sensors

MDPI

Article

A Robust Method for Ego-Motion Estimation in Urban Environment Using Stereo Camera

Wenyan Ci [1,2] **and Yingping Huang** [1,*]

[1] School of Optical-Electrical and Computer Engineering, University of Shanghai for Science & Technology, Shanghai 200093, China; wenyantz@163.com

[2] School of Electric Power Engineering, Nanjing Normal University Taizhou Colledge, Taizhou 225300, China

* Correspondence: huangyingping@usst.edu.cn; Tel.: +86-21-6511-0651

Academic Editor: Felipe Jimenez

Received: 12 July 2016; Accepted: 30 September 2016; Published: 17 October 2016

Abstract: Visual odometry estimates the ego-motion of an agent (e.g., vehicle and robot) using image information and is a key component for autonomous vehicles and robotics. This paper proposes a robust and precise method for estimating the 6-DoF ego-motion, using a stereo rig with optical flow analysis. An objective function fitted with a set of feature points is created by establishing the mathematical relationship between optical flow, depth and camera ego-motion parameters through the camera's 3-dimensional motion and planar imaging model. Accordingly, the six motion parameters are computed by minimizing the objective function, using the iterative Levenberg–Marquard method. One of key points for visual odometry is that the feature points selected for the computation should contain inliers as much as possible. In this work, the feature points and their optical flows are initially detected by using the Kanade–Lucas–Tomasi (KLT) algorithm. A circle matching is followed to remove the outliers caused by the mismatching of the KLT algorithm. A space position constraint is imposed to filter out the moving points from the point set detected by the KLT algorithm. The Random Sample Consensus (RANSAC) algorithm is employed to further refine the feature point set, i.e., to eliminate the effects of outliers. The remaining points are tracked to estimate the ego-motion parameters in the subsequent frames. The approach presented here is tested on real traffic videos and the results prove the robustness and precision of the method.

Keywords: visual odometry; ego-motion; stereovision; optical flow; RANSAC algorithm; space position constraint

1. Introduction

Vehicle ego-motion estimation is a prerequisite for applications such as autonomous navigation and obstacle detection, and therefore acts as a key component for autonomous vehicles and robotics [1]. Conventionally, vehicle ego-motion is measured using a combination of wheeled odometry and inertial sensing. However, this approach has limitations: wheeled odometry is unreliable in slippery terrain and inertial sensors are prone to drift due to error accumulation over a long driving distance, resulting in inaccurate motion estimation. Visual odometry (VO) estimates the ego-motion of an agent (e.g., vehicle and robot) using the input of a single or multiple cameras attached to it [2]. Compared to conventional wheeled odometry, visual odometry promises some advantages in terms of cost, accuracy and reliability. This paper presents a robust vehicle ego-motion estimation approach for urban environments which integrates stereovision with optical flow.

1.1. Related Work

Visual odometry has been studied for both monocular and stereo cameras. Nister et al. [3] presented one of the first visual odometry using a monocular camera. They used five consecutively matched points to generate an eigenmatrix, and accordingly solved for translational and rotational parameters with a scale factor. A major issue in monocular VO is the scale ambiguity. In monocular VO, feature points need to be observed in at least three different frames. The transformation between the first two consecutive frames in monocular vision is not fully known (scale is unknown) and is usually set to a predefined value. Ke and Kanade [4] virtually rotate the camera to the downward-looking pose to eliminate the ambiguity between rotational and translational ego-motion parameters and improve the Hessian matrix condition in the direct motion estimation process.

