Optimal Algorithms in Multiview Geometry

Richard Hartley¹ and Fredrik Kahl²

¹ Research School of Information Sciences and Engineering,
The Australian National University
National ICT Australia (NICTA)*

² Centre for Mathematical Sciences, Lund University, Sweden

Abstract. This is a survey paper summarizing recent research aimed at finding guaranteed optimal algorithms for solving problems in Multiview Geometry. Many of the traditional problems in Multiview Geometry now have optimal solutions in terms of minimizing residual imageplane error. Success has been achieved in minimizing L_2 (least-squares) or L_∞ (smallest maximum error) norm. The main methods involve Second Order Cone Programming, or quasi-convex optimization, and Branch-and-bound. The paper gives an overview of the subject while avoiding as far as possible the mathematical details, which can be found in the original papers.

J.E. Littlewood: The first test of potential in mathematics is whether you can get anything out of geometry.

G.H. Hardy: The sort of mathematics that is useful to a superior engineer, or a moderate physicist has no esthetic value and is of no interest to the real mathematician.

1 Introduction

In this paper, we describe recent work in geometric Computer Vision aimed at finding provably optimal solutions to some of the main problems. This is a subject which the two authors of this paper have been involved in for the last few years, and we offer our personal view of the subject. We cite most of the relevant papers that we are aware of and apologize for any omissions. There remain still several open problems, and it is our hope that more researchers will be encouraged to work in this area.

Research in Structure from Motion since the start of the 1990s resulted in the emergence of a dominant accepted technique – bundle adjustment [47]. In this method, a geometric problem is formulated as a (usually) non-linear optimization problem, which is then solved using an iterative optimization algorithm.

* NICTA is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

Generally, the bundle adjustment problem is formulated as follows. One defines a *cost function* (also called an *objective function*) in terms of a set of *parameters*. Solving the problem involves finding the set of parameters that minimize the cost. Generally, the parameters are associated with the geometry that we wish to discover. Often they involve parameters of a set of cameras, as well as parameters (such as 3D point coordinates) describing the geometry of the scene. The cost function usually involves image measurements, and measures how closely a given geometric configuration (for instance the scene geometry) explains the image measurements.

Bundle adjustment has many advantages, which account for its success and popularity.

1. It is quite general, and can be applied to a large range of problems.
2. It is very easy to augment the cost function to include other constraints that the problem must satisfy.
3. We can “robustify” the cost function by minimizing a robust cost function, such as the Huber cost function.
4. Typically, the estimation problem is sparse, so sparse techniques can be used to achieve quite fast run times.

One of the issues with bundle adjustment is that it requires a relatively accurate initial estimate of the geometry in order to converge to a correct minimum. This requirement led to one of the other main themes of research in Multiview Geometry through the 1990s - finding reliable initial solutions usually through so-called algebraic techniques. Most well known among such techniques is the 8-point algorithm [28] for estimation of the essential or fundamental matrix, which solves the two-view relative orientation problem. Generally, in such methods, one defines an algebraic condition that must hold in the case of a noise-free solution, and defines an *algebraic cost function* that expresses how far this condition is from being met. Unfortunately, the cost function often is not closely connected with the geometry, and the cost function may be meaningless in any geometric or statistical sense.

Multiple minima. One of the drawbacks of bundle adjustment is the possibility of converging to a local, rather than a global minimum of the cost function. The cost functions that arise in multiview optimization problems commonly do have multiple local minima. As an example, in Fig 1 we show graphs of the cost functions associated with the two-view triangulation problem (described later) and the autocalibration problem.

The triangulation problem with more than two views is substantially more complex. It has been shown in [44] that the triangulation problem with three views involves solving a polynomial of degree 47, and hence the cost function potentially may have up to 24 minima. For higher numbers of views, the degree of the polynomial grows cubically. Stewénius and Nistér have calculated the sequence of degrees of the polynomial to be 6, 47, 148, 336, 638, 1081 for 2 to 7 view triangulation, which implies a large number of potential local minima of the cost function.

Fig. 1. Top: Two-view triangulation cost function, showing up to three minima. The independent variable (x -axis) parametrizes the epipolar plane. On the left, an example with three local minima (two of them with equal cost). On the right, an example with global solution with zero cost (perfect noise-free triangulation), yet having a further local minimum. The graphs are taken from [12].

Bottom: 2-dimensional cross-section of a cost associated with an autocalibration problem. The graphs are taken from [9]. Left to right: top view, side-view, contour plot. Note the great complexity of the cost function, and the expected difficulties in finding a global minimum.

As for autocalibration, progress has been made on this problem in [5], which finds optimal methods of carrying out the projective - affine - Euclidean upgrade, the “stratified” approach to autocalibration. However, this is not quite the same as an optimal solution to autocalibration, which remains an open (very difficult) problem.

Optimal methods. The difficulties and uncertainties associated with bundle adjustment, and algebraic methods has led to a theme of research in the last few years that aims at finding guaranteed provably optimal methods for solving these problems. Although such methods have not been successful for all geometric problem, the number of problems that can be solved using such optimal methods continues to grow. It is the purpose of this paper to survey progress in this area.

2 What is meant by an optimal solution?

In this section, we will argue that what is meant by an optimal solution to a geometric problem is not clearly defined. We consider a problem in which a set of measurements are made and we need to fit a parametrized model of some kind to these measurements. Optimization involves finding the set of parameters of the model that best fit the purpose. The optimization problem is defined in terms of a defined cost function which must be minimized over the set of

meaningful parameters. However, what particular cost functions merit being called “optimal” will be considered in the rest of this section.

To set up the problem, consider a set of measurements, $\mathbf{x}_i$ to which we wish to fit a parametrized model. The model gives rise to some *model* values $\hat{\mathbf{x}}_i$ which are defined in some way in terms of the parametrization. The set of *residuals* δ_i are defined as $\delta_i = \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|$ where $\|\cdot\|$ represents a suitable norm in the measurement space. For image measurements, this is most reasonably the distance in the image between $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$. Finally, denote by Δ the vector with components δ_i . This may be called the *vector of residuals*.

