L_∞ minimization in geometric reconstruction problems

Richard Hartley[1,2] and Frederik Schaffalitzky[2,3]
[1]National ICT Australia
[2]The Australian National University
[3]Oxford University

Abstract

We investigate the use of the L_∞ cost function in geometric vision problems. This cost function measures the maximum of a set of model-fitting errors, rather than the sum-of-squares, or L_2 cost function that is commonly used (in least-squares fitting). We investigate its use in two problems; multiview triangulation and motion recovery from omnidirectional cameras, though the results may also apply to other related problems. It is shown that for these problems the L_∞ cost function is significantly simpler than the L_2 cost. In particular L_∞ minimization involves finding the minimum of a cost function with a single local (and hence global) minimum on a convex parameter domain. The problem may be recast as a constrained minimization problem and solved using commonly available software. The optimal solution was reliably achieved on problems of small dimension.

1 The triangulation problem

Let $\mathtt{P}_i; i = 1, \ldots, n$ be a sequence of n known cameras, and $\mathbf{x}_i$ be the image of some unknown point $\mathbf{X}$ in 3-space, both expressed in homogeneous coordinates. Thus, we write $\mathbf{x}_i = \mathtt{P}_i\mathbf{X}$. The problem of computing the point $\mathbf{X}$ given the camera matrices $\mathtt{P}_i$ and the image points $\mathbf{x}_i$ is known as the *triangulation* problem.

In the absence of noise, the triangulation problem is trivial, involving only finding the intersection point of rays in space. When noise is present, however, the rays corresponding to back-projections of the image points do not intersect in a common point, and obtaining the best estimate of the point $\mathbf{X}$ is not always easy. The correct procedure is to find the point $\mathbf{X}$ that projects most nearly to the image points $\mathbf{x}_i$. In this context, the words “most nearly” are usually interpreted in a least-squares sense. Thus, we are required to find the point $\mathbf{X}$ that minimizes the cost function

$$\sum_{i=1}^{n} d(\mathbf{x}_i, \mathtt{P}_i\mathbf{X})^2 \tag{1}$$

where $d(\cdot, \cdot)$ represents the geometric distance between two points in the image.

This problem has been solved for the case of two views in [2]. There, a method is given involving the solution of a sixth-degree polynomial. Unfortunately, this method is not generalizable to more than two views.

Algebraic method A simple algebraic method exists for solving this problem in the n-view case. Each equation $\mathbf{x}_i = \mathtt{P}_i\mathbf{X}$ holds only up to an unknown scale factor, and can be written more precisely as $k_i\mathbf{x}_i = \mathtt{P}_i\mathbf{X}$. This equation is linear in the unknown quantities k_i and $\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 1)^\top$. One may write a complete set of equations as

$$\begin{bmatrix} \mathtt{P}_1 & \mathbf{x}_1 & & & \\ \mathtt{P}_2 & & \mathbf{x}_1 & & \\ \vdots & & & \ddots & \\ \mathtt{P}_n & & & & \mathbf{x}_n \end{bmatrix} \begin{pmatrix} \mathbf{X} \\ -k_1 \\ -k_2 \\ \vdots \\ -k_n \end{pmatrix} = \mathbf{0} \tag{2}$$

which can be solved up to scale by a linear algebraic method. Though this method of triangulation may seem attractive, the cost function that it is minimizing has no particular meaning, and the method is not reliable.

Mid-point method In the case of calibrated cameras, and hence Euclidean triangulation, another alternative is to find the closest point in 3-space to the rays back-projected from the image points. In the case of two views, this is the mid-point of the common perpendicular to the two rays. This method fails badly in the case where the rays are almost parallel, corresponding to a point near infinity, since in this case, the computed point will be close to the point half-way between the two camera centres.

As part of the task of reconstructing a scene from a long image sequence [5] we needed to triangulate about 800,000 points many from several frames. We experienced significant problems with triangulation, around 0.1% – 0.5% of points or more giving completely wrong values for the point position, with resultant very large projection errors (1̃00 pixels). This was enough to cause problems with reconstruction.

2 Multiple minima

It was seen in [2] that even in the 2-view case there can be multiple local minima of the geometric cost function. This is because the solution involves the solution of a degree-6 polynomial, which may have as many as three maxima and three minima.

