Solving the Triangulation Problem using Cones

Felix Schwenninger
ERASMUS student from the Technical University of Vienna - TU WIEN
felix.schwenninger@gmx.net
Supervisor: Petter Strandmark

Description of the project

The basic topic we are going to investigate is the so called TRIANGULATION PROBLEM. Here one considers images \( \{v_i : i = 1,...,n\} \) of an unknown 3D point \( X \) and wants to compute \( X \) from these projections using the camera-matrices \( P_i \). The following figure\(^1\) visualises the case for two cameras.

![Triangulation Diagram](image)

In reality, the images \( v_i \) can not be measured exactly. Therefore, due to this uncertainty the calculation of \( X \) is not clear since the lines connecting the camera centers \( C_i \) and the respective image point \( v_i \) do not have to intersect. At this point it is common to regard the minimisation problem considering least squares distances in the image planes. Since this method suffers from local minima, one can use another straightforward approach which is the topic of this project: Recalling the situation illustrated above, one can consider small neighbourhoods (e.g. a disk) of the measured imagepoints \( v_i \) (in the image planes) and considers the cones naturally defined in the 3D space by the respective camera center and these ‘disks’ (circles). These cones, or precisely, the ‘extensions’ of the cones intersect if the neighbourhoods and hence the bases of the cones are sufficiently large. To get a solution for the triangulation problem we investigate the smallest possible neighbourhoods which give a non-empty intersection. Furthermore one can ‘weigh’ the distance from the ray going through the point \( v_i \) (and the camera center \( c_i \)) by a (Gaussian) distribution.

Mathematical setting

As indicated in the description, we are interested in the case where measurements in the image planes are not exact. However, let \( P_i \in \mathbb{R}^{3\times4} \) denote the camera which projects a 3D point \( X \in \mathbb{R}^4 \) (in homogenous coordinates, \( X(4) = 1 \)) onto the \( i \)-th image plane. Hence the resulting

\(^1\)from [http://en.wikipedia.org/wiki/Triangulation_(computer_vision)]
image point is computed by

\[ P_i \left( \frac{X}{1} \right) = \left( \frac{P_i(1,:)X}{P_i(2,:)X} \right) = \left( \frac{P_i(1,:)X/P_i(3,:)X}{P_i(2,:)X/P_i(3,:)X} \right), \tag{1} \]

where \( P_i(j,:) \) denotes the \( j \)-th row of \( P_i \) and the second equality is understood again in the projective sense in the image plane (provided that \( P_i(3,:)X \) is not zero). The question is now: 'How far away is this projection from our measurements?', or 'How can we choose the \( X \) which is nearest in the image plane?'. As explained in the previous part one can do geometrical considerations using the bundle of rays going through a neighbourhood of the measured point, and, doing this for all image planes, compute an intersection region/point of these cones. Even in the two camera case, an intersection region of two cones is not easy to imagine. Nevertheless, there are some special situations where this can easily be visualised, like in the case where for example the camera matrices are only rotated in a plane (this is a 2D problem then). Figure 1 shows the bird view of such a setting (in the figure the measured points are denoted by \( x_i \)).

The grey area in figure 1 shows the potential region for \( X \) which fulfills at least the requirement that their projections lie in a neighbourhood of the measured points \( v_i \). But instead of doing computations in the 3D space (calculating a representation of the cones and intersect them) it is obviously easier to 'stay' in the image planes, and work with the projections. But still, we have to specify what it means 'to look for the nearest solution'. In any case this should provide some kind of minimisation of the distance to the measurement data.

In general terminology we write the different image planes as \( V_i \ (V_i \cong \mathbb{R}^2) \) and denote the product space \( \prod_{i=1}^n V_i \) by \( V \). Furthermore, we write \( P : \text{dom}(P) \subset \mathbb{R}^3 \rightarrow V \) for the operator which computes the sequence of all images of a three dimensional input point (precisely: the input are elements from the projective space), i.e. \( PX = \{ P_iX \}_{i=1}^n \). The domain of \( P \), \( \text{dom}(P) \), is clearly not the whole space \( \mathbb{R}^3 \) since we can only compute the images for points lying in front of the camera (in this connection we mention the notion of 'Cheirality' later). Since the space \( V \) has dimension larger than 3, it is clear that \( P \) will never be surjective, i.e. \( \text{im}(P) \subsetneq V \) where \( \text{im}(P) \) denotes the range of \( P \). In these terms, a noised measurement of image points \( \{ v_i \} \in V \) is therefore in general not element of \( \text{im}(P) \).

