Verifying Global Minima for L_2 Minimization Problems

Richard Hartley
Australian National University and NICTA

Yongduek Seo
Sogang University, Korea

Abstract

We consider the least-squares (L2) triangulation problem and structure-and-motion with known rotatation, or known plane. Although optimal algorithms have been given for these algorithms under an L-infinity cost function, finding optimal least-squares (L2) solutions to these problems is difficult, since the cost functions are not convex, and in the worst case can have multiple minima. Iterative methods can usually be used to find a good solution, but this may be a local minimum. This paper provides a method for verifying whether a local-minimum solution is globally optimal, by providing a simple and rapid test involving the Hessian of the cost function. In tests of a data set involving 277,000 independent triangulation problems, it is shown that the test verifies the global optimality of an iterative solution in over 99.9% *of the cases.*

1. Introduction

There has been much research into methods for solving the simplest of geometric Vision problems, the *triangulation problem*. However, no ideal method has been given to guarantee an optimal least-squares solution. In fact, it may not be possible to find an entirely acceptable algorithm that guarantees an optimal solution. This paper, however, tries a totally different approach, by giving a procedure for verifying whether an obtained solution actually is the global optimum. Usually, in fact, it is.

Although the condition is a sufficient but not necessary condition for the solution to be a global optimum, it works in almost all cases. In the rare cases where the condition fails it is usually because the point has large noise, in which case in a large-scale reconstruction problem, the best option is just to remove the point from consideration. Alternatively it may be possible to apply one of the recent (considerably more time-consuming) optimal algorithms ([7, 1]) for solving the problem.

Although known methods do not guarantee a globally optimal solution, nevertheless, simple methods based on initialization, followed by iterative refinement usually work very well. They depend on the initial solution being within the basin of attraction of the optimal solution. However, until now there has been no way of verifying this requirement. In this paper, we address this problem and give a fast and very satisfactory method of verifying that a given local minimum is the global minimum. By experiments on a very large set of triangulation problem, the test is seen to have a 99.9% success rate and runs in around $1.6ms$ on a desktop computer of medium power. The fact that the success rate is not 100% is probably not important. In many applications, one can simply delete questionable points.

This paper introduces a new technique for proving convexity. It seems hopeful that this technique can be used on other problems. We include an analysis that shows that the same analysis applies to a much more large-scale problem, namely structure and motion knowing rotations. It has potential to be applied in any quasi-convex optimization problem, since the location of the global minimum can be constrained to a convex set, usually small. Such problems have been shown to abound in Vision ([5]).

How hard is the triangulation problem, really? It was shown in [3] that the least-squares two-view triangulation problem is solvable in closed form. However, up to three local minima may exist. Much more recently, it has been shown ([9]) that the solution for three views involves the solution of a polynomial of degree 47, and higher degree polynomials are required with more views. The degree of the polynomial involved in the solution translates into numbers of possible local minima. It is certainly possible to find explicit examples with multiple local minima. This suggests that the problem is difficult.

On the other hand, it was shown in [2] that a single local (and hence global) minimum occurs if an L_∞ (minimax) solution is sought, instead of a the least-squares solution. However, the least-squares problem remains the problem of primary interest. The L_∞ solution is useful as an initialization procedure for the least squares, but it does not guarantee an optimal solution. In a real data test it seems that common algorithms get the right answer most of the time. So, perhaps the problem is not so hard after all – if only one knows whether one has the right solution. That problem is effectively solved in this paper.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

2. Bounding the Location of the L_2 Global Minimum

This paper is about verification that a solution obtained to a geometric vision problem is the global minimum. We assume that a candidate solution already exists, found by some known technique. We describe the method in terms of the triangulation problem, though the method is also applicable to other problems.

The desired solution to the triangulation problem is a point $\mathbf{X}_{\rm opt}$ that represents a global minimum of the cost function $C(\mathbf{X}) = \sum_{i=1}^{N} f_i(\mathbf{X})$, where f_i represents a squared residual error measurement in the i-th of N images. Thus $f_i(\mathbf{X}) = \|\mathbf{x}_i - P_i(\mathbf{X})\|^2$ where $P_i(\mathbf{X})$ represents the projection of the point $\mathbf{X}$ into the i-th image, $\mathbf{x}_i$ is the measured projection point, and $\|\cdot\|$ represents Euclidean norm, a distance measure in the image.

Now, suppose that $\mathbf{X}$ is a proposed solution to this problem with residual given by $C(\mathbf{X}) = \sum_{i=1}^{N} f_i(\mathbf{X}) = \epsilon^2$. This measurement gives a constraint on the position of the optimal solution, which must satisfy

$$C(\mathbf{X}_{\rm opt}) = \sum_{i=1}^{N} f_i(\mathbf{X}_{\rm opt}) \leq \epsilon^2$$

Since the sum of terms $f_i(\mathbf{X}_{\rm opt})$ must be less than ϵ^2, so must each individual term. Thus, for all i, $f_i(\mathbf{X}_{\rm opt}) \leq \epsilon^2$, or $\|\mathbf{x}_i - P_i(\mathbf{X}_{\rm opt})\| \leq \epsilon$. The set of points $\mathbf{X}$ in $\mathbb{R}^3$ satisfying this condition constitutes a cone in $\mathbb{R}^3$, as observed in [2]. Since this condition must be satisfied by each of the projections P_i, it follows that $\mathbf{X}_{\rm opt}$ must lie in the intersection of all the cones. This is a convex region of $\mathbb{R}^3$, since each cone is a convex set. The convex region could be computed using Second Order Cone Programming, but instead we use Linear Programming, which gives a slightly larger region, but is simpler.

The overall strategy. Here we summarize the general approach of the paper. The existence of an approximate solution $\mathbf{X}$ with error residual ϵ^2 constrains the optimal solution $\mathbf{X}_{\rm opt}$ to lie in a convex region about $\mathbf{X}$. Our strategy is to provide tests that allow us to prove that the cost function must be convex on this convex region. If this is so, then finding the optimal solution $\mathbf{X}_{\rm opt}$ may be carried out using standard convex optimization methods. More significantly, if $\mathbf{X}$ already is a local minimum of the cost function, found by any geometric optimization technique, then it must be a global minimum.