We illustrate this for the n-view case and inquire into its causes. We wish to find the way the value of the cost function (1) varies as a function of $\mathbf{X}$. Rather than plot this function for all $\mathbf{X}$, we plot a 1-dimensional cross-section, parametrized by a variable t. Thus, we consider a variable point $\mathbf{X}(t) = \mathbf{X}_0 + t\delta_{\mathbf{X}}$ moving along a straight line. The cost function at "time" t is given by

$$F(t) = \sum_{i=1}^{n} d(\mathbf{x}_i, \mathtt{P}_i\mathbf{X}(t))^2 = \sum_{i=1}^{n} f_i(t)^2$$

where $f_i(t)$ is the error associated with the projection of the point in the i-th image. For values of t such that $\mathbf{X}(t)$ lies in front of the camera, the form of the function $f_i(t)$ may be visualized as follows. As the 3D point moves along the trajectory $\mathbf{X}(t)$, its image point moves monotonically in a constant direction (thought not with constant velocity) along a straight line in the image. As t varies, it will approach the point $\mathbf{x}_i$, reach a unique point of closest approach, and then recede again. At some point, the trajectory will cross the focal plane of the camera, and at this point, the imaged point recedes to infinity. A straight-forward computation shows that the squared-distance function $f_i(t)^2$ is of the form

$$f_i(t)^2 = \frac{a + bt + ct^2}{(d + et)^2} \quad . \tag{3}$$

This distance function has a single minimum, when $t = (bd - 2ae)/(be - 2cd)$, and becomes infinite when $t = -d/e$.

Details of this computation follow. Let $\mathbf{X}(t) = \mathbf{X}_0 + t\Delta$. Then the projected point is

$$\begin{pmatrix} \mathbf{p}^{1\top}(\mathbf{X}_0 + t\Delta)/\mathbf{p}^{3\top}(\mathbf{X}_0 + t\Delta) \\ \mathbf{p}^{2\top}(\mathbf{X}_0 + t\Delta)/\mathbf{p}^{3\top}(\mathbf{X}_0 + t\Delta) \end{pmatrix}$$

where $\mathbf{p}^{j\top}$ is the j-th row of the camera matrix $\mathtt{P}$. This expression is of the form

$$\begin{pmatrix} (r + st)/(d + et) \\ (u + vt)/(d + et) \end{pmatrix}$$

and the squared distance to a reference point (x, y) is then

$$\left(\frac{r + st}{d + et} - x\right)^2 + \left(\frac{u + vt}{d + et} - y\right)^2 \quad .$$

Gathering terms over a common denominator results in an expression of the form (3), as claimed.

The function is plotted in Fig 1.

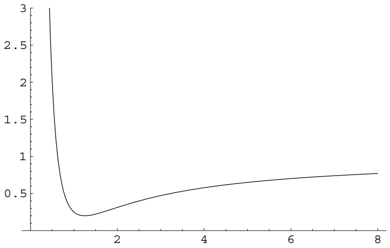

Figure 1: *Plot of a one-dimensional slice of the projection cost function (3). The function plotted is* $((t-1)^2 + 1/4)/t^2$. *Note that the function has a single minimum in this region of the graph (representing the region in front of the camera), but it is not convex.*

Cheirality By cheirality, we mean the consideration of which points are in front of the camera. Obviously, since the desired point $\mathbf{X}$ is visible in the image, it must lie in front of the camera. Consideration of cheirality requires that camera matrices are known in an affine (or at least quasi-affine) coordinate frame ([3]). For most of this paper, we consider affine or Euclidean reconstruction, briefly considering projective triangulation in a later section. Given several cameras, the region of space that lies in front of all the cameras is clearly convex, since it is defined as the intersection of a set of half-spaces bounded by the principal planes of the cameras.

From the geometric intuition given above, each function $f_i(t)$ must have a single minimum for points in front of the camera. Thus, we make the following observation regarding this cost function.

- For values of t for which $\mathbf{X}(t)$ lies in front of the camera, $f_i(t)$ has a single minimum; however it is *not* a convex function.

The total cost function for all images is $\sum_i f_i(t)^2$. If each f_i were convex, then we could conclude that their sum would be convex, and hence have a single minimum; but such is not the case. In fact, the total cost function $\sum_i f_i(t)^2$ may have several minima.

