_2}, \ldots, \widehat{X_N}) = \min_{(X_1,\ldots,X_N)} F_\infty(X_1, X_2, ..., X_N)$$

therefore

$$F_\infty(\widetilde{X_1}, \widetilde{X_2}, \ldots, \widetilde{X_N}) = F_\infty(\widehat{X_1}, \widehat{X_2}, \ldots, \widehat{X_N})$$

□

It should be noted that the proof does not mention about the cheirality constraints. This is to keep the proof simple. The proof with cheirality constraints will follow essentially the same arguments.

In the resection step, the structure is fixed and optimization is done over the camera parameters and we have the following corollary:

Corollary 1. *The problem of minimizing the L_∞ norm of reprojection error over the joint camera projection matrices $(P^1, P^2, \ldots, P^M)$ can be solved by minimizing, independently, the L_∞ norm of reprojection error corresponding to each projection matrix P^j, i.e. suppose $(\widehat{P^1}, \widehat{P^2}, \ldots, \widehat{P^M})$ solves the joint camera matrices problem and $(\widetilde{P^1}, \widetilde{P^2}, \ldots, \widetilde{P^M})$ solves the independent problem for P^1, P^2,…,P^M respectively, then*

$$F_\infty(\widehat{P^1}, \widehat{P^2}, \ldots, \widehat{P^M}) = F_\infty(\widetilde{P^1}, \widetilde{P^2}, \ldots, \widetilde{P^M}) \qquad (9)$$

The proof is similar to that of Theorem 1 with the structure parameters replaced by the camera parameters. Hence joint minimization over all the structure/camera parameters can be solved by solving for individual structure/camera parameters.

2.2 Cheirality and quasi-affine initialization

The usual way to initialize a projective bundle adjustment algorithm is a projective reconstruction obtained from the given images. Any of the methods mentioned in [2] can be used for projective reconstruction. However doing this for the proposed algorithm will increase its computational complexity. To understand and get around this problem, we make a small digression and precisely state the definition of cheirality.

Let $\mathbf{X} = (X, Y, Z, T)$ be a homogeneous representation of a point and $P = (a^T; b^T; c^T)$ be the projection matrix of a camera, then the imaged point x is given by $P\mathbf{X} = \omega\hat{x}$, where $\hat{x}$ denotes the homogeneous representation of x in which the last coordinate is 1. The depth of the point $\mathbf{X}$ with respect to the camera is given by:

$$\text{depth}(\mathbf{X}; P) = \frac{\text{sign}(\text{det}M)\omega}{T||m_3||} \qquad (10)$$

where M is the left hand 3×3 block of P and m_3 is the third row of M [2].

A point $\mathbf{X}$ is said to be in front of the camera if and only if depth($\mathbf{X}; P$) > 0.

Definition 2. The quantity sign(depth($\mathbf{X}; P$)) is known as the cheirality of the point $\mathbf{X}$ with respect to the camera [2].

If det$M > 0$ and $T > 0$, then $c^T\mathbf{X} > 0$ implies that the point is in front of the camera(since $\omega = c^T\mathbf{X}$). In section 1.1, we have seen that the triangulation and resection problems were solved while satisfying this cheirality constraint. Cheirality is invariant under Euclidean and affine transformations (in fact, also under quasi-affine transformations) [2] and so seeking a solution with the cheirality constraint is justified when solving for the Euclidean/affine triangulation and resection problems.

However, cheirality is not invariant under projective transformations. Hence when solving the projective triangulation problem, we can't just restrict our search for X in the convex region of $\{X : c^{jT}X > 0, \forall j\}$. If there are M cameras, then the M principal planes divide the projective space P^3 into $(M^3 + 3M^2 + 8M)/6$ regions [4]. The L_∞ cost, with respect to X, has to be minimized over each of these regions and the minimum among them is the desired solution of the projective triangulation problem[4]. To avoid this minimization over all the regions during triangulation, a projective reconstruction should be converted to an Euclidean/affine/quasi-affine reconstruction. Once this is done, we can minimize the cost over the convex region $\{X : c^{jT}X > 0, \forall j\}$. A similar argument holds for the projective resection problem.

A good choice would be to perform a conversion from a projective reconstruction to a quasi-affine reconstruction. A quasi-affine reconstruction lies in between projective and

Algorithm *Bundle Adjustment using L_∞ norm minimization*

Input: Set of images
Output: Projective reconstruction
1. Do initial projective reconstruction from set of images.
2. Convert to quasi-affine reconstruction.
3. **while** Reprojection error $> \epsilon$
4. **do** Get camera parameters for each camera by L_∞ resection.
5. Get 3D structure parameters for each point by L_∞ triangulation.