So our task is to find, for a given \( v \in V \) a \( w \in \text{im}(P) \) such that \( \| w - v \| \) is minimal. In other

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure1.png}
\caption{intersection region}
\end{figure}
words, we have the minimisation problem

\[ \arg \min_{w \in \text{im}(P)} \|w - v\|, \]

which can equivalently be written as

\[ \arg \min_{X \in \text{dom}(P)} \|PX - v\| = \arg \min_{X \in \text{dom}(P)} \|(P_iX - v_i)_{i=1}^n\|. \quad (2) \]

We point out that for a 'non-pathologic' set of camera matrices, the operator \( P \) is injective (an example for a 'bad choice' is the set of two cameras with parallel image planes and camera centers lying on an orthogonal line to the image plane.).

Until now, nothing was said about the considered norm in \( V \). Since this is a (finite) product of vector spaces, the product topology is among others induced by the sum norm (This is not really a deep fact, since we have a finite dimensional space. Hence all norms are equivalent.), which is just the sum over all norms in each component. Equivalently (in topological sense) one may choose the maximum norm. But this decision leads to important differences. One main question is, if we can find a unique solution of problem (2). In optimisation this is strongly connected with convexity.

**Norm in each image**

We have to specify two norms; the one in the image planes \( V_i \) and the other in the product space \( V \). To discuss this, we consider the term \( \|P_iX - v_i\| \) which can be written as

\[ f_i(X) := \|P_iX - v_i\| = \left\| \frac{P_i(1,:)X - v_{i1}}{P_i(3,:)X - v_{i2}} \right\| \]

\[ = \left\| \begin{pmatrix} P_i(1,:)-v_{i1}\ P_i(2,:)-v_{i2} \end{pmatrix}^T X \right\|, \quad (3) \]

(as long as the term \( P_i(3,:)X \) is not zero). We point out, that this norm (in the line above) is the norm in one image \( V_i \), which should not be confused with the norm in (2). Probably the most natural choice in the \( V_i \) is the Euclidean norm, which can been seen as measuring the geometrical distance to the measured point \( v_i \). This assumes that the noise is 'uniformly distributed' around the exact point (in the sense that points with same euclidian distance to the measured point have same probability). The geometrical interpretation in the 3D space is, as described in the first section, a cone defined by the associated camera center and a circle with center \( v_i \) in the image plane. However, one could also use other norms, like for example the maximum norm, \( \| \cdot \|_\infty \). The neighbourhood will then be a square which edges are parallel to the coordinate axes in \( V_i \). Another example is the \( L_1 \)-norm which also gives us a square. Figures 2,3 and 4 visualize the neighbourhoods \( \|v_i\| = 1 \). Whereas the \( L_2 \)-norm had some kind of natural motivation from the problem, it is not clear how the other norms can be interpreted in reality unless one has any further information about the distribution of noise beside one considers it as a (not really great) 'approximation' for the circles. But the computation of these norms is faster than it is for the \( L_2 \) norm, since it is a linear problem then (a cone is an intersection of half-spaces). First we want to stick to the common approach and consider the \( L_2 \) norm to explain the basic ideas. It is also possible to regard different norms in different image planes depending on potential prior information. Since this does not change the situation concerning the theory of solving the problem, we assume that all \( V_i \) are equipped with the same norm.

\[ ^2 \text{from } \text{http://de.wikipedia.org/wiki/Normierter_Raum} \]
Norm in product space

Now we look for a suitable norm in the product space $V$. As we want to take the distances in the $V_i$ into account we want a norm that is derived by the norms there. Again, simple and straight forward ideas for that would be the $L_1$, $L_2$ or the $L_{\infty}$-norm, that is $\|v\|_{\infty} := \max_i \|v_i\|_{V_i}$. We will see that this is a crucial decision since the situation regarding the existence of local minima is dependent on this choice. The $L^2$-norm is in many ways the most naturally choice since assuming Gaussian noise, minimising this norm gives the ML (maximum likelihood) estimate. But also the $L_{\infty}$-norm has a statistical interpretation, namely the assumption of uniform errors (see [Ols09] for details). Convexity of the problem is related to the choice of the norm in the product space. This is the topic of the next section.