Thus, we have shown that the sum-of-squares cost function for triangulation potentially has multiple minima along any one-dimensional cross-section. This does not prove that there is more than one minimum in the three-dimensional convex domain of the variable point $\mathbf{X}$, but it gives strong evidence for multiple minima. Besides, the existence of multiple minima was shown in [2] for the two-view case. In any case, the existence of multiple minima along straight-lines in parameter space indicates the complexity of the cost surface, and the difficulty in finding a global minimum by iterative optimization methods, many of which proceed by finding minima along suitably chosen search directions.

3 L_∞ cost function

The total cost function for triangulation may be viewed as the L_2-norm of the vector $\mathbf{f}(\mathbf{X}) = (f_1(\mathbf{X}), f_2(\mathbf{X}), \ldots, f_n(\mathbf{X}))$. As was shown in the previous section, this cost function will not have a single minimum as $\mathbf{X}$ varies, because each $f_i(\mathbf{X}(t))$ is not a convex function of t. The key observation of this paper is that the L_∞ norm of this vector *will* have a single minimum. This results from the following result:

Theorem 3.1. *For each $i = 1, \ldots, n$, let $f_i(t)$ be a continuous function on a closed interval I_i, having a single minimum on that interval. We assume that the intervals I_i have non-empty intersection $\cap_i I_i$. Define the function $f_{\max}(t)$ on $\cap_i I_i$ as the maximum of the functions, namely*

$$f_{\max}(t) = \max_i f_i(t) \ .$$

Then $f_{\max}(t)$ has a single minimum on the interval $\cap_i I_i$.

Proof. We use the result that a continuous function must have a minimum on a closed interval. Suppose, contrary to the conclusion of the theorem, that $f_{\max}(t)$ has two local minima, at values $t = a$ and $t = b$, and suppose that $f_{\max}(a) \geq f_{\max}(b)$ and (without loss of generality) that $a < b$. Since a is a local minimum of $f_{\max}(t)$, there exists a value c between a and b such that $f_{\max}(c) \geq f_{\max}(a)$. Let f_i be a function such that $f_i(c) = f_{\max}(c)$. Then we see that $f_i(c) \geq f_{\max}(a) \geq f_i(a)$, and $f_i(c) \geq f_{\max}(a) \geq f_{\max}(b) \geq f_i(b)$. The function $f_i(t)$ must therefore have a minimum on each of the intervals $[a, c)$ and $(c, b]$. This contradicts the assumption that $f_i(t)$ has a single minimum on the interval $[a, b]$.

This proposition implies that the function $\|\mathbf{f}(\mathbf{X})\|_\infty$ has a single minimum along any straight line in the domain of $\mathbf{X}$. Consequently, it must have a single minimum over the whole domain. For, if two local minima exist at two separate points, then they must also be local minima along the line joining the two points. Since this can not occur, only one minimum can exist. This gives the following result.

Theorem 3.2. *If $f_i(\mathbf{X})$ is the reprojection error $d(x_i, \mathtt{P}_i\mathbf{X})$, then the L_∞ norm $\|\mathbf{f}(\mathbf{X})\|_\infty$ has a single minimum on the region of space lying in front of all the cameras. This region is convex.*

Since $\|\mathbf{f}(\mathbf{X})\|$ is a function with a single minimum defined on a convex domain, it is a relatively easy optimization problem, and we do not risk falling into a local minimum. An unsophisticated method of finding the minimum is to start at a random point in the allowable domain of $\mathbf{X}$, then search in a sequence of random directions until no further improvement is possible. This method works reliably (though somewhat slowly) for this simple problem of triangulation, since the dimension of the parameter space is small – $\mathbf{X}$ lies in a 3-dimensional convex region.

The L_∞ norm of a vector is the maximum absolute value of its entries, $f_i(\mathbf{X})$. These entries are image-space distances, and evaluating them requires taking a square root. This is unnecessary, since we may equally well find the maximum of the values of $f_i(\mathbf{X})^2$. The value of $f_i(\mathbf{X})^2$ is given by (3), and does not require taking a square root.

