3D motion estimation from combined 2D–3D data of line segments

Ming Xie

INRIA, 2004 Route des Lucioles, B.P. 93, 06902 Sophia-Antipolis, France

Received 23 June 1992

Abstract


We present a new method of 3D motion estimation from line correspondence. The original idea is to explicitly take into account both 2D and 3D information of line segments. In such a way, a unique solution can be guaranteed by a linear estimation process if the correspondences of eight non-collinear line segments are known. In our method, the motion estimation can be performed in two sequential steps: The first consists in the estimation of the rotation matrix and the second in the estimation of the translation vector. The advantage of the presented method is twofold: First of all, we can have a closed-form solution which completely determines the 3D motion to be estimated. This is not the case if only 2D information of line segments is used (see Weng et al. (1992)). Secondly, our method is more elegant and needs less computation effort. In fact, our method just requires three perspective views. In contrast, the traditional methods which only employ 3D information of line segments require four perspective views and more computation effort (see Sabata and Aggarwal (1991) and Zhang and Faugeras (1991)). The experimental results presented at the end of this paper show the numerical stability and the usefulness of our method.

Keywords. 3D motion estimation, combined 2D–3D data, line segment.

1. Introduction

Motion estimation is an important task in dynamic scene analysis. Results of this study may be applied to object tracking as well as to robot/vehicle guidance. In [1] and [6], different approaches of 3D motion estimation have been reviewed. Up to now, two strategies of motion estimation have been investigated. The first one consists of estimating the motion parameters by using only 2D features (points, lines, etc.) extracted from a sequence of monocular images (see [5], [3] and [7]). The second one aims at motion estimation by using a set of 3D range data (points, lines, etc.) furnished either by a stereovision system or a range finder (see [2], [6] and [9]). The advantage of the second strategy over the first one is that the depth information allows the reduction of the complexity of motion estimation. For example, if we use two frames of points to estimate the motion parameters, the first strategy requires eight pairs of matched 2D points to guarantee the uniqueness of the solution; but if 3D information is available, then a unique solution can be derived from only three pairs of matched 3D points.

In this paper, we are going to present a new strategy of motion estimation. The original idea is to use combined 2D and 3D information in order to derive a solution for 3D motion estimation. Here, we are placed in the context of using a stereovision system illustrated by Figure 1. The features that we are interested in are line segments.
Consequently, the problems needed to be solved will be:

1. the extraction of line segments,
2. the temporal matching (or tracking) of line segments in one sequence of images (e.g. the left sequence),
3. the spatial matching of line segments between a pair of stereo images at a time instance,
4. the 3D reconstruction of line segments at a time instance,
5. the estimation of 3D motion between two frames of a camera (e.g. the left camera).

We suppose that the first four problems above have been solved. On this basis, we are going to investigate the problem of how to estimate the 3D motion between two frames of the left camera. Our solution will explicitly take into account both 2D and 3D information of line segments. In such a way, we will show that the motion estimation can be performed by a linear process in two sequential steps: the first consists in the estimation of the rotation matrix and the second in the estimation of the translation vector. The advantage of our method is twofold: First of all, we can have a closed-form solution which completely determines the 3D motion to be estimated. This is not the case if only 2D information of line segments is used (see [7]). Secondly, our method is more elegant and needs less computation effort. In fact, our method just requires three perspective views. In contrast, the traditional methods which only employ 3D information of line segments require four perspective views and more computation effort (see [6] and [9]). In order to show the usefulness of our method, we shall present two examples of results: one with images generated by a vision simulator [4] and the other with real images acquired by a stereo camera system mounted on a mobile robot.

2. Problem statement

Motion estimation occurs when a vision system and/or objects in a scene undergo displacements. If we take a camera coordinate system (e.g. the left camera coordinate system) as a reference, the motion estimation problem can be defined as how to estimate object’s displacements with respect to this reference coordinate system. Equivalently, this problem can also be considered as how to estimate a camera’s displacement with respect to a rigid object which may be either static or mobile. For these two cases, the problem of motion estimation is mathematically identical. Therefore, without loss of generality, we shall study the case where a dynamic stereo camera system observes a static rigid object which contains features such as line segments.