2.1 L_2 error and ML estimation

Often, it is argued that the optimal solution is the least squares solution, which minimizes the cost function

$$\|\Delta\|_2 = \sum_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 ,$$

namely the L_2 -norm of the vector of residuals.³ The argument for the optimality of this solution is as follows. If we assume that the measurements are derived from actual values, corrupted with Gaussian noise with variance σ^2 , then the probability of the set of measurements $\mathbf{x}_i$ given true values $\hat{\mathbf{x}}_i$ is given by

$$P(\{\mathbf{x}_i\} | \{\hat{\mathbf{x}}_i\}) = K \prod_i \exp(-\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 / (2\sigma^2)) .$$

where K is a normalizing constant. Taking logarithms and minimizing, we see that the modelled data that maximizes the probability of the measurements also minimizes $\sum_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2$. Thus, the least-squares solution is the maximum likelihood (ML) estimate, under an assumption of Gaussian noise in the measurements. This is the argument for optimality of the least-squares solution. Although useful, this argument does rely on two assumptions that are open to question.

1. It makes an assumption of Gaussian noise. This assumption is not really justified at all. Measurement errors in a digital image are not likely to satisfy a Gaussian noise model.
2. Maximum likelihood is not necessarily the same as optimal, a term that we have not defined. One might define optimal to mean maximum likelihood, but this is a circular argument.

2.2 L_∞ error

An alternative noise model, perhaps equally justified for image point measurements is that of uniform bounded noise. Thus, we assume that all measurements

³ In future the symbol $\|\cdot\|$ represents the 2-norm of a vector.

less than a given threshold distance from the true value are equally likely, but measurements beyond this threshold have probability zero. In the case of a discrete image, measurements more accurate than one pixel from the true value are difficult to obtain, though in fact they may be achieved in carefully controlled situations.

If we assume that the measurement error probability model is

$$P(\mathbf{x} \mid \hat{\mathbf{x}}) = K \exp(-(\|\mathbf{x} - \hat{\mathbf{x}}\|/\sigma)^p) \quad (1)$$

where K is a normalizing factor, then as before, the ML estimate is the one that minimizes $\sum_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^p$. Letting p increase to infinity, the probability distribution (1) converges uniformly (except at σ) to a uniform distribution for $\mathbf{x}$ within distance σ of $\hat{\mathbf{x}}$. Now, taking the p -th root, we see that minimizing (1) is equivalent to normalizing the p -norm of the vector Δ of residuals. Furthermore, as p increases to infinity, $\|\Delta\|_p$ converges to the L_∞ norm $\|\Delta\|_\infty$. In this way, minimizing the L_∞ error corresponds to an assumption of uniform distribution of measurement errors.

Looked at another way, under the L_∞ norm, all sets of measurements that are within a distance σ of the modelled values are equally likely, whereas a set of measurements where one of the values exceeds the threshold σ has probability zero. Then L_∞ optimization finds the smallest noise threshold for which the set of measurements is possible, and determines the ML estimate for this minimum threshold.

Note that the L_∞ norm of a vector is simply the largest component of the vector, in absolute value. Thus,

$$\min \|\Delta\|_\infty = \min \max_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\| ,$$

where the minimization is taken over the parameters of the model. For this reason, L_∞ minimization is sometimes referred to as minimax optimization.

2.3 Other criteria for optimality

It is not clear that the maximum likelihood estimate has a reasonable claim to being the optimal estimate. It is pointed out in [13] that the maximum likelihood estimate may be biased, and in fact have infinite bias even for quite simple estimation problems. Thus, as the noise level of measurements increases, the average (or expected) estimate drifts away from the true value. This is of course undesirable.

In addition, a different way of thinking of the estimation problem is in terms of the risk of making a wrong estimate. For example consider the triangulation problem (discussed in more detail later) in which several observers estimate a bearing (direction vector) to a target from known observation points. If the bearing directions are noisy, then where is the target?

In many cases, there is a cost associated with wrongly estimating the position of the target. (For instance, if the target is an incoming ballistic missile, the cost

of a wrong estimate can be quite high.) A reasonable procedure would be to choose an estimate that minimizes the expected cost. As an example, if the cost of an estimate is equal to the square of the distance between the estimate and the true value, then the expected cost is equal to the mean of the posterior probability distribution of the parameters, $P(\theta \mid \{\mathbf{x}_i\})$ ⁴

More discussion of these matters is contained in [13], Appendix 3. We are not, however, aware of any literature in multiview geometry finding estimates of this kind.

What we mean by optimality. In this survey, we will consider the estimates that minimize the L_2 or L_∞ norms of the residual error vector, with respect to a parametrized model as being the optimal. This is reasonable in that it is related in either case to a specific geometric noise model, tied directly to the statistics of the measurements.

3 Polynomial methods

One approach for obtaining optimal solutions to multiview problems is to compute all stationary points of the cost function and then check which of these is the global minimum. From a theoretical point of view, any structure and motion problem can be solved in this manner as long as the cost function can be expressed as a rational polynomial function in the parameters. This will be the case for most cost functions encountered (though not for L_∞ cost functions, which are not differentiable).

The method is as follows. A minimum of the cost function must occur at a point where the derivatives of the cost with respect to the parameters vanish. If the cost function is a rational polynomial function, then the derivatives are rational as well. Setting the derivatives to zero leads to a system of polynomial equations, and the solutions of this set of equations define the stationary points of the cost function. These can be checked one-by-one to find the minimum. This method may also be applied when the parameters satisfy certain constraints, such as a constraint of zero determinant for the fundamental matrix, or a constraint that a matrix represents a rotation. Such problems can be solved using Lagrange multipliers.

Although this method is theoretically generally applicable, in practice it is only tractable for small problems, for example the triangulation problem. A solution to the two-view triangulation problem was given in [12], involving the solution of a polynomial of degree 6. The three-view problem was addressed in [44]; the solution involves the solution of a polynomial of degree 47. Further work (unpublished) by Stewénus and Nistér has shown that the triangulation problem for 4 to 7 views can be solved by finding the roots of a polynomial of degree 148, 336, 638, 1081 respectively, and in general, the degree grows cubically. Since

⁴ This assumes that the parameters θ are in a Euclidean space, which may not always be the case. In addition, estimation of the posterior probability distribution may require the definition of a prior $P(\theta)$.

solving large sets of polynomials is numerically difficult, the issue of accuracy has been addressed in [4].