Initialization. Finding an initial estimate for $\mathbf{X}$ is a simple linear programming problem. The condition that the point $\mathbf{X}$ lies in front of the camera $\mathtt{P}_i$ is expressed by the condition that $\mathtt{P}_i\mathbf{X} = (x_i, y_i, w_i)^\top$ with $w_i > 0$, provided that $\mathtt{P}_i$ is normalized so that its left-hand 3×3 block has positive determinant ([1, 3]). This is a simple linear inequality. The cheirality conditions for all the views defines a linear-programming problem; we may formulate some suitable goal function to optimize subject to these inequalities; the precise goal function is not too important, as long as we find an initial point in the region determined by the inequalities.

Line search. Finding the minimum of a function along a single line is easy if that function has a single minimum in a known interval. The minimum may be reliably located using Fibonacci search ([4]). A more efficient way of finding the minimum of the function $\max_i f_i(t)$ is mentioned briefly in section 4.1.

4 Multiple-view reconstruction

The method of L_∞ minimization can be used to good effect in solving somewhat more complex problems. We consider the problem of multiple-view structure and motion. For simplicity of exposition, we consider the case of calibrated omnidirectional cameras, though the method applies equally well to standard projective cameras.

In our model of a calibrated omnidirectional camera, each image point is represented by a unit vector, representing the direction from the camera centre to the imaged 3D point. This vector is represented in the Euclidean coordinate frame of the camera. In applying the method of L_∞ minimization to this problem, we assume that the rotations of each of the cameras has been determined in advance. This is not unrealistic. We have implemented ([5]) a method for structure and motion computation for very long sequences (up to 10,000 frames) in which the rotation matrix for each camera is determined in a first step, leaving the positions of the cameras and the points to be determined in a second step.

The reconstruction problem may now be formalized as follows. Consider a sequence of omnidirectional cameras located at positions $\mathbf{C}_i; i = 1, \ldots, n$, and several 3D points $\mathbf{X}_j; j = 1, \ldots, m$. We are given some of the unit vectors $\mathbf{x}_{ij} = (\mathbf{X}_j - \mathbf{C}_i)/\|\mathbf{X}_j - \mathbf{C}_i\|$. These represent the "image vectors" for each pair (i, j) where point j is visible in camera i. The available image vectors $\mathbf{x}_{ij}$ are not known exactly, but rather are subject to some noise disturbance. Given the noisy image vectors $\mathbf{x}_{ij}$, the task is to determine the location of the cameras $\mathbf{C}_i$ and points $\mathbf{X}_j$.

The problem is solvable only up to translation and scale. This ambiguity may be resolved by requiring that the first camera $\mathbf{C}_1$ is located at the origin, and that some point $\mathbf{X}_j$ visible in the first camera is at unit distance along the direction $\mathbf{x}_{1j}$. That is

$$(\mathbf{X}_j - \mathbf{C}_1) \cdot \mathbf{x}_{1j} = \mathbf{X}_j \cdot \mathbf{x}_{1j} = 1 \ . \tag{4}$$

Cheirality condition. A cheirality condition applies to this problem, namely that the image vector $\mathbf{x}_{ij}$ must point in the direction from the camera centre to the point. This gives an inequality

$$(\mathbf{X}_j - \mathbf{C}_i) \cdot \mathbf{x}_{ij} > 0 \tag{5}$$

for all (i, j) for which $\mathbf{x}_{ij}$ is given. This condition may be used to determine an initial configuration for the problem (more about this later).

We show now that the region in parameter space satisfying these inequalities is a convex region. The parameters for this problem are the coordinates of all the camera centres and 3D points. Thus, a configuration is represented by a parameter vector of dimension $3(m + n)$. The parameter vectors that satisfy the conditions $\mathbf{C}_1 = (0, 0, 0)^\top$, the scale-constraint (4) and the cheirality constraints (5) form a convex set. This follows from the general result that a region in Euclidean space defined by linear equalities and inequalities is convex.