Figure 1. L_∞ **BA Algorithm**

affine reconstruction. The only information required for this conversion is the fact that if a point is imaged by a camera, then it must be in front of the camera. The transformation that takes a projective reconstruction to a quasi-affine reconstruction can be found by solving a linear programming problem [2].

To summarize, we first obtain an initial projective reconstruction from the images and then convert this to a quasi-affine reconstruction by solving a linear programming problem. This reconstruction is then used as an initialization for our algorithm. After this initialization, we can use the triangulation/resection method of section 1.1.

The summary of the algorithm is given in Figure 1.

2.3 Convergence of the algorithm

Here we give a formal proof that at each step of resection and intersection, the L_∞ reprojection error either decreases or remains constant.

Theorem 2. *Consider the following sequence of steps in the algorithm : Initial Structure → Resection (R) → Intersection (I). Then $F_\infty(I) \leq F_\infty(R)$, where $F_\infty(I)$ and $F_\infty(R)$ are L_∞ cost after intersection and resection respectively.*

Proof. Let the initial structure points be $(\hat{X}_i, i = 1, 2, \ldots, N)$. In the resection step, $P^j, j = 1, 2, \ldots, M$ are estimated based on $\hat{X}_i$. Let the estimated P^j be $\hat{P}^j = [\hat{a^{j}}^T; \hat{b^{j}}^T; \hat{c^{j}}^T]$. Then $\hat{c^{j}}^T \hat{X}_i > 0$ for all i and j (cheirality constraint imposed by the algorithm).

Now consider the intersection step, where the structure is estimated from $\hat{P}^j$s. Let C be the cheirality constraint set $\{X : \hat{c^{j}}^T X > 0\}$. By decoupling (see section 2.1), the estimated structure $\tilde{X}_i$ is given by:

$$\tilde{X}_i = \arg\min_{X_i \in C} f_{\infty,i}(X_i) \tag{11}$$

i.e.

$$f_{\infty,i}(\tilde{X}_i) \leq f_{\infty,i}(X_i) \qquad \forall X_i \in C$$

hence,

$$f_{\infty,i}(\tilde{X}_i) \leq f_{\infty,i}(\hat{X}_i)$$

This is true for all i, therefore

$$\max_i f_{\infty,i}(\tilde{X}_i) \leq \max_i f_{\infty,i}(\hat{X}_i) \tag{12}$$

$$F_\infty(\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_N) \leq F_\infty(\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_N) \tag{13}$$

$$F_\infty(I) \leq F_\infty(R) \tag{14}$$

□

The intuition of the proof is that when we look at the two steps, Initial Structure → Resection and Resection → Intersection, resection is the common step. So P^js will be same in the two steps, with the values of X_is differing. In the intersection step, we are minimizing over X_is under the cheirality constraints and since the original structure also satisfies the same constraints, the structure after intersection will contribute a smaller cost than the original structure.

Corollary 2. *Consider the following step in the algorithm: Initial camera parameters → Intersection (I) → Resection (R). Then $F_\infty(R) \leq F_\infty(I)$, where $F_\infty(R)$ and $F_\infty(I)$ are the L_∞ cost after resection and intersection respectively.*

The proof is the same with structure replaced by camera parameters and vice versa. Theorem 2 and corollary 2 together prove that the L_∞ reprojection error decreases or remains constant with every step of resection and intersection. Since the reprojection error is bounded from below by zero, by monotone convergence theorem, the algorithm converges in the L_∞ reprojection error. This proof only guarantees the convergence of the algorithm to a local minimum of the L_∞ cost function. Convergence to global minimum is dependent on good initialization. We should note that the L_2 based bundle adjustment also has similar guarantee.

3 Computational complexity and memory requirement

This section first describes the computational complexity and memory requirement of the proposed algorithm and then compares it with that of L_2 based bundle adjustment and L_2 based resection-intersection algorithms.