The polynomial method may also be applied successfully in many minimal-configuration problems. We do not attempt here to enumerate all such problems considered in the literature. One notable example, however, is the relative orientation (two-view reconstruction) problem with 5 point correspondences. This has long been known to have 10 solutions [7, 16].⁵ The second of these references gives a very pleasing derivation of this result. Recent simple algorithms for solving this problem by finding the roots of a polynomial have been given in [31, 26].

Methods for computing a polynomial solution need not result in a polynomial of the smallest possible degree. However recent work using Galois theory [32] gives a way to address this question, showing that the 2-view triangulation and the relative orientation problem essentially require solution of polynomials of degree 6 and 10 respectively.

4 L_∞ estimation and SOCP

In this section, we will consider L_∞ optimization, and discuss its advantages vis-a-vis L_2 optimization. We show that there are many problems that can be formulated in L_∞ and give a single solution. This is the main advantage, and contrasts with L_2 optimization, which may have many local minima, as was shown in Fig 1.

4.1 Convex optimization

We start by considering convex optimization problems. First, a few definitions.

Convex set. A subset S of $\mathbb{R}^n$ is said to be *convex* if the line segment joining any two point in S is contained in S . Formally, if $\mathbf{x}_0, \mathbf{x}_1 \in S$, then $(1-\alpha)\mathbf{x}_0 + \alpha\mathbf{x}_1 \in S$ for all α with $0 \leq \alpha \leq 1$.

Convex function. A function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is *convex* if its domain is a convex set and for all $\mathbf{x}, \mathbf{y} \in \text{domain}(f)$, and α with $0 \leq \alpha \leq 1$, we have

$$f((1-\alpha)\mathbf{x}_0 + \alpha\mathbf{x}_1) \leq (1-\alpha)f(\mathbf{x}_0) + \alpha f(\mathbf{x}_1) .$$

Another less formal way of defining a convex function is so say that a line joining two points on the graph of the function will always lie above the function. This is illustrated in Fig 2.

⁵ These papers find 20 solutions, not 10, since they are solving for the number of possible rotations. There are 10 possible essential matrices each of which gives two possible rotations, related via the *twisted pair ambiguity* (see [13]). Only one of these two rotations is cheirally correct, corresponding to a possible realizable solution.

Fig. 2. **Left.** Examples of convex and non-convex sets. **Middle.** The definition of a convex function; the line joining two points lies above the graph of the function. **Right.** Convex optimization.

Convex optimization. A convex optimization problem is as follows:

- Given a convex function $f : D \rightarrow \mathbb{R}$, defined on a convex domain $D \subset \mathbb{R}^n$, find the minimum of f on D .

A convex function is always continuous, and given reasonable conditions that ensure a minimum of the function (for instance D is compact) such a convex optimization problem is solvable by known algorithms⁶. A further desirable property of a convex problem is that it has no local minima apart from the global minimum. The global minimum value is attained at a single point, or at least on a convex set in $\mathbb{R}^n$ where the function takes the same minimum value at all points. For further details, we refer the reader to the book [3].

Quasi-convex functions. Unfortunately, although convex problems are agreeable, they do not come up often in multiview geometry. However, interestingly enough, certain other problems do, quasi-convex problems. Quasi-convex functions are defined in terms of *sublevel sets* as follows.

Definition 1. A function $f : D \rightarrow \mathbb{R}$ is called quasi-convex if its α -sublevel set,

$$S_\alpha = \{\mathbf{x} \in D \mid f(\mathbf{x}) \leq \alpha\}$$

is convex for all α .

Examples of quasi-convex and non-quasi-convex functions are shown in Fig 3.

Quasi-convex functions have two important properties.

1. A quasi-convex function has no local minima apart from the global minimum. It will attain its global minimum at a single point or else on a convex set where it assumes a constant value.

⁶ The most efficient algorithm to use will depend on the form of the function f , and the way the domain D is specified.

Fig. 3. *Quasi-convex functions. The left two functions are quasi-convex. All the sublevel sets are convex. The function on the right is not quasi-convex. The indentation in the function graph (on the left) means that the sublevel-sets are not convex. All convex functions are quasi-convex, but the example on the left shows that the converse is not true.*

2. The pointwise maximum of a set of quasi-convex functions is quasi-convex. This is illustrated in Fig 4 for the case of functions of a single variable. The general case follows directly from the following observation concerning sublevel sets.

$$S_\delta(\max_i f_i) = \bigcap_i S_\delta(f_i)$$

which is convex, since each $S_\delta(f_i)$ is convex.

Fig. 4. *The pointwise maximum of a set of quasi-convex functions is quasi-convex.*

A quasi-convex optimization problem is defined in the same way as a convex optimization problem, except that the function to be minimized is quasi-convex. Nevertheless, quasi-convex optimization problems share many of the pleasant properties of convex optimization.

Why consider quasi-convex functions? The primary reason for considering such functions is that the residual of a measured image point $\mathbf{x}$ with respect to the projection of a point $\mathbf{X}$ in space is a quasi-convex function. In other words, $f(\mathbf{X}) = d(\mathbf{x}, \mathbf{P}\mathbf{X})$ is a quasi-convex function of $\mathbf{X}$. Here, $\mathbf{P}\mathbf{X}$ is the projection of a point $\mathbf{X}$ into an image, and $d(\cdot, \cdot)$ represents distance in the image.

Specifically, the sublevel set $S_\delta(f(\mathbf{X}))$ is a convex set, namely a cone with vertex the centre of projection, as will be discussed in more detail shortly (see Fig 5).

As the reader may easily verify by example, however, the sum of quasi-convex functions is not in general quasi-convex. If we take several image measurements then the sum of squares of the residuals will not in general be a quasi-convex function. In other words, an L_2 cost function of the form $\sum_{i=1}^N f_i(\mathbf{X})^2$ will not in general be a quasi-convex function of $\mathbf{X}$, nor have a single minimum.

On the other hand, as remarked above, the maximum of a set of quasi-convex functions is quasi-convex, and hence will have a single minimum. Specifically, $\max_i f_i(\mathbf{X})$ will be quasi-convex, and have a single minimum with respect to $\mathbf{X}$. For this reason, it is typically easier to solve the minimax problem $\min_{\mathbf{X}} \max_i f_i(\mathbf{X})$ than the corresponding least-squares (L_2) problem.