A cost function. Given hypothesized positions $\mathbf{C}_i$ and $\mathbf{X}_j$ for the cameras and points, the error associated with a given measurement $\mathbf{x}_{ij}$ may reasonably be defined as the angle θ_{ij} between the two vectors $\mathbf{x}_{ij}$ and $(\mathbf{X}_j - \mathbf{C}_i)$. To compute this angle would require an inverse trigonometric function, however, which we wish to avoid. Since we are interested in the maximum of all such errors, we may instead consider any monotonically increasing function of this angle. We choose $\tan^2(\theta_{ij})$, which indeed increases monotonically, as long as $\theta_{ij} < \pi/2$, which will always be the case when the cheirality conditions are satisfied. Writing $\mathbf{v}_{ij} = \mathbf{X}_j - \mathbf{C}_i$, the cost of the measurement $\mathbf{x}_{ij}$ is then equal to

$$\begin{aligned} \frac{\|\mathbf{v}_{ij} \times \mathbf{x}_{ij}\|^2}{(\mathbf{v}_{ij} \cdot \mathbf{x}_{ij})^2} &= \frac{\|\mathbf{v}_{ij}\|^2 \sin^2(\theta_{ij})}{\|\mathbf{v}_{ij}\|^2 \cos^2(\theta_{ij})} \qquad (6) \\ &= \tan^2(\theta_{ij}) \ . \end{aligned}$$

Since the values of the measured image vectors $\mathbf{x}_{ij}$ are known, both the numerator and denominator of this cost function are quadratic functions of the coordinates of $\mathbf{X}_j$ and $\mathbf{C}_i$.

A single minimum. We now show that the cost function

$$\max_{i,j} \tan^2(\theta_{ij})$$

has a single minimum in the whole convex parameter space defined by the cheirality conditions. To this end, we show that it has a single minimum along any one-dimensional cross-section (line) in parameter space. Thus, let $\mathbf{X}_j(0)$ and $\mathbf{C}_i(0)$ be initial parameter values and let $\delta_{\mathbf{X}_j}$ and $\delta_{\mathbf{C}_i}$ define an incremental direction for the parameters. Define $\mathbf{X}_j(t) = \mathbf{X}_j(0) + t\delta_{\mathbf{X}_j}$ and $\mathbf{C}_i(t) = \mathbf{C}_i(0) + t\delta_{\mathbf{C}_i}$. The direction from camera i to point j at time t is then

$$\begin{aligned} \mathbf{v}_{ij}(t) &= \mathbf{X}_j(t) - \mathbf{C}_i(t) \\ &= (\mathbf{X}_j(0) - \mathbf{C}_i(0)) + t(\delta_{\mathbf{X}_j} - \delta_{\mathbf{C}_i}) \\ &= \mathbf{v}_{ij}(0) + t\delta_{ij} \end{aligned}$$

where δ_{ij} is defined by this relation. The angle $\theta_{ij}(t)$ is defined as the angle between $\mathbf{x}_{ij}$ and $\mathbf{v}_{ij}(t)$. Finally, we define

$$\begin{aligned} f_{ij}(t) &= \tan^2(\theta_{ij}(t)) \\ &= \frac{\|\mathbf{x}_{ij} \times (\mathbf{v}_{ij}(0) + t\delta_{ij})\|^2}{(\mathbf{x}_{ij} \cdot (\mathbf{v}_{ij}(0) + t\delta_{ij}))^2} \end{aligned}$$

Expanding this expression yields a rational quadratic cost function of the same form as (3), having a single minimum for values of t satisfying the cheirality conditions.

Thus, the minimization problem is very much the same as for the triangulation problem. Along a one-dimensional direction in space, the cost function $\max_{i,j} f_{ij}(t)$ has a single minimum, and hence the L_∞ cost function has a single minimum in the allowable convex region of parameter space.

There is an important difference, however, in that the minimum value is not achieved at a single point in parameter space. Instead, some of the parameters may vary locally at the minimum without changing the minimax function value. This is because at a minimum, some of the constraints $f_{ij} \leq f_{\mathrm{minimax}}$ are "active" whereas others are not. An active constraint is one in which equality holds. Parameters that are not involved in the active constraints may vary locally without changing the value of the L_∞ cost function. We thank Bill Triggs for clarifying this point.

4.1 Minimizing the cost function

In the triangulation problem, the parameter space is three-dimensional, since we are searching for the best point in a 3-dimensional space. This small-dimensionality makes the minimization problem relatively simple. Linear searches along random directions will ultimately find the minimum, though more intelligent methods of selecting search directions have been tried.