2.1. The camera model.

We consider initially a calibrated camera model, with different cameras having potentially different focal lengths. For calibrated cameras, we prefer to think in terms of image points being represented by points on a sphere, that is unit vectors, rather than an image plane. Later in this paper we will also consider standard projective cameras with an image plane.

Image error is represented by the angular difference between a measured direction $\mathbf{w}$ and the direction vector from the camera centre to the point $\mathbf{X}$. Since resolution of different cameras may vary, we allow for weighting the error by a factor k, related to (perhaps identical with) focal length, and potentially different for each camera. The total cost function is the sum of squares of the weighted angle errors.

Initially, for simplification, our cost function will instead be the square of the *tangent* of the angular error, rather than the squared angle itself. Since errors are usually quite small, the difference is insignificant, and the analysis is simpler. In section 5, however, we will extend the result to minimization of squared angle error.

2.2. The Hessian of the Cost Function

A function is convex on a convex region if and only if its Hessian is positive semidefinite. Hence, we are led in this section to consider the Hessian of the cost function.

Consider a vector $\mathbf{w}$ pointing from the origin in the direction of the Z-axis. Now, let $\mathbf{X} = (\mathbf{x}, \mathbf{y}, \mathbf{z})$ be a point lying close to the positive Z axis, such that the vector from the origin to $\mathbf{X}$ makes an angle ϕ from the vector $\mathbf{w}$. Consider the *error function* given by

$$f(\mathrm{X}, \mathrm{Y}, \mathrm{Z}) = k^2 \tan^2 \phi = k^2(\mathrm{X}^2 + \mathrm{Y}^2)/\mathrm{Z}^2 \ . \qquad (1)$$

This represents the squared projection error of the point $\mathbf{X}$ with respect to the "measured direction", $\mathbf{w}$.

The Hessian matrix of this function with respect to X, Y and Z is easily computed to be

$$\mathtt{H} = \frac{2k^2}{\mathrm{Z}^2} \begin{bmatrix} 1 & 0 & -2\mathrm{X}/\mathrm{Z} \\ 0 & 1 & -2\mathrm{Y}/\mathrm{Z} \\ -2\mathrm{X}/\mathrm{Z} & -2\mathrm{Y}/\mathrm{Z} & 3(\mathrm{X}^2 + \mathrm{Y}^2)/\mathrm{Z}^2 \end{bmatrix}$$

Since the function is circularly symmetric about the vector $\mathbf{w}$, so is the Hessian, so we may simplify the Hessian by evaluating it at a point with zero Y-coordinate, giving

$$\mathtt{H} = \frac{2k^2}{\mathrm{Z}^2} \begin{bmatrix} 1 & 0 & -2\mathrm{X}/\mathrm{Z} \\ 0 & 1 & 0 \\ -2\mathrm{X}/\mathrm{Z} & 0 & 3\mathrm{X}^2/\mathrm{Z}^2 \end{bmatrix} \qquad (2)$$

and substituting τ for X/Z (this is actually the tangent of the angle between the axis and point $\mathbf{X}$), and writing d instead of Z/k, to represent the *depth* of the point along the Z-axis scaled by k, we get

$$\mathtt{H} = \frac{2}{d^2} \begin{bmatrix} 1 & 0 & -2\tau \\ 0 & 1 & 0 \\ -2\tau & 0 & 3\tau^2 \end{bmatrix} \ . \qquad (3)$$

It may be verified that this matrix has two positive eigenvalues and one negative eigenvalue, meaning that the function f is neither concave nor convex at any point.

Now this is not a very convenient value for the Hessian since it is heavily dependent on the particular point $\mathbf{X}$ being considered. Instead, we consider a different matrix, which we would like to use instead, given by

$$\mathtt{H}' = \frac{2}{d^2}\begin{bmatrix} 1/3 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & -3\tau^2 \end{bmatrix} . \quad (4)$$

This matrix is seemingly pulled out of a hat – the reasoning that led to its choice was intricate, but its properties are simple enough. The key property of this matrix is the following observation:

$$\mathtt{H} - \mathtt{H}' = \frac{2}{d^2}\begin{bmatrix} 2/3 & 0 & -2\tau \\ 0 & 2/3 & 0 \\ -2\tau & 0 & 6\tau^2 \end{bmatrix}$$

is positive semi-definite. Indeed, it is easily seen that $\mathtt{H} - \mathtt{H}'$ has eigenvalues 0, $2/3$ and $2(1 + 9\tau^2)/3$, all non-negative. We write $\mathtt{H} - \mathtt{H}' \succeq 0$, or $\mathtt{H} \succeq \mathtt{H}'$. The matrix $\mathtt{H}'$ is more easily handled than $\mathtt{H}$, since it is dependent on the point $\mathbf{X}$ only through the values of d and τ, both of which can be bounded, as we shall see.

The eigenvalues of $\mathtt{H}'$ are obviously the diagonal entries. We see that $\mathtt{H}'$ has two positive and one negative eigenvalue. The eigenvector corresponding to the negative eigenvalue is directed along the $\mathtt{Z}$ axis, and the other two eigenvalues are in the plane perpendicular to the $\mathtt{Z}$ axis. However, since the two corresponding eigenvalues are equal, the eigenvectors may be taken as any two orthogonal vectors perpendicular to the $\mathtt{Z}$-axis.