Example: The triangulation problem. The triangulation problem is the most simple problem in multiview geometry. Nevertheless, in the L_2 formulation, it still suffers from the problem of local minima, as shown in Fig 1. In this problem, we have a set of known camera centres $\mathbf{O}_i$ and a set of direction vectors $\mathbf{v}_i$ which give the direction of a target point $\mathbf{X}$ from each of the camera centres. Thus, nominally, $\mathbf{v}_i = (\mathbf{X} - \mathbf{O}_i) / \|\mathbf{X} - \mathbf{O}_i\|$. The problem is to find the position of the point $\mathbf{X}$. We choose to solve this problem in L_∞ norm, in other words, we seek the point $\mathbf{X}$ that minimizes the maximum error (over all i) between $\mathbf{v}_i$ and the directions given by $\mathbf{X} - \mathbf{O}_i$.

Consider Fig 5. Some simple notation is required. Define $\angle(\mathbf{X} - \mathbf{O}_i, \mathbf{v}_i)$ to be the angle between the vectors $\mathbf{X} - \mathbf{O}_i$ and $\mathbf{v}_i$. Given a value $\delta > 0$, the set of points

$$C_\delta(\mathbf{O}_i, \mathbf{v}_i) = \{\mathbf{X} \mid \angle(\mathbf{X} - \mathbf{O}_i, \mathbf{v}_i) \leq \delta\}$$

forms a cone in $\mathbb{R}^3$ with vertex $\mathbf{O}_i$, axis $\mathbf{v}_i$ and angle determined by δ .

We begin by hypothesizing that there exists a solution $\mathbf{X}$ to the triangulation problem for which the maximum error is δ . In this case, the point $\mathbf{X}$ must lie inside cones in $\mathbb{R}^3$ with vertex $\mathbf{O}_i$, axis $\mathbf{v}_i$ and angle determined by δ . If the cones are too narrow, they do not have a common intersection and there can be no solution with maximum error less than δ . On the other hand, if δ is sufficiently large, then the cones intersect, and the desired solution $\mathbf{X}$ must lie in the intersection of the cones.

The optimal value of δ is found by a binary search over values of δ to find the smallest value such that the cones $C_\delta(\mathbf{O}_i, \mathbf{v}_i)$ intersect in at least one point. The intersection will be a single point, or in special configurations, a segment of a line.

The problem of determining whether a set of cones have non-empty intersection is solved by a convex optimization technique called Second Order Cone Programming (SOCP), for which open source libraries exist [45].

We make certain observations about this problem:

1. Each cone $C_\delta(\mathbf{O}_i, \mathbf{v}_i)$ is a convex set, and hence their intersection is convex.

Fig. 5. The triangulation problem: Assuming that the maximum reprojection error is less than some value δ , the sought point $\mathbf{X}$ must lie in the intersection of a set of cones. If δ is set too small, then the cones do not have a common intersection (**left**). If δ is set too large, then the cones intersect in a convex region in space, and the desired solution $\mathbf{X}$ must lie in this region (**right**). The optimal value of δ lies between these two extremes, and can be found by a binary search (bisection) testing successive values of δ . For more details, refer to the text.

2. If we define a cost function

$$\text{Cost}_\infty(\mathbf{X}) = \max_i \angle(\mathbf{X} - \mathbf{O}_i, \mathbf{v}_i) ,$$

then the sublevel set $S_\delta(\text{Cost}_\infty)$ is simply the intersection of the cones $C_\delta(\mathbf{O}_i, \mathbf{v}_i)$, which is convex for all δ . This by definition says that $\text{Cost}_\infty(\mathbf{X})$ is a quasi-convex function of $\mathbf{X}$.

3. Finding the optimum

$$\min_{\mathbf{X}} \text{Cost}_\infty(\mathbf{X}) = \min_{\mathbf{X}} \max_i \angle(\mathbf{X} - \mathbf{O}_i, \mathbf{v}_i)$$

is accomplished by a binary search over possible values of δ , where for each value of δ we solve an *SOCP feasibility problem*, (determine whether a set of cones have a common intersection). Such a problem is known as a minimax or L_∞ optimization problem.

Generally speaking, this procedure generalizes to arbitrary quasi-convex optimization problems; they may be solved by binary search involving a convex feasibility problem at each step. If we have a set of individual cost functions $f_i(\mathbf{X})$, perhaps each associated with a single measurement, and each of them quasi-convex, then the maximum of these cost functions $\max_i f_i(\mathbf{X})$ is also quasi-convex, as illustrated in Fig 4. In this case, the minimax problem of finding $\min_{\mathbf{X}} \max_i f_i(\mathbf{X})$ is solvable by binary search.

Reconstruction with known rotations. Another problem that may be solved by very similar means to the triangulation problem is that of Structure and Motion with known rotations, which is illustrated in Fig 6.

Fig. 6. Structure and motion with known rotations. *If the orientations of several cameras are all known, then image points correspond to direction vectors in a common coordinate frame. Here, blue (or grey) circles represent the positions of cameras, and black circles the positions of points. The arrows represent direction vectors (their length is not known) from cameras to points. The positions of all the points and cameras may be computed (up to scale and translation) using SOCP.*

Abstractly, this is the embedding problem for a bi-partite graph in 3D, where the orientation of the edges of the graph is known. The analogous problem for an arbitrary (not bi-partite) graph was applied in [42] to solve for motion of the cameras without computing the point positions.

The role of cheirality. It is important in solving problems of this kind to take into account the concept of “cheirality”, which means the requirement that points visible in an image must lie in front of the camera, not behind. If we subdivide space by planes separating front and back of the camera, then there will be at least one local minimum of the cost function (whether L_∞ or L_2) in each region of space. Since the number of regions separated by n planes grows cubically, so do the number of local minima, unless we constrain the solution so that the points lie in front of the cameras.

Algorithm acceleration. Although the bisection algorithm using SOCP has been the standard approach to L_∞ geometric optimization problems, there has been recent work for speeding up the computations [40, 41, 34]. However, it has been shown that the general structure and motion (with missing data) is *NP*-hard no matter what criterion of optimality of reprojection error is used [33].