As discussed in section 2.1, at any time we are either solving the triangulation problem for one structure point or the resection problem for one camera. We first analyze the computational complexity and memory requirements for solving one triangulation problem. The triangulation problem is solved by a bisection algorithm [3],[5]. At each step of the bisection, we solve a convex feasibility problem, given in (5) of section 1.1. If the point that we are triangulating is visible in m cameras then we have to solve m second order cone feasibility problem. This problem has a computational complexity of $O(m^{1.5})$ and a memory requirement of $O(m)$ [7]. By analogy, the resection problem for one camera has an empirical computational complexity $O(n^{1.5})$ and memory requirement $O(n)$ where n points are visible in that camera. Now consider one iteration of our algorithm. For the case where there are m cameras and n points and all the points are visible in all the cameras, per iteration empirical computational complexity is $O(mn(\sqrt{m}+\sqrt{n}))$ and the memory requirement is $O(max(m,n))$. Furthermore a parallel implementation of the algorithm is possible because during each resection/intersection step all the cameras/points can be estimated at the same time. This is because of the decoupling discussed in section 2.1. Such an implementation will result in a reduction of computational complexity.

The L_2 norm bundle adjustment algorithm (L_2 BA) is based on the Levenberg-Marquardt (LM) method. The central step involves solving an equation with all the camera and structure parameters as unknowns. Hence its computational complexity is $O((m+n)^3)$ per iteration. For memory requirement, we can consider the Jacobian which is $O(mn(m+n))$. For structure from motion problems, there exists a sparse LM method which uses the fact that the cameras are related to each other only through structure and vice-versa [2]. For sparse LM, computation complexity is $O(m^3+mn)$ per iteration. The memory requirement is $O(mn)$ [2].

The L_2 based resection-intersection algorithms [1],[9] have computational complexity of $O(mn)$ per iteration and same memory requirement as our algorithm, i.e, $O(max(m,n))$ [9]. But since these algorithms minimize approximate algebraic errors, they are not so reliable as we found in our experiments 4.1.

Uptill now we have discussed only about the computational complexity per iteration. But there is also the question of number of iterations required for convergence. Theoretically it is a difficult question to answer, but we would like to point out one fact which favors our algorithm. In each step of resection and intersection our algorithm finds the global minimum of the respective cost functions. This is a big advantage over L_2 BA algorithm which doesn't do an exact line search over it's search direction. Deciding how much to move along a search direction is an important issue for fast convergence and it goes by the name of step control [14]. The L_2 based resection-intersection algorithms [1], [9] do find the global minimum in each step but they do so of approximate, algebraic cost functions.

4 Experiments

We have done experimental evaluation of the proposed algorithm (L_∞ BA) for convergence, computational scalability and robustness to noise. Comparisons with L_2 bundle adjustment based on LM algorithm (L_2 BA) and L_2 resection-intersection algorithms are also given. For L_2 BA we have used a publicly available implementation of projective bundle adjustment [8] based on the sparse LM method. The L_2 based resection-intersection algorithms that we have compared with are the Weighted Iterative Eigen algorithm (WIE) proposed by Chen et al [1] and a variation of the same algorithm where we avoid the reweighting step, henceforth called the IE algorithm. In section 4.1, while studying the convergence of the four algorithms we found that the performance of WIE and IE are unreliable and hence in the rest of the sections we have compared L_∞ BA with only L_2 BA.

For initial reconstruction, we have used the projective factorization method of Triggs et al [13] with proper handling of missing data. This reconstruction was then converted to a quasi-affine reconstruction. Any other projective reconstruction followed by a conversion to a quasi-affine reconstruction will also work fine. When comparing different algorithms, all of them have been provided with the same initial reconstruction.

4.1 Convergence

Here we study the convergence with iteration of L_∞ BA, L_2 BA, WIE and IE algorithms. Experiments are done on one synthetic data set and three real data sets. The synthetic data set consists of 100 points distributed uniformly within a unit sphere and 50 cameras in a circle around the sphere looking straight at the sphere. Gaussian noise of standard deviation 1 pixel is added to the feature points. In all the experiments, reprojection error for this data set is the mean reprojection error over ten trials. The real data sets used are the corridor and dinosaur data set[2] and the hotel data set[3]. We have used a subset of 464 structure points from the dinosaur data set and a subset of 60 views from the hotel data set. For the corridor and the dinosaur data sets, feature points are already available and we have used them as they are. For the hotel data set we have used the KLT tracker to track feature points and then used Torr's Matlab SFM Toolbox [12] to remove the outliers.