Now, we may write $\mathtt{H}'$ as

$$\mathtt{H}' = (2/3d^2)\ \left(\mathtt{I} - (1 + 9\tau^2)\mathrm{diag}(0, 0, 1)\right) .$$

We assume that the point $\mathbf{X}$ lies in a cone with angle $\arctan\tau_{\max}$. Thus, $\tau \le \tau_{\max}$, and we see that

$$\begin{aligned} \mathtt{H} &\succeq \mathtt{H}' = (2/3d^2)\left(\mathtt{I} - (1 + 9\tau^2)\mathrm{diag}(0, 0, 1)\right) \\ &\succeq \mathtt{H}'' = (2/3d^2)\left(\mathtt{I} - (1 + 9\tau_{\max}^2)\mathrm{diag}(0, 0, 1)\right) . \end{aligned}$$

To verify the second line, simply observe that $\mathtt{H}' - \mathtt{H}''$ is positive definite if $\tau < \tau_{\max}$. Note that this matrix $\mathtt{H}''$ is a lower bound (in the semi-definite partial ordering) for the Hessian of f at any point $\mathbf{X}$ lying in the cone. It depends only on the depth d of the point from the vertex of the cone.

The computations above showed that $\mathtt{H} \succeq \mathtt{H}' = (2/d^2)\mathrm{diag}(1/3, 1/3, -3\phi^2)$ when $\mathtt{H}$ is evaluated at a point with $\mathtt{Y} = 0$. However, the error function is invariant under rotation about the $\mathtt{Z}$-axis. It follows from this condition that ${\mathtt{R}_\mathtt{Z}}^\top\mathtt{H}\mathtt{R}_\mathtt{Z} \succeq {\mathtt{R}_\mathtt{Z}}^\top\mathtt{H}'\mathtt{R}_\mathtt{Z} = \mathtt{H}'$ where $\mathtt{R}_\mathtt{Z}$ is a rotation about the $\mathtt{Z}$-axis. This shows that the inequality holds at all points, since ${\mathtt{R}_\mathtt{Z}}^\top\mathtt{H}\mathtt{R}_\mathtt{Z}$ is the Hessian at an arbitrary other point.

We now consider a cone with axis represented by a unit vector $\mathbf{w}$, and a point $\mathbf{X}$ lying at depth d from the vertex. Let $f(\mathbf{X}) = \tau^2$, where $\tau < \tau_{\max}$ is the tangent of the angle between the axis and the vector from the vertex to $\mathbf{X}$, and let $\mathtt{H}$ be the Hessian of f. Then it follows easily that

$$\mathtt{H} \succeq (2/3d^2)\left(\mathtt{I} - (1 + 9\tau_{\max}^2)\mathbf{w}\mathbf{w}^\top)\right) . \quad (5)$$

3. L_2 cost function from several measurements

Now, we consider a point $\mathbf{X}$, subject to several measurements, represented by vectors $\mathbf{w}_i$. We do not care where the vertex of the cone (corresponding to the camera centre) is located, but only that the depth of the point $\mathbf{X}$ in the i-th cone is d_i. We suppose that the point $\mathbf{X}$ is situated in the intersection of cones with angle $\arctan\tau_{\max}$. Let f_i be τ_i^2 where $\arctan\tau_i$ is the angle of $\mathbf{X}$ from the axis of the i-th cone. The L_2 error associated with the point $\mathbf{X}$ is given by $f(\mathbf{X}) = \sum_i f_i(\mathbf{X})$ and the Hessian of f is $\mathtt{H} = \sum_i \mathtt{H}_i$. Now applying the inequality (5) to each $\mathtt{H}_i$, we get

$$\mathtt{H} = \sum_i \mathtt{H}_i \succeq 2/3\sum_i (1/d_i^2)\left(\mathtt{I} - (1 + 9\tau_{\max}^2)\mathbf{w}_i{\mathbf{w}_i}^\top)\right) .$$

Writing $N = \sum_i 1/d_i^2$ and defining a matrix $\mathtt{A}$ by $\mathtt{A} = \sum_i \mathbf{w}_i{\mathbf{w}_i}^\top/d_i^2$ we see that

$$\mathtt{H} \succeq 2/3\left(N\mathtt{I} - (1 + 9\tau_{\max}^2)\mathtt{A}\right) .$$

It is our purpose to show (under certain conditions) that the function f is convex inside the region of interest. To do this, it is sufficient to show that $\mathtt{H}$ is positive definite, and in light of the inequality above, this will hold if $N\mathtt{I} - (1 + 9\tau_{\max}^2)\mathtt{A}$ is positive-definite. A matrix $\mathtt{M}$ is positive-definite if $\mathbf{x}^\top\mathtt{M}\mathbf{x} > 0$ for any vector $\mathbf{x}$. We may assume that $\mathbf{x}$ is a unit vector, and then applying this, we see that the required sufficient condition is that

$$N - (1 + 9\tau_{\max}^2)\mathbf{x}^\top\mathtt{A}\mathbf{x} > 0$$

for any unit vector $\mathbf{x}$. In other words,

$$N > (1 + 9\tau_{\max}^2)\max_{\mathbf{x}}\ \mathbf{x}^\top\mathtt{A}\mathbf{x}$$

where the maximum is taken over all unit vectors $\mathbf{x}$. The quantity $\max_{\mathbf{x}}\ \mathbf{x}^\top\mathtt{A}\mathbf{x}$ is the *matrix 2-norm*, written $\|\mathtt{A}\|$, equal to the largest eigenvalue (or singular value) of $\mathtt{A}$. The condition for the function f to be convex at the point $\mathbf{X}$ is then that

$$(1 + 9\tau_{\max}^2)\|\mathtt{A}(\mathbf{X})\| < N(\mathbf{X}) . \quad (6)$$

Here, by writing $\mathtt{A}(\mathbf{X})$ and $N(\mathbf{X})$, we have explicity indicated the dependency of the matrix $\mathtt{A}$ and the value N on the particular point $\mathbf{X}$ considered, lying inside the region of interest.

Observe that the inequality $\|\mathtt{A}(\mathbf{X})\| \leq N(\mathbf{X})$ always holds. The convexity condition (6) represents only a slightly more stringent condition. If $\tau_{\max}$ is small, which will be true in all cases of real interest, then the condition is only very slightly stronger. In this case we may expect the convexity condition to be satisfied.

4. Conditions for Convexity

Let D be the convex domain formed as the intersection of cones with angle equal to $\arctan \tau_{\max}$. We are interested in showing that the L_2 error function f is convex on the domain D.