4.2 Problems solved in L_∞ and L_2 norm

The list of problems that can be solved globally with L_∞ estimation continues to grow and by now it is quite long, see Table 1. We also include a list of problems solvable under L_2 norm. The list is smaller, but has grown substantially recently.

In [21], an L_∞ estimation algorithm serves as the basis for solving the least-median squares problem for various geometric problems. However, the extension to such problems is essentially based on heuristics and it has no guarantee of finding the global optimum.

L_∞ -norm	References
– Multiview triangulation	[11, 18, 20, 6]
– Camera resectioning (uncalibrated case)	[18, 20]
– Camera pose (calibrated case)	[10, 48]
– Homography estimation	[18, 20]
– Structure and motion recovery with known camera orientation	[11, 18, 20]
– Reconstruction by using a reference plane	[18]
– Camera motion recovery	[42]
– Outlier detection	[43, 20, 25, 48]
– Reconstruction with covariance-based uncertainty	[42, 21]
– Two-view relative orientation	[10]
– 1D retinal vision	[2]
L_2 -norm	References
– Affine reconstruction from affine cameras	[23, 46]
– Multiview triangulation	[1, 29, 17]
– Camera resectioning (uncalibrated case)	[1]
– Camera pose (calibrated case)	[35]
– Homography estimation	[1]
– 3D – 3D registration	[14, 36]
– 3D – 3D registration and matching (unknown pairing)	[27]

Table 1. *Geometric reconstruction problems that can be solved globally with the L_∞ or L_2 norm.*

5 Branch-and-bound theory

The method based on L_∞ optimization is not applicable to all problems. In this section, we will describe in general terms a different method that has been used with success in obtaining globally optimal solutions.

Branch and bound algorithms are non-heuristic methods for global optimization in non-convex problems. They maintain a provable upper and/or lower bound on the (globally) optimal objective value and terminate with a certificate proving that the solution is within ϵ of the global optimum, for arbitrarily small ϵ .

Consider a non-convex, scalar-valued objective function $\Phi(x)$, for which we seek a global minimum over a domain Q_0 . For a subdomain $Q \subseteq Q_0$, let $\Phi_{\min}(Q)$ denote the minimum value of the function Φ over Q . Also, let $\Phi_{\text{lb}}(Q)$ be a function that computes a lower bound for $\Phi_{\min}(Q)$, that is, $\Phi_{\text{lb}}(Q) \leq \Phi_{\min}(Q)$.

An intuitive technique to determine the solution would be to divide the whole search region Q_0 into a grid with cells of sides δ and compute the minimum of a lower bounding function Φ_{lb} defined over each grid cell, with the presumption that each $\Phi_{\text{lb}}(Q)$ is easier to compute than the corresponding $\Phi_{\min}(Q)$. However, the number of such grid cells increases rapidly as $\delta \rightarrow 0$, so a clever procedure must be deployed to create as few cells as possible and “prune” away as many of these grid cells as possible (without having to compute the lower bounding function for these cells).

Fig. 7. This figure illustrates the operation of a branch and bound algorithm on a one dimensional non-convex minimization problem. Figure (a) shows the function $\Phi(x)$ and the interval $l \leq x \leq u$ in which it is to be minimized. Figure (b) shows the convex relaxation of $\Phi(x)$ (indicated in yellow/dashed), its domain (indicated in blue/shaded) and the point for which it attains a minimum value. q_1^* is the corresponding value of the function Φ . This value is the current best estimate of the minimum of $\Phi(x)$, and is used to reject the left subinterval in Figure (c) because the minimum value of the convex relaxation is higher than q_1^* . Figure (d) shows the lower bounding operation on the right sub-interval in which a new estimate q_2^* of the minimum value of $\Phi(x)$ is found.

Branch and bound algorithms iteratively subdivide the domain into subregions (which we refer to as rectangles) and employ clever strategies to prune away as many rectangles as possible to restrict the search region.

A graphical illustration of the algorithm is presented in Fig 7. Computation of the lower bounding functions is referred to as *bounding*, while the procedure that chooses a domain and subdivides it is called *branching*. The choice of the domain picked for refinement in the branching step and the actual subdivision itself are essentially heuristic.

Although guaranteed to find the global optimum (or a point arbitrarily close to it), the worst case complexity of a branch and bound algorithm is exponential. However, in many cases the properties offered by multiview problems lead to fast convergence rates in practice.

6 Branch-and-bound for L_2 minimization

The branch-and-bound method can be applied to find L_2 norm solutions to certain simple problems. This is done by a direct application of branch-and-bound over the parameter space. Up to now, methods used for branching have been quite simple, consisting of simple subdivision of rectangles in half, or in half along all dimensions.

In order to converge as quickly as possible to the solution, it is useful to confine the region of parameter space that needs to be searched. This can be conveniently done if the cost function is a sum of quasi-convex functions. For instance, suppose the cost is $C_2(\mathbf{X}) = \sum_i f_i(\mathbf{X})^2$, and the optimal value is denoted

by $\mathbf{X}_{\text{opt}}$. If a good initial estimate $\mathbf{X}_0$ is available, with $C_2(\mathbf{X}_0) = \delta^2$, then

$$C_2(\mathbf{X}_{\text{opt}}) = \sum_i f_i(\mathbf{X}_{\text{opt}})^2 \leq C_2(\mathbf{X}_0) = \delta^2 .$$

This implies that each $f_i(\mathbf{X}_{\text{opt}}) \leq \delta$, and so $\mathbf{X}_{\text{opt}} \in \bigcap_i S_\delta(f_i)$, which is a convex set enclosing both $\mathbf{X}_0$ and $\mathbf{X}_{\text{opt}}$. One can find a rectangle in parameter space that encloses this convex region, and begin the branch-and-bound algorithm starting with this rectangle.

This general method was used in [1] to solve the L_2 multiview triangulation problem and the uncalibrated camera resection (pose) problem. In that paper fractional programming (described later in section 8.2) was used to define the *convex sub-envelope* of the cost function, and hence provide a cost lower bound on each rectangle. The same branch-and-bound idea, but with a different bounding method, was described in [29] to provide a simpler solution to the triangulation problem. The triangulation problem for other geometric features, more specifically lines and conics, was solved in [17] using the same approach.

