

Triangulation (computer vision)

In computer vision **triangulation** refers to the process of determining a point in 3D space given its projections onto two, or more, images. In order to solve this problem it is necessary to know the parameters of the camera projection function from 3D to 2D for the cameras involved, in the simplest case represented by the camera matrices. Triangulation is sometimes also referred to as **reconstruction** or **intersection**.

The triangulation problem is in principle trivial. Since each point in an image corresponds to a line in 3D space, all points on the line in 3D are projected to the point in the image. If a pair of corresponding points in two, or more images, can be found it must be the case that they are the projection of a common 3D point **x**. The set of lines generated by the image points must intersect at **x** (3D point) and the algebraic formulation of the coordinates of **x** (3D point) can be computed in a variety of ways, as is presented below.

In practice, however, the coordinates of image points cannot be measured with arbitrary accuracy. Instead, various types of noise, such as geometric noise from lens distortion or interest point detection error, lead to inaccuracies in the measured image coordinates. As a consequence, the lines generated by the corresponding image points do not always intersect in 3D space. The problem, then, is to find a 3D point which optimally fits the measured image points. In the literature there are multiple proposals for how to define optimality and how to find the optimal 3D point. Since they are based on different optimality criteria, the various methods produce different estimates of the 3D point **x** when noise is involved.

Contents

Introduction

Properties

Singularities

Invariance

Computational complexity

Methods

Mid-point method

Direct linear transformation

Via the essential matrix

Optimal triangulation

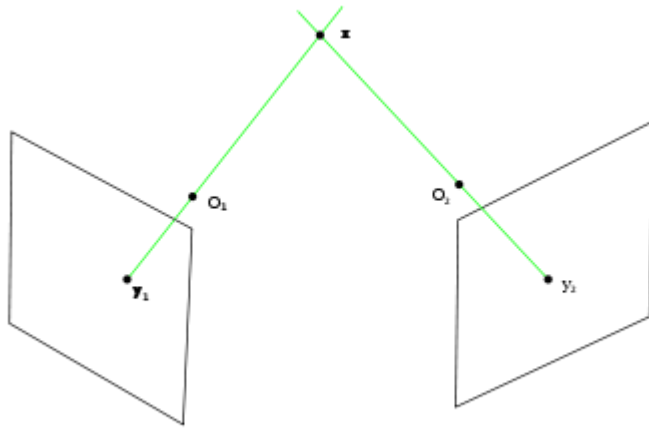
See also

References

External links

Introduction

In the following, it is assumed that triangulation is made on corresponding image points from two views generated by pinhole cameras. Generalization from these assumptions are discussed here.



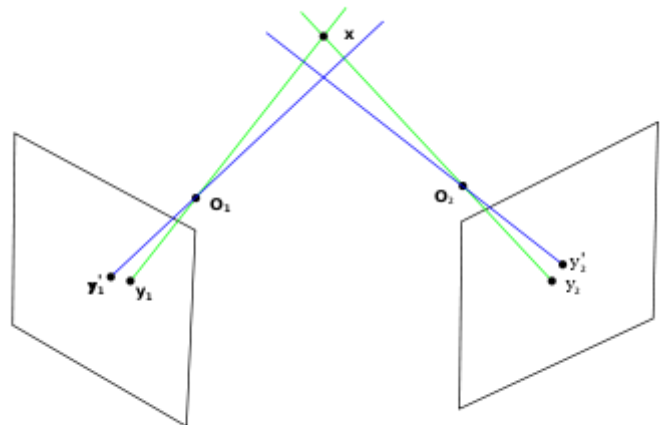
The ideal case of epipolar geometry. A 3D point x is projected onto two camera images through lines (green) which intersect with each camera's focal point, O_1 and O_2 . The resulting image points are y_1 and y_2 . The green lines intersect at x .

The image to the left illustrates the epipolar geometry of a pair of stereo cameras of pinhole model. A point x (3D point) in 3D space is projected onto the respective image plane along a line (green) which goes through the camera's focal point, O_1 and O_2 , resulting in the two corresponding image points y_1 and y_2 . If y_1 and y_2 are given and the geometry of the two cameras are known, the two projection lines (green lines) can be determined and it must be the case that they intersect at point x (3D point). Using basic linear algebra that intersection point can be determined in a straightforward way.

The image to the right shows the real case. The position of the image points y_1

and y_2 cannot be measured exactly. The reason is a combination of factors such as

- Geometric distortion, for example lens distortion, which means that the 3D to 2D mapping of the camera deviates from the pinhole camera model. To some extent these errors can be compensated for, leaving a residual geometric error.
- A single ray of light from x (3D point) is dispersed in the lens system of the cameras according to a point spread function. The recovery of the corresponding image point from measurements of the dispersed intensity function in the images gives errors.
- In a digital camera, the image intensity function is only measured in discrete sensor elements. Inexact interpolation of the discrete intensity function have to be used to recover the true one.
- The image points y_1' and y_2' used for triangulation are often found using various types of feature extractors, for example of corners or interest points in general. There is an inherent localization error for any type of feature extraction based on neighborhood operations.