In (6), we note that both sides of the inequality are positive quantities. We will be able to show that the function f is convex on the domain D if the maximum value of $\|\mathtt{A}(\mathbf{X})\|$ on the domain D is less than the minimum value of $N(\mathbf{X})/(1+9\tau_{\max}^2)$. This allows us to derive conditions for convexity. However, before doing this, we rearrange the inequality as follows:

$$9\tau_{\max}^2 \|\mathtt{A}(\mathbf{X})\| < N(\mathbf{X}) - \|\mathtt{A}(\mathbf{X})\| \ . \tag{7}$$

This is a sufficient condition for the function to be convex at point $\mathbf{X}$. An important point is that it is still the case that both sides of this inequality are positive. The point now is that the left hand side of this equation is a small quantity (assuming $\tau_{\max}$ is small). Note also that the left and right hand sides of this inequality correspond to the contributions of the negative and positive eigenvalues of the individual Hessian matrices $\mathtt{H}_i'$ given in (4).

We now rewrite the definitions of $N(\mathbf{X})$ and the matrix $\mathtt{A}(\mathbf{X})$.

$$N(\mathbf{X}) = \sum_i g_i^2(\mathbf{X}) \ ; \ \mathtt{A}(\mathbf{X}) = \sum_i g_i(\mathbf{X})^2 \mathbf{w}_i \mathbf{w}_i^\top \tag{8}$$

where we have written $g_i(\mathbf{X})$ instead of $1/d_i$. We will call the values g_i the *weight coefficients*. For simplicity of notation, we will generally write g_i instead of $g_i(\mathbf{X})$, but it should be remembered that the weight coefficients g_i are different for each different point $\mathbf{X}$. Writing g_i instead of $1/d_i$ simplifies the notation very slightly, but this is done for better reasons than notational convenience, as we will see later on.

We look a little more carefully at the right-hand side of (7). We write $\underline{\lambda}(\cdot)$ to be the *smallest* eigenvalues of a matrix. Since $\|\mathtt{A}(\mathbf{X})\|$ is the largest eigenvalue of $\mathtt{A}(\mathbf{X})$, we can write $-\|\mathtt{A}(\mathbf{X})\| = \underline{\lambda}(-\mathtt{A}(\mathbf{X}))$, and since $N(\mathbf{X})$ is a scalar,

$$\begin{aligned} N(\mathbf{X}) - \|\mathtt{A}(\mathbf{X})\| &= \underline{\lambda}(N(\mathbf{X})\mathtt{I} - \mathtt{A}(\mathbf{X})) \\ &= \underline{\lambda}\left(\sum_i g_i^2 (\mathtt{I} - \mathbf{w}_i \mathbf{w}_i^\top)\right) \\ &= \underline{\lambda}\left(\sum_i g_i^2 (\mathbf{u}_i \mathbf{u}_i^\top + \mathbf{v}_i \mathbf{v}_i^\top)\right) \\ &= \underline{\lambda}(\mathtt{A}'(\mathbf{X})) \end{aligned}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are two orthogonal unit vectors perpendicular to $\mathbf{w}_i$. The second-last line holds because $\mathtt{I} = \mathbf{u}_i \mathbf{u}_i^\top + \mathbf{v}_i \mathbf{v}_i^\top + \mathbf{w}_i \mathbf{w}_i^\top$. The matrix $\mathtt{A}'$ thus defined is positive-semi-definite, since it is the sum of positive semi-definite matrices.

A sufficient condition for convexity of the function f on the domain D is then that

$$9\tau_{\max}^2 \max_{\mathbf{X}\in D} \|\mathtt{A}(\mathbf{X})\| < \min_{\mathbf{X}\in D} \underline{\lambda}(\mathtt{A}'(\mathbf{X})) \ . \tag{9}$$

Now, suppose that for each index i, we can bound g_i between two values $g_{i,\min}$ and $g_{i,\max}$, namely $g_{i,\min} \leq g_i(\mathbf{X}) \leq g_{i,\max}$. Correspondingly, we may bound the values of $\|\mathtt{A}(\mathbf{X})\|$ and $\underline{\lambda}(\mathtt{A}'(\mathbf{X}))$ expressed in terms of these upper and lower bounds. This gives

$$\begin{aligned} \max_{\mathbf{X}\in D} \|\mathtt{A}(\mathbf{X})\| &\leq \|\mathtt{A}_{\max}\| \\ \underline{\lambda}(\mathtt{A}'_{\min}) &\leq \min_{\mathbf{X}\in D} \underline{\lambda}(\mathtt{A}'(\mathbf{X})) \end{aligned} \tag{10}$$

where $\mathtt{A}'_{\min}$ and $\mathtt{A}_{\max}$ are matrices defined just like $\mathtt{A}'(\mathbf{X})$ and $\mathtt{A}(\mathbf{X})$ but using the weight coefficients $g_{i,\min}$ and $g_{i,\max}$ instead of $g_i(\mathbf{X})$. Now, consider the condition

$$9\tau_{\max}^2 \ \|\mathtt{A}_{\max}\| \ \leq \ \underline{\lambda}(\mathtt{A}'_{\min}) \ . \tag{11}$$

Combining this inequality with (10) yields (9). Since (9) is a sufficient condition for convexity of the L_2 error function on D, so is (11).

Finally, since $\underline{\lambda}(\mathtt{A}'_{\min}) = N_{\min} - \|\mathtt{A}_{\min}\|$ where $N_{\min} = \sum_i g_{i,\min}^2$, we may replace (11) by

$$9\tau_{\max}^2 \|\mathtt{A}_{\max}\| + \|\mathtt{A}_{\min}\| \ \leq \ N_{\min} \ , \tag{12}$$

which is an alternative form of the convexity condition (11).

Bounded max/min ratio. We consider a special case where there exists a value $\gamma > 1$ such that $\gamma g_{i,\min} \geq g_{i,\max}$ for all i. In this case, we may assert that $\|\mathtt{A}_{\max}\| \leq \gamma^2 \|\mathtt{A}_{\min}\|$. Thus, consider the condition

$$(1 + 9\tau_{\max}^2 \gamma^2) \ \|\mathtt{A}_{\min}\| \ \leq \ N_{\min} \ . \tag{13}$$

Since (12) may be deduced from (13), it follows that (13) is a sufficient condition for convexity.