Convexity

In the following we want to give a short view over the mathematical fundament we need for the optimisation problem (2). As known from classic optimisation, a key point is convexity. In our minimisation problem (2) the question is, if the function

$$X \mapsto \|PX - v\|,$$

defined on $\text{dom}(P)$ is convex. Writing the function like this can cause confusions, since it looks like as if the problem was obviously convex by the composition of a 'seemingly' affine function and a norm (which is convex which can be seen easily). But the function $X \mapsto PX - v$ is not affine because of the projective setting. Indeed, the form can be seen in (1). Therefore we rewrite the line from above as

$$X \mapsto \left\|\left(\frac{(P_i(1,:) - v_{i1}P_i(3,:), P_i(2,:) - v_{i2}P_i(3,:))^{T}X}{P_i(3,:)X}\right)_{i=1}^{n}\right\|.
\tag{4}
$$

Note that the norm is the norm in $V$ here. The goal is to minimise this function. Since we can not expect convexity, we consider a weaker property.

**Definition** (Quasi-convex) A function $f : \text{dom}(f) \to \mathbb{R}$ with convex $\text{dom}(f)$ is called quasi-convex if all sublevel sets,

$$S_{\gamma}(f) = \{x \in \text{dom}(f) : f(x) \leq \gamma\} \quad \gamma \in \mathbb{R},$$

are convex.

By definition of convexity it follows directly that a convex function is always quasi-convex. Indeed, for a fixed $\gamma \in \mathbb{R}$ let $\theta \in [0, 1]$ and $x, y \in S_{\gamma}(f)$. By convexity we get that $f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$. But since $x, y \in S_{\gamma}(f)$ the last term is less or equal $\gamma$, hence $\theta(x + (1 - \theta)y) \in S_{\gamma}(f)$ and therefore $S_{\gamma}$ is convex.
The most important property of a quasi-convex function is that strict local minima are also global.

**Theorem 1.1** Let \( f : \text{dom}(f) \to \mathbb{R} \) be quasi-convex and \( x_0 \) a strict local minimum, i.e. there exists a radius \( \epsilon > 0 \) such that
\[
    f(x_0) < f(x) \quad \forall x \in B_\epsilon(x_0),
\]
where \( B_\epsilon(x_0) \) denotes the ball with radius \( \epsilon \) and center \( x_0 \). Then \( x_0 \) is a global minimum, i.e.
\[
    f(x_0) < f(x) \quad \forall x \in \text{dom}(f)
\]

**Proof** Denote \( \gamma = f(x_0) \). Assume that there exists a \( y \in \text{dom}(f) \setminus B_\epsilon(x_0) \) with \( f(y) \leq f(x_0) \). Clearly, \( x_0, y \in S_\gamma(f) \). Since \( S_\gamma(f) \) is convex, it is connected and hence the set \( (S_\gamma(f) \setminus \{x_0\}) \cap B_\epsilon(x_0) \) is not empty. But this is a contradiction to the assumption that \( x_0 \) is a strict local minimum.

Moreover, it can even be shown that local minima which have a neighbourhood such that the local minimum can only be attained in the interior are also global minima (nearly the same proof, see [KH08]). By definition of quasi-convexity the global minima have to lie in a convex set. So we have a generalisation of convexity of a function that guarantees us global minima from local minima. The next theorem shows another important property of quasi-convex functions which will be crucial for the choice of the norm in the product space.

**Theorem 1.2** Let \( \{f_i : \text{dom}(f_i) \to \mathbb{R} | i \in I\} \) be a family of quasi-convex functions. Then, \( \max_{i \in I} f_i \) defined on the intersection \( \bigcap_{i \in I} \text{dom}(f_i) \) is also quasi-convex.

**Proof** We use the fact that the intersection of convex sets is again convex. Define \( f := \max_{i \in I} f_i \). It follows that the domain of \( f \), \( \text{dom}(f) = \bigcap_{i \in I} \text{dom}(f_i) \) is convex. Furthermore, it is easy to see that for a \( \gamma \in \mathbb{R} \) the set \( S_\gamma(f) \) equals the intersection of all sublevel sets \( S_{\gamma_i}(f_i) \). Again, it follows that the intersection is convex, since the \( f_i \) are quasi-convex which completes the proof.

On the other hand, it can be shown that the sum of quasi-convex functions is in general not quasi-convex.

We are now going to apply this theory to our problem. Before, we state some information concerning *Cheirality*.