In practice, the image points y_1 and y_2 cannot be measured with arbitrary accuracy. Instead points y_1' and y_2' are detected and used for the triangulation. The corresponding projection lines (blue) do not, in general, intersect in 3D space and may also not intersect with point x .

As a consequence, the measured image points are y_1' and y_2' instead of y_1 and y_2 . However, their projection lines (blue) do not have to intersect in 3D space or come close to x . In fact, these lines intersect if and only if y_1' and y_2' satisfy the epipolar constraint defined by the fundamental matrix. Given the

measurement noise in \mathbf{y}'_1 and \mathbf{y}'_2 it is rather likely that the epipolar constraint is not satisfied and the projection lines do not intersect.

This observation leads to the problem which is solved in triangulation. Which 3D point \mathbf{x}_{est} is the best estimate of \mathbf{x} given \mathbf{y}'_1 and \mathbf{y}'_2 and the geometry of the cameras? The answer is often found by defining an error measure which depends on \mathbf{x}_{est} and then minimizing this error. In the following sections, some of the various methods for computing \mathbf{x}_{est} presented in the literature are briefly described.

All triangulation methods produce $\mathbf{x}_{\text{est}} = \mathbf{x}$ in the case that $\mathbf{y}_1 = \mathbf{y}'_1$ and $\mathbf{y}_2 = \mathbf{y}'_2$, that is, when the epipolar constraint is satisfied (except for singular points, see below). It is what happens when the constraint is not satisfied which differs between the methods.

Properties

A triangulation method can be described in terms of a function τ such that

$$\mathbf{x} \sim \tau(\mathbf{y}'_1, \mathbf{y}'_2, \mathbf{C}_1, \mathbf{C}_2)$$

where $\mathbf{y}'_1, \mathbf{y}'_2$ are the homogeneous coordinates of the detected image points and $\mathbf{C}_1, \mathbf{C}_2$ are the camera matrices. \mathbf{x} (3D point) is the homogeneous representation of the resulting 3D point. The \sim sign implies that τ is only required to produce a vector which is equal to \mathbf{x} up to a multiplication by a non-zero scalar since homogeneous vectors are involved.

Before looking at the specific methods, that is, specific functions τ , there are some general concepts related to the methods that need to be explained. Which triangulation method is chosen for a particular problem depends to some extent on these characteristics.

Singularities

Some of the methods fail to correctly compute an estimate of \mathbf{x} (3D point) if it lies in a certain subset of the 3D space, corresponding to some combination of $\mathbf{y}'_1, \mathbf{y}'_2, \mathbf{C}_1, \mathbf{C}_2$. A point in this subset is then a *singularity* of the triangulation method. The reason for the failure can be that some equation system to be solved is under-determined or that the projective representation of \mathbf{x}_{est} becomes the zero vector for the singular points.

Invariance

In some applications, it is desirable that the triangulation is independent of the coordinate system used to represent 3D points; if the triangulation problem is formulated in one coordinate system and then transformed into another the resulting estimate \mathbf{x}_{est} should transform in the same way. This property is commonly referred to as *invariance*. Not every triangulation method assures invariance, at least not for general types of coordinate transformations.

For a homogeneous representation of 3D coordinates, the most general transformation is a projective transformation, represented by a 4×4 matrix \mathbf{T} . If the homogeneous coordinates are transformed according to

$$\bar{\mathbf{x}} \sim \mathbf{T} \mathbf{x}$$

then the camera matrices must transform as (\mathbf{C}_k)

$$\bar{\mathbf{C}}_k \sim \mathbf{C}_k \mathbf{T}^{-1}$$

to produce the same homogeneous image coordinates (\mathbf{y}_k)

$$\mathbf{y}_k \sim \bar{\mathbf{C}}_k \bar{\mathbf{x}} = \mathbf{C}_k \mathbf{x}$$

If the triangulation function τ is invariant to \mathbf{T} then the following relation must be valid

$$\bar{\mathbf{x}}_{\text{est}} \sim \mathbf{T} \mathbf{x}_{\text{est}}$$

from which follows that

$$\tau(\mathbf{y}'_1, \mathbf{y}'_2, \mathbf{C}_1, \mathbf{C}_2) \sim \mathbf{T}^{-1} \tau(\mathbf{y}'_1, \mathbf{y}'_2, \mathbf{C}_1 \mathbf{T}^{-1}, \mathbf{C}_2 \mathbf{T}^{-1}), \text{ for all } \mathbf{y}'_1, \mathbf{y}'_2$$

For each triangulation method, it can be determined if this last relation is valid. If it is, it may be satisfied only for a subset of the projective transformations, for example, rigid or affine transformations.