To summarise this section, the equations (12) and (13) are alternative sufficient conditions for convexity of the error function. The second condition is slightly weaker.

4.1. How to use these conditions.

We now give the details of how to prove convexity. Consider a set of cameras with centres $\mathbf{C}_i$, and let $\mathbf{w}_i$ be a direction vectors representing the measured direction of an observed point $\mathbf{X}$ from $\mathbf{C}_i$. Let $\mathbf{u}_i$ and $\mathbf{v}_i$ be two unit vectors orthogonal to $\mathbf{w}_i$ constituting (along with $\mathbf{w}_i$) an orthogonal coordinate frame. These may be the three rows of the rotation matrix for the camera, oriented with principal direction pointing in the direction $\mathbf{w}_i$ ([4]).

Step 1. Bounding the region. Let $\mathbf{X}$ be a 3D point constituting a potential solution to the triangulation problem. The cost of this point is the value of the cost function

$$\sum_i f_i(\mathbf{X}) = \sum_i \left(\frac{\mathbf{u}_i{}^\top(\mathbf{X}-\mathbf{C}_i)}{\mathbf{w}_i{}^\top(\mathbf{X}-\mathbf{C}_i)}\right)^2 + \left(\frac{\mathbf{v}_i{}^\top(\mathbf{X}-\mathbf{C}_i)}{\mathbf{w}_i{}^\top(\mathbf{X}-\mathbf{C}_i)}\right)^2 .$$

Let the value of this cost for the given point $\mathbf{X}$ be ϵ^2.

We may then define a region of space in which the optimal point $\mathbf{X}_{\text{opt}}$ lies according to the inequalities.

$$-\epsilon \;\le\; \frac{\mathbf{u}_i{}^\top(\mathbf{X}_{\text{opt}}-\mathbf{C}_i)}{\mathbf{w}_i{}^\top(\mathbf{X}_{\text{opt}}-\mathbf{C}_i)} \;\le\; \epsilon \qquad (14)$$

and similar inequalities involving $\mathbf{v}_i$ instead of $\mathbf{u}_i$. Since $\mathbf{w}_i{}^\top(\mathbf{X}_{\text{opt}}-\mathbf{C}_i) > 0$ (the cheirality constraint that the point must lie in the direction it is observed), we can multiply out by $\mathbf{w}_i{}^\top(\mathbf{X}_{\text{opt}}-\mathbf{C}_i)$ to obtain a total of four linear inequalities in the positions of the point $\mathbf{X}_{\text{opt}}$, constraining it to a polyhedral region of space, D.

Step 2. Finding depth bounds. The next step is to find minimum and maximum of d_i on the region D. Since d_i is defined to be $\mathbf{w}_i{}^\top(\mathbf{X}-\mathbf{C}_i)$, determining its minimum and maximum over the polyhedral region D are simply a pair of linear programming problems.

Step 3. Performing the test. We can now compute the matrices $\mathtt{A}_{\text{min}}$, $\mathtt{A}_{\text{max}}$ and the scalar N_{min} defined by (8) in terms of the values $g_{\text{min}} = 1/d_{\text{max}}$ and $g_{\text{max}} = 1/d_{\text{min}}$. The 2-norms of $\mathtt{A}_{\text{min}}$ and $\mathtt{A}_{\text{max}}$ are now computed by finding their maximum eigenvalues, and the inequality (12) is tested. If the inequality is true, then the cost function is convex on the region D. If the initial estimate $\mathbf{X}$ is a local minimum, then it is also a global minimum.

4.2. Infinite region

If the domain D is infinite, or very extended, then the conditions (12) and (13) are not very useful in the case where $g_i(\mathbf{X}) = 1/d_i(\mathbf{X})$, since in this case $g_{i,\text{max}}$ becomes zero. However, it is possible to choose g_i differently.

In particular, suppose we choose $g_i(\mathbf{X}) = \alpha(\mathbf{X})/d_i(\mathbf{X})$, where $\alpha(\mathbf{X})$ is any non-negative function of the point $\mathbf{X}$ (independent of i however), then the theory will work as before. In particular, the basic sufficient condition for convexity is (9):

$$9\tau^2_{\text{max}} \max_{\mathbf{X}\in D} \|\mathtt{A}(\mathbf{X})\| < \min_{\mathbf{X}\in D} \underline{\lambda}(\mathtt{A}'(\mathbf{X}))$$

which is a sufficient condition for convexity, provided that the matrices $\mathtt{A}(\mathbf{X})$ and $\mathtt{A}'(\mathbf{X})$ are defined in terms of weight coefficients $g_i = 1/d_i$. Thus, $\mathtt{A}(\mathbf{X}) = \sum_i \mathbf{w}_i\mathbf{w}_i{}^\top/d_i(\mathbf{X})^2$. Multiplying by $\alpha(\mathbf{X})^2$, we get

$$\begin{aligned} \alpha(\mathbf{X})^2\,\mathtt{A} &= \sum_i (\alpha(\mathbf{X})/d_i(\mathbf{X}))^2 \mathbf{w}_i\mathbf{w}_i{}^\top \\ &= \sum_i g_i(\mathbf{X})^2 \mathbf{w}_i\mathbf{w}_i{}^\top \end{aligned}$$

where we have defined $g_i = \alpha(\mathbf{X})/d_i(\mathbf{X})$. The effect is to multiply $\mathtt{A}$, and hence its matrix-norm $\|\mathtt{A}\|$ by $\alpha(\mathbf{X})^2$. The same is true of the minimum eigenvalue, $\underline{\lambda}(\mathtt{A}'(\mathbf{X}))$. Thus, using weights $g_i = \alpha(\mathbf{X})/d_i(\mathbf{X})$ has the effect of multiplying both sides of (9) by the non-negative value $\alpha(\mathbf{X})^2$, which results in an equivalent inequality.