**Cheirality**

Basically a camera matrix \( P \) gives us nearly all informations we need to know the geometrical setting of the experiment. But in some way we have to be careful. The projective setting allows us to calculate the corresponding 3D ray of reprojected points from a 2D point of the image plan. But what happens for instance, if we multiply the matrix \( P \) by \(-1\)?! Since we have the projective setting this does not change the 3D ray for the same image point. The thing that changes is the sign of the depth parameter. From the geometrical considerations (see e.g. figure or 1) we know also that 3D points which can be visualised in the image plane have to lie 'in front of the camera’. More precisely, these points have to lie in the (open) half space defined by the image plane, which does not include the camera center. We recall briefly the definition of a camera matrix as a product
\[
P = K[R \ t],
\]
where $K \in \mathbb{R}^{3 \times 3}$ is an upper triangular matrix (where $K(3,3) = 1$) describing the interior parameters (coordinate system in the image plane), whereas $R \in \mathbb{R}^{3 \times 3}$ is an orthogonal matrix and $t \in \mathbb{R}^3$ describing the position of the image plane. From that, one can show that the image plan can be described through the last row of the matrix $P$, by $P(3,:)X = 0$ (because this is equivalent to $[R(3,:), t]X = 0$). Therefore, the requirement for a point to lie 'in front of the camera' is either

$$P(3,:)X > 0,$$

or

$$P(3,:)X < 0.$$

The object which reflects this sign of the depth is the matrix $K$. In general, and also in the experiments in this project, the decomposition of $P$ into $K$ and $R$ is not known, which is called the 'uncalibrated case'. The term 'cheirality' is used to assume that the 'camera matrices are known in an affine coordinate frame' (from [KH08]).

Now we can also define the domain of each $P_i$ which should be either

$$\{ X \in \mathbb{R}^4 : P(3,:)X > 0, X(4) = 1 \},$$

or

$$\{ X \in \mathbb{R}^4 : P(3,:)X < 0, X(4) = 1 \}.$$

Consequently $\text{dom}(P)$ is defined as the intersection of all dom($P_i$) (an intersection of half spaces, see figure 1). For the sake of simplicity we will use following notation

$$c_{\text{cheir}}P(3,:)X = |P(3,:)X| > 0,$$

where $c_{\text{cheir}} \in \{-1, 1\}$ will be called cheirality constant to distinguish between above cases.

Solving the Problem

We are going to apply the theory from above to the triangulation problem. According to the 'Cheirality' section we assume here that we are in the calibrated case and give a remark for the general case afterwards since the method will be the same. First of all, we want to formulate the problem in terms of a minimisation of a quasi-convex function. We recall the functions defined in (3) and consider a general class of functions of such a form

$$f_i(X) = \frac{\|g_i(X)\|_{V_i}}{h_i(X)},$$

where $g_i$ and $h_i$ are affine function (non-trivial) and $f_i$ is defined for $X$ such that $c_{f_i}h_i(X) > 0$ and $X(4) = 1$ for a fixed $c_{f_i} \in \{-1, 1\}$.

**Lemma 1.3** Functions $f_i : \text{dom}(f_i) \rightarrow \mathbb{R}$ of the form (6) are quasi-convex.

**Proof** Since $\text{dom}(f_i) = \{ X \in \mathbb{R}^4 : c_{f_i}h_i(X) > 0, X(4) = 1 \}$ is convex since it is an open half space (because $h_i$ is affine). For $\gamma \in \mathbb{R}$ we consider the sublevel set $S_{\gamma}(f_i)$, i.e the set of all $X \in \text{dom}(f_i)$ such that

$$\gamma \geq f_i(X) = \frac{\|g_i(X)\|}{h_i(X)},$$

Multiplying with $c_{f_i}h_i(X) > 0$ yields

$$\|g_i(X)\| - \gamma c_{f_i}h_i(X) \leq 0,$$

Now the left hand side is a convex function in $X$ since the norm and affine functions are convex and the sum of convex functions as well. In particular this function is quasi-convex. But the set of $X$ fulfilling this inequality is then the 0–sublevel set and hence convex. This we wanted to show.
After we calculated the distances in the image plane we are now turning to the product space \( V \). In order to preserve the quasi-convexity we need a product space norm that preserves this property. By theorem 1.2 we know that the maximum norm (\( L_\infty \)-norm) is the proper candidate for that.