We compare the convergence of the algorithms in the L_∞ norm of reprojection error and the Root Mean Squares reprojection error (RMS) which is a measure of the L_2 norm. Figure 2 shows that the L_∞ error decreases monotonically for L_∞ BA, as claimed in section 2.3, but not so

[2]http://www.robots.ox.ac.uk/∼vgg/data/data-mview.html
[3]http://vasc.ri.cmu.edu/idb/

for the other algorithms. Figure 3 shows RMS error decreases (almost) monotonically for L_∞ BA and L_2 BA but not so for WIE and IE. From Figure 3, we can further conclude the following. All the algorithms converge well for the sphere data set. For the corridor data set, WIE converges at a higher value than others. For the hotel data set, both WIE and IE first diverge and then later converge. For the dinosaur data set, IE fails to converge. To further study the nature of WIE and IE, we added Gaussian noise of standard deviation 1 pixel to the real data sets and found that the algorithms fail to converge at many trials. However, each time these algorithms have converged in the algebraic cost that they minimize. This study tells us that algebraic cost based algorithms may not be very reliable. For all of the above data sets, our algorithm converges within ten iterations with similar RMS reprojection error as L_2 BA. Figure 4 shows the final 3-D reconstruction by our algorithm for the datasets, sphere and corridor.

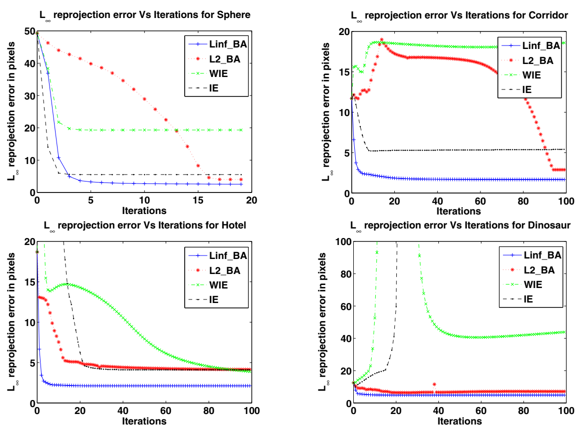

Figure 2. L_∞ reprojection error versus iteration for the four algorithms on the data sets: sphere, corridor, hotel and dinosaur. L_∞ error decreases monotonically for L_∞ BA but not so for the other algorithms.

4.2 Computational scalability

We did experiment on the synthetic sphere data set to compare the total convergence time for L_2 BA and L_∞ BA as the number of cameras is varied with the number of points fixed at 500, Figure 5. To ensure fairness, both the algorithms were implemented in Matlab with the computationally intensive routines as mex files. L_2 BA converges at about 10 iterations and L_∞ BA at about 2 iterations. Figure 5 clearly shows that our algorithm has the advantage in terms of time from 250 cameras onwards. For a video with 30 frames per second this is approximately 8 sec of data.

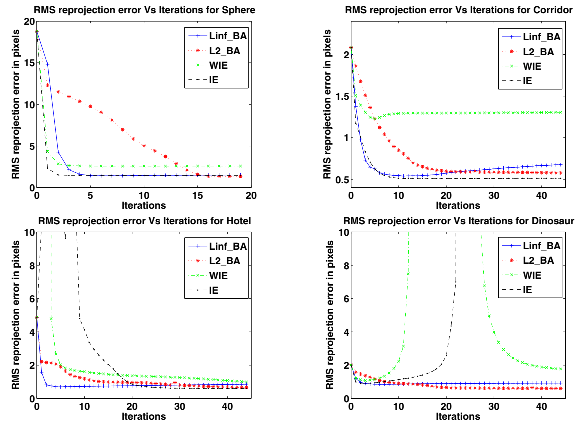

Figure 3. RMS reprojection error versus iteration : RMS error decreases monotonically for L_∞ BA and L_2 BA but not so for WIE and IE. IE fails to converge for the dinosaur data set.

Note that we have to estimate the camera parameters corresponding to each frame of the video. Thus our algorithm is suitable for solving reconstruction problems for video data where the number of frames can be large.

Recently, there has been some work on faster computations of L_∞ triangulation and resection problems [10] and incorporating this will reduce the convergence time of our algorithm. Further reduction in convergence time is possible by a parallel implementation, which we have not done here.

4.3 Behavior with noise

Gaussian noise of different standard deviations are added to the feature points. Figure 6 shows the RMS reprojection error in pixels with noise for the synthetic data set, sphere. Generally the L_∞ norm has the reputation of being very sensitive to noise, but here we see a graceful degradation with noise. Further to handle noise with strong directional dependence, we can incorporate the directional uncertainty model of Ke et. al. [6] into the resection and triangulation steps of our algorithm, though we have not done it here. We have not considered outliers here, as bundle adjustment is considered to be the last step in the reconstruction process and outlier detection is generally done in the earlier stages of the reconstruction. In fact as mentioned earlier in section 4.1, we have done outlier removal for the hotel data set before the initial reconstruction step.