The conclusion is that (9) defined in terms of weights $g_i = \alpha(\mathbf{X})/d_i$ is a sufficient condition for complexity of f on the domain D. From this, all the other sufficiency conditions follow in the same way as before. This is the reason that the analysis in section 3 was carried out with arbitrary weights $g_i(\mathbf{X})$.

A suitable choice of the weight coefficients $\alpha(\mathbf{X})$ is some linear function of the weights d_i. Thus, let $\alpha(\mathbf{X}) = \sum_i \alpha_i d_i(\mathbf{X})$, where the α_i are fixed non-negative constants. The reason for this choice is that it is then easy to find the minimum and maximum of $g_i(\mathbf{X}) = \alpha(\mathbf{X})/d_i(\mathbf{X})$ on the convex domain D.

Possible reasonable choices of the weight function $\alpha(\mathbf{X})$ are

1. $\alpha(\mathbf{X}) = 1$. In this case $g_i = 1/d_i$. This will be suitable for a small domain D in which the ratio of maximum to minimum depth is small. However, if the domain D stretches towards infinity, then it is not suitable.

2. $\alpha(\mathbf{X}) = (1/n)\sum_{i=1}^n d_i(\mathbf{X})$. Thus w measures the average depth, and for each i, the coefficient $g_i = \alpha(\mathbf{X})/d_i(\mathbf{X})$ is the ratio of the average depth of a point $\mathbf{X}$ to the i-th depth $d_i(\mathbf{X})$. Even for domains D reaching to infinity, this ratio will remain within reasonable maximum and minimum bounds, since the average depth $\alpha(\mathbf{X})$ increases to infinity at the same rate as any specific depth d_i.

We now may summarize the previous discussion by stating the complete convexity theorem.

Theorem 4.1. *Let domain D be the intersection of a set of n cones with axes represented by unit vectors $\mathbf{w}_i$, and with* *angle bounded by* $\arctan \tau_{\max}$. *For a point* $\mathbf{X} \in D$, *let* $d_i(\mathbf{X})$ *represent its depth from the vertex of the* i*-th cone. Define a function* $\alpha(\mathbf{X}) = \alpha_0 + \sum_{i=1}^{n} \alpha_i d_i(\mathbf{X})$ *and* $g_i(\mathbf{X}) = \alpha(\mathbf{X})/d_i(\mathbf{X})$. *Let* $g_{i,\max}$ *and* $g_{i,\min}$ *be the maximum and minimum values of* g_i *on* D. *Define matrices*

$$\mathtt{A}_{\max} = \sum_{i=1}^{n} g_{i,\max}^2 \mathbf{w}_i \mathbf{w}_i^\top \qquad \mathtt{A}_{\min} = \sum_{i=1}^{n} g_{i,\min}^2 \mathbf{w}_i \mathbf{w}_i^\top$$

and values

$$N_{\max} = \sum_{i=1}^{n} g_{i,\max}^2 \qquad N_{\min} = \sum_{i=1}^{n} g_{i,\min}^2$$

Then (12) and (13) are both sufficient conditions for the least-squares error function f *to be convex of* D. □

5. Minimizing squared angle

In the previous discussion, the error function used was the sum of squares of the *tangents* of the angles, as given by (1). This leads to a relatively simple result in terms of computing and bounding the Hessian. On the other hand, it would be more natural to wish to minimize the sum of squares of the error angles, and not their tangents. The difference is very small, but for exactness, we now derive a similar result for this error function.

In the following discussion, we give the outline of the argument. To verify the details, the reader may need to use a computer algebra system, such as Mathematica. Now, under the same conditions as before, we define the error function

$$f(\mathtt{X}, \mathtt{Y}, \mathtt{Z}) = \phi^2 = \arctan\left(\frac{\sqrt{\mathtt{X}^2 + \mathtt{Y}^2}}{\mathtt{Z}}\right)^2 \quad . \qquad (15)$$

We compute the Hessian at a point with $(\mathtt{X}, \mathtt{Y}, \mathtt{Z})$ with respect to the coordinates $\mathtt{X}$, $\mathtt{Y}$ and $\mathtt{Z}$. Subsequently, evaluating with $\mathtt{X} \geq 0$ and $\mathtt{Y} = 0$, and making the substitutions ϕ for $\arctan(\mathtt{X}/\mathtt{Z})$, radial distance r for $\sqrt{\mathtt{X}^2 + \mathtt{Z}^2}$ and $\tan\phi$ for $\mathtt{X}/\mathtt{Z}$, we arrive after some computation at the expression for the Hessian given in Fig 1. This matrix has eigenvalues $2\phi/(r^2 \tan(\phi))$ and $\left(1 \pm \sqrt{1 + 4\phi^2}\right)/r^2$, namely two positive and one negative eigenvalue.

Similarly as before[1], let $\mathtt{H}'$ be the matrix $(2/r^2)\mathrm{diag}(1/4, 1/4, -4\phi^2)$. We claim that $\mathtt{H} - \mathtt{H}'$ is positive semi-definite. Unfortunately, the proof is a little more intricate than before.

First, we observe that $(0, 1, 0)^\top$ is an eigenvector of $\mathtt{H} - \mathtt{H}'$, with eigenvalue $(2/r^2)(\phi/\tan(\phi) - 1/4)$, which is positive at least for $\phi < 1$. The other two eigenvalues are the eigenvalues of the reduced-size matrix obtained by eliminating the second row and column from $\mathtt{H} - \mathtt{H}'$. This matrix will have two positive eigenvalues, as long as its trace and determinant are both positive. Apart from the factor $1/r^2$ the trace is equal to $8\phi^2 + 3/2$ which is positive. The determinant is equal to

$$D(\phi) = -1 + (1 + 16\phi^2)(\cos(2\phi) - 2\phi \sin(2\phi)) \ .$$

This function is positive for $\phi < 0.3$ as may be shown by plotting the function $D(\phi)/\phi^2$ (see Fig 2). A more formal proof can be given by computing the series expansion of this function.