7 Branch-and-bound in rotation space

7.1 Essential matrix

All the problems that we have so-far considered, solvable using quasi-convex optimization or SOCP, involved no rotations. If there are rotations, then the optimization problem is no longer quasi-convex. An example of this type of problem is estimation of the essential matrix from a set of matching points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. A linear solution to this problem was given in 1981 by Longuet-Higgins [28]. From the Essential matrix $\mathbf{E}$, that satisfies the defining equation $\mathbf{x}'_i{}^\top \mathbf{E} \mathbf{x}_i = 0$ for all i , we can extract the relative orientation (rotation and translation) of the two cameras.

To understand why this is not a quasi-convex optimization problem, we look at the minimal problem involving 5 point correspondences. It is well known that with 5 points, there are 10 solutions for the relative orientation. (Recent algorithms for solving the 5-point orientation are given in [31, 26]). However, if there are many possible discrete solutions, then the problem can not be quasi-convex or convex – such problems have a unique solution. This is so whether we are seeking an L_∞ or L_2 solution, since the 5-point problem has an exact solution, and hence the cost is zero in either norm.

Many algorithms for estimating the essential matrix have been given, without however any claim to optimality. Recently, an optimal solution, at least in L_∞ -norm was given in [10].

To solve the essential matrix problem optimally (at least in L_∞ norm) we make the following observation. If the rotation of the two cameras were somehow known, then the problem would reduce to the one discussed in Fig 6, where the translation of the cameras can be estimated optimally (in L_∞ norm) given the rotations. The residual cost of this optimal solution may be found, as a function

of the assumed rotation. To solve for the relative pose (rotation and translation) of the two cameras, we merely have to consider all possible rotations, and select the one that yields the smallest residual. The trick is to do this without having to look at an infinite number of rotations.

Fortunately, branch-and-bound provides a means of carrying out this search. A key to the success of branch-and-bound is that the optimal cost (residual) estimated for one value of the rotation constrains the optimal cost for “nearby” rotations. This allows us to put a lower bound on the optimal cost associated with all rotations in a region of rotation space.

The branch-and-bound algorithm carries out a search of rotation space (a 3-dimensional space). The translation is not included in the branch-and-bound search. Instead, for a given rotation, the optimal translation may be computed using SOCP, and hence factored out of the parameter-space search.

A similar method of search over rotation space is used in [10] to solve the calibrated camera pose problem. This is the problem of finding the position and orientation of a camera given known $3D$ points and their corresponding image points. An earlier iterative algorithm that addresses this problem using L_∞ norm is given in [48]. The algorithm appears to converge to the global optimum, but the author states that this is unproven.

7.2 General structure and motion

The method outlined in section 7.1 for solving the structure and motion problem for two views could in principle be extended to give an optimal solution (albeit in L_∞ norm) to the complete structure and motion problem for any number of views.

This would involve a search over the combined space of all rotations. For the two camera problem there is only one relative rotation, and hence the branch-and-bound algorithm involves a search over a 3-dimensional parameter space. In the case of n cameras, however the parameter space is $3(n - 1)$ -dimensional.

Had we but world enough and time a branch-and-bound search over the combined rotation parameter space would yield an optimal L_∞ solution to structure and motion with any number of cameras and points. Unfortunately, in terms of space and time requirements, this algorithm would be vaster than empires and more slow[30].

7.3 1D retinal vision

One-dimensional cameras have proven useful in several different applications, most prominently for autonomous guided vehicles (see Fig 8), but also in ordinary vision for analysing planar motion and the projection of lines. Previous results on one-dimensional vision are limited to classifying and solving of minimal cases, bundle adjustment for finding local minima to the structure and motion problem and linear algorithms based on algebraic cost functions.

A method for finding the global minimum to the structure and motion problem using the max norm of reprojection errors is given in [2]. In contrast to

the 2D case which uses SOCP, the optimal solution can be computed efficiently using simple linear programming techniques.

Fig. 8. *Left: A laser guided vehicle. Middle: A laser scanner or angle meter. Right: Calculated structure and motion for the icehockey experiment.*

It is assumed that neither the positions of the objects nor the positions and orientations of the cameras are known. However, it is assumed that the correspondence problem is solved, i.e., it is known which measured bearings correspond to the same object. The problem can formally be stated as follows. Given n bearings from m different positions, find the camera positions and 2D points in the plane, such that the reprojected solution has minimal residual errors. The norm for measuring the errors will be the L_∞ norm.

The basic idea of the optimization scheme is to first consider optimization with fixed camera orientations - which is a quasi-convex problem - and then use branch-and-bound over the space of possible orientations, similar to that of section 7.1.

Hockey rink data. By combining optimal structure and motion with optimal resection and intersection it is possible to solve for arbitrarily many cameras and views. We illustrate this with the data from a real set of measurements performed at an ice-hockey rink. The set contains 70 images of 14 points. The result is shown in the right of Fig 8.

7.4 3D – 3D alignment

Unknown point correspondences. A similar method of doing branch-and-bound in rotation space was used in [27] to find an optimal solution for the

problem of aligning two sets of points in $3D$ with unknown pairing (or correspondences). The solution consists of a specified pairing of the two point sets, along with a rotation and translation to align the paired points.

The algorithm relies on the fact that if the rotation is known, then the optimal pairing can be computed directly using the Hungarian algorithm [37]. This enables the problem to be addressed by a branch-and-bound search over rotations. The problem is solved for L_2 norm in [27] using a bounding method based on Lipschitz bounds. Though the L_∞ problem is not specifically addressed in the paper, it would probably also yield to a similar approach.

Known correspondences of points, lines and planes. While aligning two sets of $3D$ points with known pairings can be done in closed-form [14], the optimization problem for other feature correspondences is much harder. In [36], an optimal solution was given based on branch and bound, but with a slightly different approach compared to the other methods described above. The unknown rotation is parametrized with a unit quaternion and bounds are obtained by computing the subenvelope of the quadratic quaternion terms appearing in the objective function. The (calibrated) camera pose problem was also solved using a similar technique in [35].