Consider that a pair of moving stereo cameras acquires images at two consecutive time instances: the time instance $t_1$ and the time instance $t_2$. Thus, we have two pairs of stereo images. We denote $(I_{l_1}, I_{r_1})$ the pair of stereo images obtained at the time instance $t_1$ and $(I_{l_2}, I_{r_2})$ the pair of stereo images obtained at the time instance $t_2$. Knowing the stereo geometry and stereo correspondences of line segments, we can calculate the 3D information of line segments at a time instance, say, $t_1$. Now, if we know the motion correspondences of line segments between two image frames $(I_{l_1}, I_{l_2})$, a relevant question will be how to estimate the 3D motion (displacement) of the left camera with respect to a rigid object observed. In other words, we deal with the problem of 3D motion estimation by using three perspective views, that is: $(I_{l_1}, I_{l_2}, I_{r_2})$ in which 3D information of line segments is available for the two views $(I_{l_1}, I_{r_1})$.  

---

Figure 1. The scheme of a stereovision system using line segments as feature.
Owing to the fact that \( \tan \theta = 1 \) for a diagonal line, both uncertainties can be expressed in terms of the well known parameters \( N \) and \( \delta y 
olimits \)

\[
\sin \delta \theta = \frac{\delta y}{2N}, \quad \delta \varphi = \frac{\delta y}{2\sqrt{2}}.
\]

(12)

Substituting equation (12) into equation (6) and assuming that \( \delta \varphi \) is sufficiently small (i.e., \( \sin \delta \theta \) can be approximated by \( \delta \varphi \)), the two diagonal quantization intervals of an \( N \times N \) image system can be obtained:

\[
\begin{align*}
\Delta \theta_\mathcal{Q} &= 2|\delta \theta| = \frac{|\delta y|}{N}, \\
\Delta \varphi_\mathcal{Q} &= 2|\delta \varphi| = \frac{|\delta y|}{\sqrt{2}}.
\end{align*}
\]

(13)

By contrast, if \((N,0)\) is the smallest uncertainty point and \((0,N)\) is the greatest uncertainty point, and if we ignore the second-order term \( \delta y \sin \delta \theta \), then from equation (9),

\[
\begin{align*}
\delta \varphi &= -N \sin \theta \sin \delta \theta, \\
\delta \theta &= N \cos \theta \sin \delta \varphi + \delta y \sin \theta.
\end{align*}
\]

Simplifying,

\[
\delta \theta - \sin \delta \theta = \frac{-\delta y}{2N},
\]

\[
\delta \varphi = \frac{\delta y}{2\sqrt{2}}.
\]

(14)

A consistent diagonal quantization interval can be obtained:

\[
\begin{align*}
\Delta \theta_\mathcal{Q} &= 2|\delta \theta| = \frac{|\delta y|}{N}, \\
\Delta \varphi_\mathcal{Q} &= 2|\delta \varphi| = \frac{|\delta y|}{\sqrt{2}}.
\end{align*}
\]

(15)

Case 2: Uncertainty of \( x \) for every \( y \)

The uncertainty of line parameters \((\delta \varphi, \delta \theta)\) can also give rise to an uncertainty of the \( x \) component \((\delta x)\) in the back transformation for every \( y \) component as follows:

\[
\begin{align*}
\varphi + \delta \varphi &= (x + \delta x) \cos(\theta + \delta \theta) \\
&\quad + y \sin(\theta + \delta \theta).
\end{align*}
\]

(16)

Similarly, if \( \delta \theta \) is sufficiently small, \( \cos \delta \theta \) can be approximated by 1 and subtracting equation (7) from equation (16) gives

\[
\begin{align*}
\delta \varphi &= -x \sin \theta \sin \delta \theta + y \cos \theta \sin \delta \varphi \\
&\quad + \delta x (\cos \theta - \sin \theta \sin \delta \varphi).
\end{align*}
\]

For the worst case, the diagonal line of the image plane is considered. Assume that \((0,N)\) is the smallest uncertainty point and \((N,0)\) is the greatest uncertainty point in an \( N \times N \) image and ignore the second-order term \( \delta y \sin \delta \theta \), then

\[
\begin{align*}
\delta \varphi &= N \cos \theta \sin \delta \varphi, \\
\delta \theta &= -N \sin \theta \sin \delta \varphi + \delta x \cos \theta.
\end{align*}
\]

By eliminating \( \delta \varphi \), \( \sin \delta \theta \) can be determined in terms of \( N, \theta \) and \( \delta x \). Similarly, \( \delta \varphi \) can also be expressed in terms of \( N, \theta \), and \( \delta x \):

\[
\begin{align*}
\sin \delta \theta &= \frac{\delta x \cos \theta}{N \sin \theta + N \cos \theta}, \\
\delta \varphi &= \frac{\delta x \cos \theta}{\sin \theta + \cos \theta}.
\end{align*}
\]

(17)