So we have a minimisation problem of a quasi-convex function

\[
\arg \min_{X \in \text{dom}(P)} \left\| (f_i(X))_{i=1}^n \right\|_\infty.
\]

Finally we present a common method to solve such a problem.

**Cone programming**

In this section and the experiment below we will focus on Second-Order Cone Programming (for detailed information we refer to [BV04]). In particular we are considering a feasibility problem where the task is to answer the question if there exist any \( X \) satisfying a second-order cone constraint, i.e.

\[
\| g_i(X) \|_2 \leq h_i(X) \quad \forall i = 1, \ldots, n
\]

where \( g_i \) and \( h_i \) are affine functions. Using the \( L_1 \) or \( L_\infty \) norm instead leads even to a linear program. For second-order Cone Programming (SOCP) there exists efficient algorithms like in the cvx-environment for matlab [GB10]. In [KH08] it is shown that our problem (8) can be rewritten as

\[
\min_{\gamma \geq 0, X \in \text{dom}(P)} \gamma \quad \text{subject to} \quad \| g_i(X) \|_2 - \gamma \cdot c_{\text{cheir}}^i h_i(X) \leq 0 \quad \forall i = 1, \ldots, n,
\]

where \( c_{\text{cheir}}^i \in \{-1, 1\} \) is the respective cheirality constant, \( g_i \) and \( h_i \) are from representation (6). More precisely,

\[
g_i(X) = (P_i(1,:) - v_{i1} P_i(3,:), P_i(2,:) - v_{i2} P_i(3,:))^T X
\]

\[
h_i(X) = P_i(3,:) X.
\]

Problem (8) can be solved using a bisection algorithm: For fixed \( \gamma \) the feasibility problem for the constraints can be considered. This is alternated with choosing a new \( \gamma \) via bisectional iteration; depending on whether the feasibility problem has a solution or not. For details we refer again to [KH08].

**Projective Case**

As already mentioned, we often do not know the cheirality constants. Then there are \( 2^n \) cases that theoretically can occur. Basically one has to run through all these cases to see which one leads the solution. With some clever considerations one can eventually avoid running all cases, but this depends completely on the problem as described in [KH08].

**Experiments**

In this last section an example for a triangulation is covered. The data is a well known example in Computer Vision and known as 'dinosaur'\(^3\). The task is to reconstruct 3D points from a set of 36 camera matrices and given (measured) image points. Figures 5 and show images of the 'dinosaur' taken by different cameras.
The algorithm explained above was implemented in Matlab using cvx. With this environment convex optimisation problems can easily be formulated. The facts about the ’dinosaur’:

- 36 camera matrices, the camera centers are slightly above the object, arranged in a circle in the xy-plane in which the object is standing.
- 4983 tracked 3D points in the image planes (some of them are complete noise, they are not points of the object)
- REMARK: Although this model has always been called ’dinosaur’, this creature is actually ’Godzilla’ which is not a dinosaur, but some invention by the movie industry.

Instead of a classical bisection algorithm a slight modification mentioned in [KH08] was used which provides faster computation. Furthermore, the chirality constants where observed to be all equal to $-1$. The following figures show some views of the result. The camera centers and the ray orthogonal to the image planes are also included in the figures.

Further ideas

The experiment was calculated using the $L_2$-norm in the image planes which is classical and naturally motivated for the triangulation problem. Then we have a SOCP. But analogously one can take the $L_1$- or the $L_\infty$-norm. Then the computations can be done faster since we have a linear program as already mentioned. Moreover, one can choose a combination of the $L_1$ and

\[ \text{available on http://www.robots.ox.ac.uk/~vgg/data/data-mview.html} \]
the $L_\infty$ norm. The idea is to approximate the circle $\|v\|_2 = 1$ by a regular polygon. For example is it easy to find the factor $\beta > 0$ such that for $\|\cdot\| := \frac{1}{2}(\|\cdot\|_1 + \beta \|\cdot\|_\infty)$ the set $\|v\| = 1$ forms a (equiangular, regular) octagon which lies in the unit circle. The experiment was also computed with these other choices of the $V_i$ norm and one was led to similar (satisfying) results. Instead of the SOCP algorithm one should then use routines for linear programs which are more efficient.

Another idea is to weight the distance in the image plane by using a covariance matrix. This is explained in [KH08]. Finally one can also consider the angular errors of the rays instead of the distance in the image plane (again, see [KH08]).

References