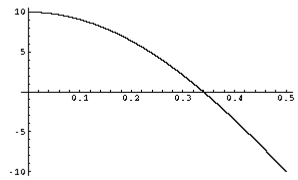

Figure 2. *Plot of the function* $D(\phi)/\phi^2$ *demonstrating that the product of the eigenvalues of* $\mathtt{H} - \mathtt{H}'$ *is positive for* $\phi < 0.3$.

The result of this computation is the following result.

Lemma 5.2. *If* $\mathtt{H}$ *is the Hessian of the error function (15) evaluated at a point with error less than angle* $\phi < 0.3$, *then*

$$\mathtt{H} \succeq (2/r^2)\mathrm{diag}(1/4, 1/4, -4\phi^2) \ .$$

From here on, conditions for convexity of the error function in the intersection of a set of cones with angle bounded by $\phi_{\max}$ procedes just the same as previously.

6. Projective Transform

If the direction of a point is virtually the same from all cameras (the point is at infinity for instance) the proposed method will not work, because there will be no cancellation of the negative eigenvalues. In this case, the situation can be saved by the application of a projective transformation. Effectively, we will reparametrize the domain of the error function, which can often turn a non-convex function into a convex one. If the view directions for several cameras are similar, this implies that the point $\mathbf{X}$ is near to infinity. We apply a projective transformation that maps this point to a point closer to the cameras, so that the viewing rays are no longer near parallel.

This method has been verified on simple examples, but can not be discussed further here.

[1]By analogy with section 2.2 one may be tempted to define $\mathtt{H}' = (2/r^2)\mathrm{diag}(1/3, 1/3, -3\phi^2)$ which gives a slightly better bound, but this does not work in this case.

$$
\mathtt{H} = \frac{1}{r^2} \begin{bmatrix} 1+\cos(2\phi) - 2\phi\sin(2\phi) & 0 & -2\phi\cos(2\phi) - \sin(2\phi) \\ 0 & 2\phi/\tan(\phi) & 0 \\ -2\phi\cos(2\phi) - \sin(2\phi) & 0 & 1-\cos(2\phi) + 2\phi\sin(2\phi) \end{bmatrix}
$$

Figure 1. *Hessian for cost function (15)*

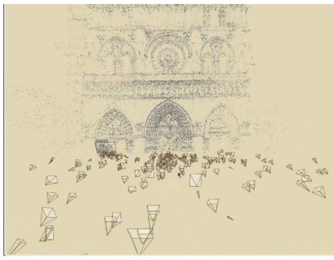

Figure 3. *Representation of the dataset used for triangulation experiments. Thanks to Noah Snavely for supplying this data.*

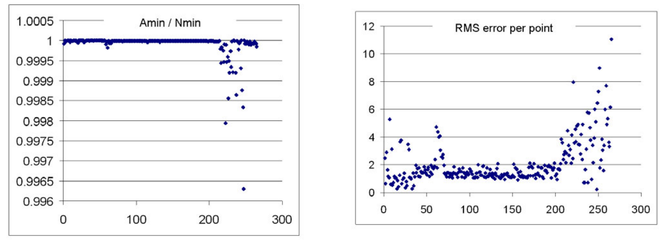

Figure 4. *Left: Plot of the ratio $\|A_{\min}\|/N_{\min}$ for the points that failed the primary convexity test. The ratio is close to 1.0 indicating that the viewing rays are nearly parallel. Right: Plot of RMS reprojection error per for the points that failed. Many points have quite high residual, implying a bad point match.*

7. Experiments

We tried the triangulation experiment out on virtually all points in the "Notre Dame" image set ([8]). A reconstruction of the point cloud and some of the camera positions from this set are shown in Fig 3. Some points were removed from this data set, since they were obviously bad matches, with residual projection errors up to 25 pixels. All points with projection errors less than 10 pixels were retained. This resulted in a total of $277,887$ points which were triangulated. The initial points $\mathbf{X}$ were computed using bundle adjustment in [8], and those were the points that we used. The results were as follows.

1. Our *primary test* is the condition (12) with $g_i(\mathbf{X}) = 1/d_i(\mathbf{X})$. Of the $277,887$ points, only 281 failed to pass the test – about 0.1%.

2. Test (13) is slightly weaker. It failed on 284 cases.

3. We also ran the test with the weight function $g_i(\mathbf{X}) = \sum_{k=1}^{n} d_k(\mathbf{X})/(nd_i(\mathbf{X}))$ discussed in section 4.2. This test is slower than the primary test. This is because it is more difficult to find the maximum and minimum of the function $g_i(\mathbf{X})$ on the region of interest. For this reason we apply it only when the other test fails. Of the 281 failure cases for the primary test, 130 were shown to be convex on the region of interest by this test. In other words 151 cases still failed. We were unable to prove convexity for these 151 out of $277,887$ points.

4. It is possible, perhaps likely that the method involving applying a projective transformation, described in section 6 would work for the majority of the remaining cases, since they all were for cameras with closely aligned principal rays. To verify this, we plot the ratio $\|A_{\min}\|/N_{\min}$ for the points that failed the primary test. This ratio can be equal to 1.0 only when all the direction rays are in the same direction. The result is shown in Fig 4 where it is seen that in almost all cases, this value is very close to 1.

5. The time taken for all $277,887$ convexity tests was 7 minutes and 23 seconds – that is less than 1.6 miliseconds per point – on a 3.2GHz Pentium. The program was written in C++ with optimation flags on.

8. Structure and Motion knowing Rotations

In this final section, we indicate that this method might be applicable to other much bigger problems. Although the theory is worked out below, implementation and testing of this method falls in the category of future work.

It was shown in [2] that the structure-and-motion problem with known rotations has a unique global minimum under the L_∞ norm, and in [5, 6] it was shown that this may be found using Second Order Cone Programming. The solution of this problem is not very much different from the simple triangulation problem considered so far, and it is not surprising that we may achieve similar results. This will be shown in this section.