Finding the minimax of a set of functions along a line – one-dimensional search – may be achieved using subdivision search, such as Fibonaci search. We have also implemented an extremely efficient method of doing one-dimensional search, using a heap-based ordering of the curves. This method appears to be logarithmic (or sub-logarithmic) in complexity, with respect to the number of curves. In one experiment, the minimax of $1,000,000$ functions required only about $1,100,000$ curve intersections to be computed.

Whereas in the case of triangulation, the parameter space was only three-dimensional. for the multiple-view reconstruction problem, it has dimension $3(m+n)-4$, taking into account the equality constraints on the parameters. In this case, the strategy of selecting random search directions is not satisfactory, given the large dimension of the space. A more intelligent strategy for minimizing the cost function is needed.

We tried several strategies for minimizing the L_∞ cost function. As just mentioned, we implemented a very efficient strategy for determining the minimum value of the cost function along a line search. The remaining problem is to find a method for determining the best direction for each subsequent line search. In the case of least-squares minimization problems, various strategies for determining the line-search direction are currently used, such as conjugate gradient methods, gradient descent, Levenberg-Marquardt. All such strategies rely on determining the gradient, and possibly the Hessian of the cost function at the current point. These methods are not suitable for L_∞ minimization, simply because the gradient of the cost at a given point is not a good indicator for the behaviour of the cost function even locally. Recall that the cost function for a parameter set $\theta = \{\mathbf{X}_i\} \cup \{\mathbf{C}_j\}$ is of the form $\max_i f_i(\theta)$. The cost function is non-differentiable at many points in the space (in particular, when the two largest components of the cost function f_{i_1} and f_{i_2} switch their order).

We implemented various methods for determining new search directions. The general strategy is to choose a direction in which one can move the largest distance before meeting the local minimum. These methods, which will not be described here, were quite successful in the triangulation problem, but less successful in the full reconstruction problem, because of the large dimension of the parameter space.

Eventually, the strategy that worked best was to reformulate the minimization problem as a constrained minimization problem. The L_∞ optimization problem we are addressing is to find $\min_\theta \max_i f_i(\theta)$, where θ is a parameter set, and the $f_i(\theta)$ are certain well-behaved functions (with a single minimum on the convex parameter domain). This may be reformulated as

- *Find the parameter value θ that minimizes the value of a supplementary variable k subject to constraints $f_i(\theta) < k$ for all i.*

Thus, we add one extra parameter k and minimize (over all choices of θ and k) the cost function k subject to all constraints $f_i(\theta) < k$. This is a constrained minimization problem with a linear cost function and non-linear constraints. We used the program LOQO ([6]) to minimize this cost function, starting from an initial feasible solution found using linear programming (also implemented using LOQO). The results were very good, usually achieving results very close to the minimum. Nevertheless, on large problems it did not appear always to find the absolute minimum. The iteration seemed to become constricted by coming hard up against a large number of constraints. In addition, it was a little slow for large problems. To this point we have not quantified the performance completely.

5 Projective triangulation

The algorithm for triangulation that was given previously was valid in the case of affine reconstruction, in which the plane at infinity in the world is known. With this assumption, it was possible to determine when a point was in front of the cameras, and hence constrain the position of the point $\mathbf{X}$ to a convex region of 3D affine space. In a projective reconstruction, we do not know where the plane at infinity lies, and hence can not determine the region of space that lies in front of all the cameras.

In the projective problem, we are given several (suppose n) camera matrices $\mathtt{P}_i$ and the coordinates of matching point $\{\mathbf{x}_i\}$. The goal is to determine the point $\mathbf{X}$ that most nearly satisfies $\mathbf{x}_i = \mathtt{P}_i\mathbf{X}$ for all i. So far, this is the same problem as the affine (or Euclidean) triangulation problem. As with the affine problem, the point $\mathbf{X}$ should be chosen to be "in front of" the cameras. In the affine context, it is clear what this means. Let us examine the projective case, however.