7.5 An open problem: Optimal essential matrix estimation in L_2 norm

The question naturally arises of whether we can use similar techniques to section 7.1 to obtain the optimal L_2 solution for the essential matrix. At present, we have no solution to this problem. Two essential steps are missing.

1. If the relative rotation $\mathbf{R}$ between the two cameras is given, can we estimate the optimal translation $\mathbf{t}$. This is simple in L_∞ norm using SOCP, or in fact linear programming. In L_2 norm, a solution has been proposed in [15], but still it is iterative, and it is not clear that it is guaranteed to converge. For n points, this seems to be a harder problem than optimal n -view L_2 triangulation (for which solutions have been recently given [1, 29]).
2. If we can find the optimal residual for a given rotation, how does this constrain the solution for nearby rotations? Loose bounds may be given, but they may not be sufficiently tight to allow for efficient convergence of the branch-and-bound algorithm.

8 Other methods for minimizing L_2 norm

8.1 Convex relaxations and semidefinite programming

Another general approach for solving L_2 problems was introduced in [19] based on convex relaxations (underestimators) and semidefinite programming.

More specifically, the approach is based on a hierarchy of convex relaxations to solve non-convex optimization problems. Linear matrix inequalities (LMIs) are

used to construct the convex relaxations. These relaxations generate a monotone sequence of lower bounds of the minimal value of the objective function and it is shown how one can detect whether the global optimum is attained at a given relaxation. Standard semidefinite programming software (like SeDuMi [45]) is extensively used for computing the bounds.

The technique is applied to a number of classical vision problems: triangulation, camera pose, homography estimation and epipolar geometry estimation. Although good results are obtained, there is no guarantee of achieving the optimal solution and the sizes of the problem instances are small.

8.2 Fractional programming

Yet another method was introduced in [1, 35]. It was the first method to solve the n -view L_2 triangulation problem with a guarantee of optimality. Other problem applications include camera resectioning (that is, uncalibrated camera pose), camera pose estimation and homography estimation.

In its most general form, fractional programming seeks to minimize/maximize the sum of fractions subject to convex constraints. Our interest from the point of view of multiview geometry, however, is specific to the minimization problem

$$\min_{\mathbf{x}} \sum_{i=1}^p \frac{f_i(\mathbf{x})}{g_i(\mathbf{x})} \quad \text{subject to } \mathbf{x} \in D$$

where $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ are *convex* and *concave* functions, respectively, and the domain $D \subset \mathbb{R}^n$ is a convex compact set. This is because the residual of the projection of a 3D point into an image may be written in this form. Further, it is assumed that both f_i and g_i are positive with lower and upper bounds over D . Even with these restrictions the above problem is *NP*-complete [8], but practical and reliable estimation of the global optimum is still possible for many multiview problems through an iterative algorithm that solve an appropriate convex optimization problem at each step. The procedure is based on branch and bound.

Perhaps the most important observation made in [1] is that many multiview geometry problems can be formulated as a sum of fractions where each fraction consists of a convex over a concave function. This has inspired a new more efficient ways of computing the L_2 -minimum for n -view triangulation, see [29].

9 Applications

There have been various application papers that have involved this type of optimization methodology, though they can not be said to have found an optimal solution to the respective problems.

In [39] SOCP has been used to solve the problem of tracking and modelling a deforming surface (such as a sheet of paper) from a single view. Results are shown in Fig 9.

Fig. 9. *Modelling a deforming surface from a single view. Left: the input image, with automatically overlaid grid. Right: the computed surface model viewed from a new viewpoint.*

Image features provide cone constraints that constrain the corresponding 3D points to lie on or near the corresponding line-of-sight, namely a ray through the camera centre. Additional convex constraints on the geometry of the surface allow the shape to be determined unambiguously using SOCP.

In another application-inspired problem, SOCP has been applied (see [22]) to the odometry problem for a vehicle with several rigidly mounted cameras with almost non-overlapping fields of view. Although the algorithm in [22] is tested on laboratory data, it is motivated by its potential use with vehicles such as the one shown in Fig 10. Such vehicles are used for urban-mapping. Computation of individual essential matrices for each of the cameras reduces the computation of the translation of the vehicle to a multiple view triangulation problem, which is solved using SOCP.

Fig. 10. *Camera and car mount used for urban mapping. Images from non-overlapping cameras on both sides of the car can be used to do odometry of the vehicle. An algorithm based on triangulation and SOCP is proposed in [22]. (The image is used with permission of the UNC Vision group).*

10 Concluding remarks

The application of new optimization methods to the problems of Multiview Geometry have led to the development of reliably and provably optimal solutions

under different geometrically meaningful cost functions. At present these algorithms are not as fast as standard methods, such as bundle adjustment. Nevertheless the run times are not wildly impractical. Recent work on speeding up the optimization process is yielding much faster run times, and further progress is likely.

Such optimization techniques are also being investigated in other areas of Computer Vision, such as discrete optimization. A representative paper is [24].

For 15 years or more, geometric computer vision has relied on a small repertoire of optimization methods, with Levenberg-Marquardt [38] being the most popular. The benefit of using new methods such as SOCP and other convex and quasi-convex optimization methods is being realised.