Owing to the fact that \( \tan \theta = 1 \) for the diagonal line, both uncertainties can be expressed in terms of the well known parameters \( N \) and \( \delta y \):

\[
\begin{align*}
\sin \delta \theta &= \frac{-\delta x \cos \theta}{2N}, \\
\delta \varphi &= \frac{-\delta x \cos \theta}{2\sqrt{2}}.
\end{align*}
\]

(18)

Substituting equation (18) into equation (6) and assuming that \( \delta \varphi \) is sufficiently small so that \( \sin \delta \theta \) can be approximated by \( \delta \varphi \), the two diagonal quantization intervals of an \( N \times N \) image system can be obtained:

\[
\begin{align*}
\Delta \theta_\mathcal{Q} &= 2|\delta \theta| = \frac{|\delta x|}{N}, \\
\Delta \varphi_\mathcal{Q} &= 2|\delta \varphi| = \frac{|\delta x|}{\sqrt{2}}.
\end{align*}
\]

(19)

By contrast, if \((N,0)\) is the smallest uncertainty point and \((0,N)\) is the greatest uncertainty point and if we ignore the second-order term \( \delta y \sin \delta \theta \), then

\[
\begin{align*}
\delta \varphi &= -N \sin \theta \sin \delta \theta, \\
\delta \theta &= -N \cos \theta \sin \delta \varphi + \delta x \cos \theta.
\end{align*}
\]
estimate the nine unknowns in $R_{1/2}$. Subsequently, knowing the rotation matrix $R_{1/2}$, the translation vector $T_{1/2}$ can be estimated from the linear equation (7).

As for the estimation of $R_{1/2}$, we consider that there will not be a very important rotation (i.e., the absolute rotation angle about its rotation axis is less than 90 degrees). Thus, the parameter $r_{11}$ will not be equal to zero. Consequently, we can define a new set of eight parameters which are:

$$
\begin{pmatrix}
    r_{12} & r_{13} & r_{21} & r_{22} & r_{23} & r_{31} & r_{32} & r_{33} \\
    r_{11} & r_{11} & r_{11} & r_{11} & r_{11} & r_{11} & r_{11} & r_{11}
\end{pmatrix}
$$

(11)

We know that a line segment gives us one linear equation like (10). So, we need at least eight line segments to establish a linear system that allows us to determine unequivocally the eight unknowns in (11). To guarantee that the linear system has full rank, the necessary and sufficient condition is that the eight line segments are not collinear in 3D space. Once the eight unknowns in (11) have been determined, by using the orthonormal constraint of a rotation matrix (e.g., $|R_2| = 1$), we can directly calculate the parameter $r_{11}$ as follows:

$$
\frac{1}{r_{11}} = \sqrt{\left(\frac{r_{21}}{r_{11}}\right)^2 + \left(\frac{r_{22}}{r_{11}}\right)^2 + \left(\frac{r_{23}}{r_{11}}\right)^2}.
$$

(12)

Knowing the estimated rotation matrix $R_{1/2}$, the linear equation (7) allows the estimation of the three unknowns in $T_{1/2}$ by using three non-collinear line segments.

So far, it has been shown that given a set of eight non-collinear line segments we can derive a unique solution for 3D motion estimation. In practice, we can use more than eight line segments if they belong to a same rigid object. The two fundamental equations of our solution are (7) and (10).

4. Experimental results

It is easy to see that the implementation of our method is straightforward if we already have algorithms for performing edge detection, line segment extraction, temporal/spatial matching and 3D reconstruction of line segments. Here, we give two examples of experimental results.

**Example 1.** This example is designed to evaluate the accuracy of our motion estimation algorithm. We use the vision simulator presented in [4] to create a scene and to specify a moving stereo camera. In this way, we can compare the estimated motion parameters with the true ones.

In our experiment, we first create a scene of a single polyhedral object which can be characterized by a set of boundary line segments. We then specify a stereo camera and its path of displacement. The distance between the vision system and the object is about 2 meters. At two consecutive time instances ($t_1, t_2$), the vision simulator produces two pairs of stereo images. Figure 2 shows these synthetical images. Between the two time instances ($t_1, t_2$), the stereo camera undergoes a relative important displacement. The true motion of the left camera is:

1. The rotation matrix $R$:

$$
R = \begin{pmatrix}
0.947381 & -0.195013 & 0.253850 \\
0.188368 & 0.980801 & 0.050473 \\
-0.258819 & -0.000000 & 0.965926
\end{pmatrix},
$$

(13)

with its axis of rotation to be $(-0.078600, 0.798362, 0.597026)$ and the angle of rotation to be 18.727978 (degree).
Figure 3. The results of matching of line segments: (a) the temporal correspondence of line segments and (b) the spatial correspondence of line segments.