The problem may be stated as follows. We are given a set of cameras with centres $\mathbf{C}_i$ and a set of 3D points represented by $\mathbf{X}_j$. The problem is to estimate all the camera centres $\mathbf{C}_i$ and the points $\mathbf{X}_j$ given only some measurements of the direction vector from the camera centres to the points. Ideally, we have

$$
\mathbf{v}_{ij} = \frac{\mathbf{X}_j - \mathbf{C}_i}{\|\mathbf{X}_j - \mathbf{C}_i\|} \quad .
$$

Thus, the measurements, $\mathbf{v}_{ij}$ are unit vectors. We do not assume that $\mathbf{v}_{ij}$ is known for all i and j.

Of course, in the presence of noise, the measurements are not exact, and so we seek an approximate solution that minimizes the sum-of-squares angle distances. For the present, instead we use the tangents of the angles instead of the actual angles. The sum-of-squares cost function is then

$$\sum_{i,j} \frac{\|\mathbf{v}_{ij} \times (\mathbf{X}_j - \mathbf{C}_i)\|^2}{\left(\mathbf{v}_{ij}{}^\top (\mathbf{X}_j - \mathbf{C}_i)\right)^2} \ . \tag{16}$$

The sum is over all i, j for which a measurement $\mathbf{v}_{ij}$ is available.

In order to avoid an obvious ambiguity of scale and translation, we may remove these so-called gauge-freedoms by constraining one camera centre $\mathbf{C}_0$ to be at the origin, and some specified point $\mathbf{X}_j$, visible from camera with centre $\mathbf{C}_0$, to satisfy $\mathbf{v}_{0j}{}^\top \mathbf{X}_j = 1$, effectively meaning that it at unit distance from the origin.

As before we wish to compute the Hessian of a single term of this sum. In doing so, we may assume that the vector $\mathbf{v}_{ij}$ is directed along the positive $\mathtt{Z}$ axis, and that the point $\mathbf{C}_i$ is located at the coordinate origin. Further, we assume that $\mathbf{X}_j$ is located in the plane $\mathtt{Y} = 0$. Under these circumstances, the Hessian with respect to the coordinates of $\mathbf{C}_i$ and $\mathbf{X}_j$ may be computed to be the 6×6 matrix

$$\mathtt{G} = \begin{bmatrix} \mathtt{H} & -\mathtt{H} \\ -\mathtt{H} & \mathtt{H} \end{bmatrix} \ ,$$

where $\mathtt{H}$ is the Hessian involved in the triangulation problem given in (3). It is easily seen that the eigenvectors of this matrix $\mathtt{G}$ are as follows. There are three eigenvectors $(1,0,0,1,0,0)^\top$, $(0,1,0,0,1,0)^\top$ and $(0,0,1,0,0,1)^\top$ with zero eigenvalues. In addition, there are three eigenvectors $(\mathbf{y}^\top, -\mathbf{y}^\top)^\top$ where $\mathbf{y}$ is one of the eigenvectors of the matrix $\mathtt{H}$. The corresponding eigenvalues are simply twice the corresponding eigenvalues of $\mathtt{H}$. Thus, the matrix $\mathtt{G}$ has three zero eigenvalues along with two positive eigenvalues (close to $2/\mathtt{Z}^2$) and one small negative eigenvalue.

As before, we may find a lower bound (in terms of the positive-definite partial ordering) for this matrix, given by

$$\mathtt{G}' = \begin{bmatrix} \mathtt{H}' & -\mathtt{H}' \\ -\mathtt{H}' & \mathtt{H}' \end{bmatrix} \ ,$$

where $\mathtt{H}'$ is the diagonal matrix given in (4). We see that $\mathtt{G} - \mathtt{G}'$ is positive-definite. This follows from the fact, previously verified that $\mathtt{H} - \mathtt{H}'$ is positive definite. In fact, it is clear that the non-zero eigenvalues of $\mathtt{G} - \mathtt{G}'$ are just double those of $\mathtt{H} - \mathtt{H}'$.

With this observation, the analysis of this problem follows virtually identically the analysis for the one-point triangulation problem.

9. Conclusions

The tests described here are extremely effective at verifying convexity, and hence global optimality of a local minimum. Test (13) seems to do almost as well as (12), but neither condition is difficult to compute, so one might as well use the stronger condition. Using a different weighting function as in section 4.2 can resolve some but not all remaining cases. Probably most remaining cases can be resolved by applying a projective transformation as in section 6, but this is untried.

The theory also applies to the problem of n-view structure and motion with known rotation. Experimental evaluation of its effectiveness on this problem belongs to further work. Space dictates that we omit the theoretical analysis from the present version of the paper.

Acknowledgement

The authors wish to thank the anonymous reviewers who pointed out a small but embarassing mistake in the original derivation of the condition (11).

This research was supported by NICTA, a research centre funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program. It was also supported by a Korea Science and Engineering Foundation (KOSEF) grant funded by the Korean government (MOST) (No. R01-2006-000-11374-0).

References

[1] S. Agarwal, M.K. Chandraker, F. Kahl, S. Belongie, and D.J. Kriegman. Practical global optimization for multiview geometry. In *European Conf. Computer Vision*, pages 592–605, Graz, Austria, 2006. 1

[2] R. Hartley and F. Schaffalitzky. L_∞ minimization in geometric reconstruction problems. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Washington DC*, pages I–504–509, June 2004. 1, 2, 7

[3] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, November 1997. 1

[4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 5

[5] F. Kahl. Multiple view geometry and the L_∞-norm. In *Proc. International Conference on Computer Vision*, pages 1002–1009, 2005. 1, 7

[6] Q Ke and T Kanade. Quasiconvex optimization for robust geometric reconstruction. In *Proc. International Conference on Computer Vision*, pages 986 – 993, 2005. 7

[7] Fangfang Lu and Richard Hartley. A fast optimal algorithm for L_2 triangulation. In *Proc. Asian Conference on Computer Vision*, November 2007. 1

[8] N. Snavely, S.M Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM SIGGRAPH*, 25(3):835–846, 2006. 7

[9] H. Stewenius, F. Schaffalitzky, and D. Nister. How hard is 3-view triangulation really. In *Proc. International Conference on Computer Vision*, pages 686 – 693, 2005. 1