5 Conclusion

We have proposed a scalable projective bundle adjustment algorithm using the L_∞ norm. It is a resection-

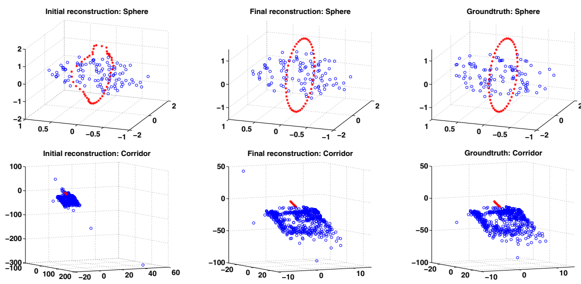

Figure 4. 3-D reconstruction result for the datasets, Sphere and Corridor. The Red '*' represents the camera center and Blue 'o' represents the structure point. The first column shows the initialization, second column shows the final reconstruction and the third column shows the groundtruth.

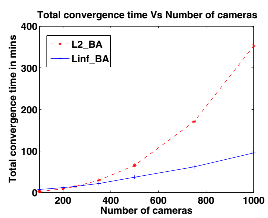

Figure 5. Total convergence time of L_2 BA and L_∞ BA as the number of cameras is varied with number of points fixed at 500.

intersection algorithm which converts the large scale optimization problem to many small scaled ones. There are scalable resection-intersection algorithms based on the L_2 norm but they are not so reliable as we have found out in our experiments. Our algorithm gives similar performance as L_2 BA in terms of minimizing the reprojection error. It has a good convergence rate and degrades gracefully with noise.

In conclusion we can say that our algorithm will outperform L_2 BA when solving for problems with large number of cameras or frames like in a video data set. It is possible to make the present algorithm faster using a parallel implementation and by a more efficient implementation of L_∞ resection and triangulation.

Acknowledgements

The first author wishes to thank Sameer Shirdhonkar and Ashok Veeraraghavan for discussions and encouragement.

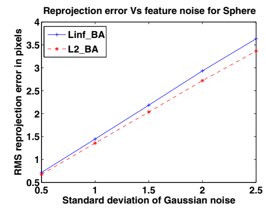

Figure 6. Behavior of L_∞ BA and L_2 BA with image feature noise for the sphere data set.

References

[1] Q.Chen and G.Medioni, *Efficient iterative solution to M-view projective reconstruction problem*, IEEE Conf. on CVPR, 1999.

[2] R.Hartley and A.Zisserman, *Multiple view geometry in computer vision*, Cambridge University press, 2nd edition, 2004.

[3] F.Kahl, *Multiple view geometry and the L_∞-norm*, IEEE ICCV, 2005.

[4] F.Kahl and R.Hartley, *Multiple view geometry under the L_∞ -norm*, IEEE TPAMI, 2007.

[5] Q.Ke and T.Kanade, *Quasiconvex optimization for robust geometric reconstruction*, IEEE ICCV, 2005.

[6] Q.Ke and T.Kanade, *Uncertainty Models in Quasiconvex Optimization for Geometric Reconstruction*, IEEE CVPR, 2006.

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

[8] M.I.A. Lourakis and A.A. Argyros, *The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm*, ICS/FORTH Technical Report No. 340, 2004.

[9] S.Mahamud, M.Herbert, Y.Omori and J.Ponce, *Provably-Convergent Iterative Methods for Projective Structure from Motion*, IEEE Conf. on CVPR, 2001.

[10] C.Olsson, A.P.Eriksson and F.Kahl, *Efficient Optimization for L_∞-problems using Pseudoconvexity*, Proc. IEEE ICCV, 2007.

[11] J.Sturm, *Using SeDuMi 1.02, A* MATLAB *toolbox for optimization over symmetric cones.*, Optimization methods and software, 11-12:625-653, 1999.

[12] P.H.S.Torr, *A Structure and Motion Toolkit in Matlab "Interactive advantures in S and M"*, Technical report, MSR-TR-2002-56, June 2002.

[13] B.Triggs, *Factorization methods for projective structure and motion*, Proc. IEEE Conf. on CVPR, 1996.

[14] B.Triggs, P.McLauchlan, R.Hartley and A.Fitzgibbon, *Bundle adjustment - a modern synthesis*, In Vision Algorithms: Theory and Practice, Springer-Verlag, 2000.