2. The translation vector $T$:

$$T = (-50.002630, -9.942038, 6.711843) \text{ (cm).}$$

(14)

For each synthetical image, we apply the state-of-the-art algorithms to extract edge maps and to estimate line segments. Subsequently, we establish the stereo correspondence of line segments between the image frames $(I_1, I_2)$ as well as the motion correspondence of line segments between the image frames $(I_1, I_2)$. The matching algorithm used first estimates the disparity vectors or optical flow of contour points and then matches line segments by using a voting technique. Figure 3 shows the results of the matching of line segments. Knowing the stereo geometry (given by the vision simulator) and the stereo correspondence, we can directly calculate the 3D information of line segments. Then we apply the presented method to estimate the 3D displacement of the left camera between the time instances $(t_1, t_2)$. The estimated 3D motion of the left camera is:

1. The rotation matrix $R^*$:

$$R^* = \begin{bmatrix}
0.944936 & -0.203592 & 0.256217 \\
0.201965 & 0.978839 & 0.032940 \\
-0.257501 & 0.020620 & 0.966058
\end{bmatrix}.$$

(15)

with its axis of rotation to be $(-0.018820, 0.784751, 0.619525)$ and the angle of rotation to be 19.105722 (degree).

2. The translation vector $T^*$:

$$T^* = (-50.435058, -6.951255, 6.994973) \text{ (cm).}$$

(16)

Once the motion of the left camera between the two perspective views $(I_1, I_2)$ has been estimated, we can apply this motion to transform the set of 3D line segments reconstructed at the time instance $t_2$ (in the left camera coordinate system) into the left camera coordinate system at the time instance $t_1$. So, we can superpose it with the set of 3D line segments reconstructed at the time instance $t_1$. Figure 4 shows four views of the superposition of the two sets of 3D line segments. We can see that the motion has been correctly estimated.

Example 2. This example shows a real application of our algorithm. We have a stereo camera system mounted on an indoor mobile robot (INRIA robot). In a scene, we have a polyhedral object which undergoes a movement and is at about 1.5 meters in front of the stereo camera. The goal is to estimate the motion of this object with respect to the left camera coordinate system of the vision system. Figure 5 shows two pairs of stereo images.
acquired by the vision system at two consecutive time instances \((t_1, t_2)\). We carry out the same processing as in Example 1 to extract and match line segments. Figure 6 shows the results of stereo/motion correspondence of line segments. Then, we apply the presented algorithm to estimate the 3D motion of the polyhedral object. The estimated 3D motion is:

1. The rotation matrix \(R^*\):

\[
R^* = \begin{bmatrix}
0.980591 & 0.107180 & -0.164174 \\
-0.104814 & 0.994225 & 0.023033 \\
0.165694 & -0.005378 & 0.986162
\end{bmatrix},
\]

with its axis of rotation to be \((-0.072265, -0.839053, -0.539229)\) and the angle of rotation to be 11.336561 (degree).

2. The translation vector \(T^*\):

\[
T^* = (22.871370, -3.478399, 1.009674) \text{ (cm)}.
\]

Finally, we apply the estimated motion to transform the set of 3D line segments reconstructed at the time instance \(t_2\) (in the left camera coordinate system) into the left camera coordinate system at the time instance \(t_1\). Furthermore, we superpose it with the set of 3D line segments reconstructed at the time instance \(t_1\). Figure 7 shows four views of the superposition of the two sets of 3D line segments. We can see that the rotation component has been correctly estimated. But, the error of the \(Y\) component in the estimated translation vector is visible in Figure 7.
5. Conclusions

We have demonstrated that 3D motion estimation can be achieved by a linear estimation process if we explicitly take into account the 2D and 3D information of at least eight non-collinear line segments. The advantage of the presented method is twofold: First of all, we can have a closed-form solution which completely determines the 3D motion to be estimated. Secondly, our method is more elegant and needs less computation effort. Another interesting point is that our algorithm just requires three perspective views to perform 3D motion estimation. So, this method is directly applicable to solve the problem of estimating verging motion in an active stereovision system. Furthermore, the experimental results show the numerical stability of the presented method since we have not used perfect data as input to test our motion estimation algorithm. We must point out that a motion-based segmentation preprocessing is necessary if we want to apply the presented method to estimate 3D motion of multiple moving objects in a scene. The goal of such segmentation is to group the line segments into sets each of which belongs to a single moving object. This problem has not been addressed here.

References