Let π_i represent the principal plane of the i-th camera – that is the plane consisting of points that map to infinity in the image. The n principal planes divide projective space $\mathcal{P}^3$ into $M_n = \binom{n}{3} + \binom{n}{1} = (n^3 - 3n^2 + 8n)/6$ regions.[1] If the point $\mathbf{X}$ lies on any of the principal planes π_i, then it maps to a point at infinity in the corresponding image,

---

[1] This assumes that no four planes pass through a common point. Verification of this formula is left to the reader.

and hence has infinite reprojection error with respect to any finite image point. Thus, both the L_2 and L_∞ costs of a point $\mathbf{X}$ lying on one of the principal planes must be infinite. Consequently, the cost function must have at least one minimum in each of the regions. As we have shown, the L_∞ cost function must have just one minimum on a convex region, and hence will have just M_n local minima. The L_2 cost function may potentially have more than one minimum in each region.

To find the minimum of the L_∞ cost function, it is necessary to find the minimum in each of the M_n regions. However, once it is known that some point $\mathbf{X}$ visible in all the images lies in one of the regions delimited by the principal planes, then any other point must lie in the same region, in order to be in front of all the cameras. Thus, we are reduced to a search over a single convex region. We have not implemented this approach to projective triangulation from several images.

The large number of local minima for L_2 or L_∞ triangulation show the dangers of approaching triangulation as an unconstrained minimization problem; falling into a local minima is a real possibility. The approach suggested in this paper involves constrained minimization, which in the case of L_∞ minimization leads to a unique minimum.

6 Pros and cons of L_∞ fitting

An obvious criticism of L_∞ optimization is that it is highly dependent on the presence of outliers. In fact, in a sense we are fitting the outliers, and not the good data. This point of view has some merit. However, it is undeniable that outliers are also fatal to ordinary least-squares minimization. We suggest this algorithm be used on data from which outliers have been removed.

A simplistic scheme to outlier removal would be to carry out an L_∞ optimization, and if the residual error is too great, then remove the offending measurements and continue. To date we have not addressed methods of outlier removal in L_∞ optimization.

L_∞ optimization has other advantages beyond the obvious one of having simpler cost functions. Generally we have some idea of expected error bounds. For instance in measuring point correspondences in images, we may expect accuracy within about one pixel. Being able to find the L_∞ optimum gives a certain answer as to whether it is possible to fit the data to the model within the expected error bounds. In the case of L_2 minimization there is always the uncertainty that the data may be good, but we have fallen into a local minimum. If it is possible to fit the data with a model, within stringent L_∞ bounds, then we have a fair assurance that either the answer is correct, or else the problem is too badly conditioned to allow for a stable solution.

7 Conclusions

Much was learnt from this investigation. The shape of the cost functions in some important geometric vision problems was investigated, shedding light on why there may be many local minima for L_2 cost function, whereas L_∞ cost function has a simpler shape. In the problems we have considered, L_∞ optimization comes down to minimizing a cost function with a single minimum (local or global) on a convex domain. In low-dimensional problems, such as triangulation, this minimization task may be solved, reliably achieving the (global) cost minimum. For high-dimension problems, general purpose constrained minimization problems (such as LOQO) do well by finding a near optimum solution. Given the general simple nature of the cost function and the convex parameter region, we feel that an optimal solution should be possible, and we open this problem up to other researchers.

In the course of this work we obtained a firm feeling that constrained optimization has an important role in geometric computer vision, hereto untapped. Seemingly typical is the situation with projective triangulation. If the point is allowed to roam freely in 3D projective space, then there are large numbers of local minima, either for L_∞ or L_2 cost functions. However, by constraining the sought point to satisfy the cheirality constraint in the course of the minimization, the possible number of local minima is reduced very greatly (to one in the case of L_∞ cost function). Even if the initial point is in the correct region, many optimization procedures (such as Levenberg-Marquardt) may jump out of the correct region due to their way of inferring global structure from local information about the cost surface. Much more research needs to be done on constrained minimization for vision problems.

References

[1] R. I. Hartley. Chirality. *International Journal of Computer Vision*, 26(1):41–61, 1998.

[2] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, November 1997.

[3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[4] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.

[5] M. Uyttendaele, A. Criminisi, S.B. Kang, S. Winder, R. Hartley, and R. Szeliski. High-quality image-based interactive exploration of real-world environments. *IEEE Computer Graphics and Applications, to appear*, 2004.

[6] R. Vanderbei. Loqo users' manual. Technical report, Princeton University: http://www.orfe.princeton.edu/ loqo, 1997.