Computational complexity

The function τ is only an abstract representation of a computation which, in practice, may be relatively complex. Some methods result in a τ which is a closed-form continuous function while others need to be decomposed into a series of computational steps involving, for example, SVD or finding the roots of a polynomial. Yet another class of methods results in τ which must rely on iterative estimation of some parameters. This means that both the computation time and the complexity of the operations involved may vary between the different methods.

Methods

Mid-point method

Each of the two image points \mathbf{y}'_1 and \mathbf{y}'_2 has a corresponding projection line (blue in the right image above), here denoted as \mathbf{L}'_1 and \mathbf{L}'_2 , which can be determined given the camera matrices $\mathbf{C}_1, \mathbf{C}_2$. Let d be a distance function between a (3D line) \mathbf{L} and a \mathbf{x} (3D point) such that $d(\mathbf{L}, \mathbf{x})$ is the Euclidean distance between \mathbf{L} and \mathbf{x} . The *midpoint method* finds the point \mathbf{x}_{est} which minimizes

$$d(\mathbf{L}'_1, \mathbf{x})^2 + d(\mathbf{L}'_2, \mathbf{x})^2$$

It turns out that \mathbf{x}_{est} lies exactly at the middle of the shortest line segment which joins the two projection lines.

Direct linear transformation

Via the essential matrix

The problem to be solved there is how to compute (x_1, x_2, x_3) given corresponding normalized image coordinates (y_1, y_2) and (y'_1, y'_2) . If the essential matrix is known and the corresponding rotation and translation transformations have been determined, this algorithm (described in Longuet-Higgins' paper) provides a solution.

Let \mathbf{r}_k denote row k of the rotation matrix \mathbf{R} :

$$\mathbf{R} = \begin{pmatrix} -\mathbf{r}_1 - \\ -\mathbf{r}_2 - \\ -\mathbf{r}_3 - \end{pmatrix}$$

Combining the above relations between 3D coordinates in the two coordinate systems and the mapping between 3D and 2D points described earlier gives

$$y'_1 = \frac{x'_1}{x'_3} = \frac{\mathbf{r}_1 \cdot (\tilde{\mathbf{x}} - \mathbf{t})}{\mathbf{r}_3 \cdot (\tilde{\mathbf{x}} - \mathbf{t})} = \frac{\mathbf{r}_1 \cdot (\mathbf{y} - \mathbf{t}/x_3)}{\mathbf{r}_3 \cdot (\mathbf{y} - \mathbf{t}/x_3)}$$

or

$$x_3 = \frac{(\mathbf{r}_1 - y'_1 \mathbf{r}_3) \cdot \mathbf{t}}{(\mathbf{r}_1 - y'_1 \mathbf{r}_3) \cdot \mathbf{y}}$$

Once x_3 is determined, the other two coordinates can be computed as

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = x_3 \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

The above derivation is not unique. It is also possible to start with an expression for y'_2 and derive an expression for x_3 according to

$$x_3 = \frac{(\mathbf{r}_2 - y'_2 \mathbf{r}_3) \cdot \mathbf{t}}{(\mathbf{r}_2 - y'_2 \mathbf{r}_3) \cdot \mathbf{y}}$$

In the ideal case, when the camera maps the 3D points according to a perfect pinhole camera and the resulting 2D points can be detected without any noise, the two expressions for x_3 are equal. In practice, however, they are not and it may be advantageous to combine the two estimates of x_3 , for example, in terms of some sort of average.

There are also other types of extensions of the above computations which are possible. They started with an expression of the primed image coordinates and derived 3D coordinates in the unprimed system. It is also possible to start with unprimed image coordinates and obtain primed 3D coordinates, which finally can be transformed into unprimed 3D coordinates. Again, in the ideal case the result should be equal to the above expressions, but in practice they may deviate.

A final remark relates to the fact that if the essential matrix is determined from corresponding image coordinate, which often is the case when 3D points are determined in this way, the translation vector \mathbf{t} is known only up to an unknown positive scaling. As a consequence, the reconstructed 3D points, too, are undetermined with respect to a positive scaling.

Optimal triangulation

See also

- [Bundle adjustment](#)
- [Line–line intersection](#)

References

- Richard Hartley and Andrew Zisserman (2003). *Multiple View Geometry in computer vision*. Cambridge University Press. [ISBN 978-0-521-54051-3](#).

External links

- [Two view \(https://www.mathworks.com/matlabcentral/fileexchange/67383-stereo-triangulation\)](https://www.mathworks.com/matlabcentral/fileexchange/67383-stereo-triangulation) and [multi-view \(https://www.mathworks.com/matlabcentral/fileexchange/78955-multi-view-triangulation\)](https://www.mathworks.com/matlabcentral/fileexchange/78955-multi-view-triangulation) triangulation in Matlab

Retrieved from "[https://en.wikipedia.org/w/index.php?title=Triangulation_\(computer_vision\)&oldid=1061267712](https://en.wikipedia.org/w/index.php?title=Triangulation_(computer_vision)&oldid=1061267712)"

This page was last edited on 20 December 2021, at 18:12 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.