References

1. S. Agarwal, M. K. Chandraker, F. Kahl, D. J. Kriegman, and S. Belongie. Practical global optimization for multiview geometry. In *European Conf. Computer Vision*, pages 592–605, Graz, Austria, 2006.
2. K. Åström, O. Enqvist, C. Olsson, F. Kahl, and R. Hartley. An L_∞ approach to structure and motion problems in 1d-vision. In *Int. Conf. Computer Vision*, Rio de Janeiro, Brazil, 2007.
3. S. Boyd and L. Vanderberghe. *Convex Optimization*. Cambridge University Press, 2004.
4. M. Byröd, K. Josephson, and K. Åström. Improving numerical accuracy in gröbner basis polynomial equation solvers. In *Int. Conf. Computer Vision*, Rio de Janeiro, Brazil, 2007.
5. M. K. Chandraker, S. Agarwal, D. J. Kriegman, and S. Belongie. Globally convergent algorithms for affine and metric upgrades in stratified autocalibration. In *Int. Conf. Computer Vision*, Rio de Janeiro, Brazil, 2007.
6. M. Farenzena, A. Fusiello, and A. Dovic. Reconstruction with interval constraints propagation. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 1185–1190, New York City, USA, 2006.
7. O. D. Faugeras and S. J. Maybank. Motion from point matches: Multiplicity of solutions. *Int. Journal Computer Vision*, 4:225–246, 1990.
8. R. W. Freund and F. Jarre. Solving the sum-of-ratios problem by an interior-point method. *J. Glob. Opt.*, 19(1):83–102, 2001.
9. R. Hartley, L. de Agapito, E. Hayman, and I. Reid. Camera calibration and the search for infinity. In *Proc. 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 510–517, September 1999.
10. R. Hartley and F. Kahl. Global optimization through searching rotation space and optimal estimation of the essential matrix. In *Int. Conf. Computer Vision*, Oct 2007.
11. R. Hartley and F. Schaffalitzky. L_∞ minimization in geometric reconstruction problems. In *Conf. Computer Vision and Pattern Recognition*, volume I, pages 504–509, Washington DC, USA, 2004.
12. R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.
13. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision – 2nd Edition*. Cambridge University Press, 2004.

14. B. K. P. Horn. Closed form solution of absolute orientation using unit quaternions. *J. Opt. Soc. America*, 4(4):629–642, 1987.
15. B. K. P. Horn. Relative orientation. *Int. Journal Computer Vision*, 4:59–78, 1990.
16. B. K. P. Horn. Relative orientation revisited. *J. Opt. Soc. America*, 8(10):1630–1638, 1991.
17. K. Josephson and F. Kahl. Triangulation of points, lines and conics. In *Scandinavian Conf. on Image Analysis*, Aalborg, Denmark, 2007.
18. F. Kahl. Multiple view geometry and the L_∞ -norm. In *Int. Conf. Computer Vision*, pages 1002–1009, Beijing, China, 2005.
19. F. Kahl and D. Henrion. Globally optimal estimates for geometric reconstruction problems. *Int. Journal Computer Vision*, 74(1):3–15, 2007.
20. Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. In *Int. Conf. Computer Vision*, pages 986 – 993, Beijing, China, 2005.
21. Q. Ke and T. Kanade. Uncertainty models in quasiconvex optimization for geometric reconstruction. In *Conf. Computer Vision and Pattern Recognition*, pages 1199 – 1205, New York City, USA, 2006.
22. J. H. Kim, R. Hartley, J. M. Frahm, and M. Pollefeys. Visual odometry for non-overlapping views using second-order cone programming. In *Asian Conf. Computer Vision*, November 2007.
23. J. J. Koenderink and A. J. van Doorn. Affine structure from motion. *J. Opt. Soc. America*, 8(2):377–385, 1991.
24. P. Kumar, P. H. S. Torr, and A. Zisserman. Solving markov random fields using second order cone programming relaxations. In *Conf. Computer Vision and Pattern Recognition*, pages 1045 – 1052, 2006.
25. H. Li. A practical algorithm for L-infinity triangulation with outliers. In *CVPR*, volume 1, pages 1–8. IEEE Computer Society, 2007.
26. H. Li and R. Hartley. Five-point motion estimation made easy. In *Int. Conf. Pattern Recognition*, pages 630–633, August 2006.
27. H. Li and R. Hartley. The 3D – 3D registration problem revisited. In *Int. Conf. Computer Vision*, Oct 2007.
28. H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.
29. F. Lu and R. Hartley. A fast optimal algorithm for L_2 triangulation. In *Asian Conf. Computer Vision*, November 2007.
30. Andrew Marvell. To his coy mistress. circa 1650.
31. D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.
32. D. Nistér, R. Hartley, and H. Stewénus. Using Galois theory to prove that structure from motion algorithms are optimal. In *Conf. Computer Vision and Pattern Recognition*, June 2007.
33. D. Nistér, F. Kahl, and H. Stewénus. Structure from motion with missing data is NP-hard. In *Int. Conf. Computer Vision*, Rio de Janeiro, Brazil, 2007.
34. C. Olsson, A. Eriksson, and F. Kahl. Efficient optimization of L_∞ -problems using pseudoconvexity. In *Int. Conf. Computer Vision*, Rio de Janeiro, Brazil, 2007.
35. C. Olsson, F. Kahl, and M. Oskarsson. Optimal estimation of perspective camera pose. In *Int. Conf. Pattern Recognition*, volume II, pages 5–8, Hong Kong, China, 2006.
36. C. Olsson, F. Kahl, and M. Oskarsson. The registration problem revisited: Optimal solutions from points, lines and planes. In *Conf. Computer Vision and Pattern Recognition*, volume I, pages 1206–1213, New York City, USA, 2006.

37. C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, 1982.
38. W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
39. M. Salzmann, R. Hartley, and P. Fua. Convex optimization for deformable surface 3D tracking. In *Int. Conf. Computer Vision*, Oct 2007.
40. Y. Seo and R. Hartley. A fast method to minimize L_∞ error norm for geometric vision problems. In *Int. Conf. Computer Vision*, Oct 2007.
41. Y. Seo and R. Hartley. Sequential L_∞ norm minimization for triangulation. In *Asian Conf. Computer Vision*, November 2007.
42. K. Sim and R. Hartley. Recovering camera motion using the L_∞ -norm. In *Conf. Computer Vision and Pattern Recognition*, pages 1230–1237, New York City, USA, 2006.
43. K. Sim and R. Hartley. Removing outliers using the L_∞ -norm. In *Conf. Computer Vision and Pattern Recognition*, pages 485–492, New York City, USA, 2006.
44. H. Stewénius, F. Schaffalitzky, and D. Nistér. How hard is three-view triangulation really? In *Int. Conf. Computer Vision*, pages 686–693, Beijing, China, 2005.
45. J. F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.
46. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *Int. Journal Computer Vision*, 9(2):137–154, November 1992.
47. W. Triggs, P. F. McLauchlan, R. I. Hartley, and A. Fitzgibbon. Bundle adjustment for structure from motion. In *Vision Algorithms: Theory and Practice*, pages 298–372. Springer-Verlag, 2000.
48. X. Zhang. Pose estimation using L_∞ . In *Image and Vision Computing New Zealand*, 2005.