In comparison, stereovision enforces a baseline distance between the two cameras and can triangulate the feature positions with left and right frames to obtain their 3-dimensional (3D) coordinates, thereby resolving the scale ambiguity problem and leading to more accurate results. A solid foundation of a stereovision based visual odometry was provided in [5]. The presented algorithm is for estimating the robot's position by tracking landmarks in a scene. It proved that a stereovision system provides better estimation results than a monocular system. Since then, many stereo-vision based methods have been developed. The two cruxes of these methods are the way of establishing the objective function, and that the feature points selected for the calculation should contain inliers as much as possible. The approach presented in [6] estimated the motion parameters based on dual number quaternions, and the objective function to be minimized was expressed in 3D space. Ni et al. [7] computed poses using 3D space point correspondences. The transformation between the frames was then estimated by minimizing the 3D Euclidean distance between the corresponding 3D points and the solution was solved by an optimization procedure. The main disadvantage of reconstructing the 3D points and formulating the problem as 3D to 3D pose estimation is the sensitivity to stereo reconstruction errors. Some methods estimate the 6 degrees of freedom (6-DoF) motion and the extrinsic parameters of the camera rig in image plane through trifocal tensor [8] or quadrifocal tensor [9] approaches, which help to eliminate reconstruction errors.

Many stereo-based methods combine stereo with optical flow to establish an objective function fitted with a set of feature points, and accordingly solve for the motion parameters by minimizing the objective function [10–15]. One of the key points for these methods is that the feature points selected for computation should not contain outliers, which can be produced by moving objects, mismatching or other factors. Geiger et al. [10] projected feature points from the previous frame into 3D via triangulation using the calibration parameters of the stereo camera rig, and then re-projected them into the original image. The camera ego-motion was computed by minimizing the sum of re-projection errors. They also placed a standard Kalman filter to refine the results. Talukder et al. [11] initially estimated the motion parameters using all image points, rejected outlier points based on initially estimated motion parameters, and re-estimated the motion parameters with the iterative least mean-square error (LMSE) estimation, using filtered feature points. Kitt et al. [12] performed the Longuet–Higgins equations with a sparse set of feature points with given optical flow and disparity to reject feature points lying on independently moving objects in an iterative manner. Kreso et al. [13] removed outliers with a permissive threshold on the re-projection error with respect to the ground truth motion. Fanfani et al. [14] proposed a stereo visual odometry framework based on motion structure analysis. The approach combines the key-point tracking and matching mechanism with the effective key-frame selection strategy to improve the estimation robustness. The approach introduced in [15] was a stereo-based Parallel Tracking and Mapping (S–PTAM) method for map navigation. The process of the method was composed of two parts: tracking the features and creating a map. The extracted feature descriptors are matched against descriptors of the points stored in the map. The matches are then used to refine the estimated camera pose using an iterative least squares minimization method to minimize re-projection errors.

Urban traffic situations present difficulties for visual odometry since there are many moving objects in a scene. The point set obtained by the above methods unavoidably contains many moving points when a large area of image is composed of moving objects. To achieve true static feature points, Reference [16,17] proposed that only feature points on the ground surface were selected for ego-motion estimation. However, the methods are comparatively poor in adaptability to the environment because a plenty number of effective ground feature points are hard to be extracted in some situations. He et al. [18] utilized visual-inertial sensors to cope with complex environmental dynamics. The robustness of motion estimation was improved since they proposed a visual sanity check mechanism by comparing visually estimated rotation with measured rotation by a gyroscope. However, this method incorporates the gyroscope and thus is not a purely vision-based approach. To overcome these issues, this paper proposes a novel method for estimating the 6-DoF ego-motion in urban environments by integrating optical flow with stereovision.

1.2. Overview of the Approach

In this work, stereovision and the sparse optical flow tracking method are combined to estimate the motion state of a moving camera. An objective function is created by establishing the mathematical relationship between the optical flow caused by a camera's motion, the depth and the ego-motion parameters. The motion parameters of the camera are computed by minimizing the objective function using an iterative optimization method.

Figure 1 gives an overview of the approach in a tracking cycle. Firstly, the feature points and their optical flow are detected by using the Kanade–Lucas–Tomasi (KLT) algorithm [19]. The depth information of the feature points is obtained from the stereovision method detailed in our previous work [20]. To remove the outliers caused by KLT mismatching between the consecutive frames and stereo mismatching between left and right images, a circle matching is followed with. Since the circle matching is only conducted in the first frame of a tracking cycle, the block in the flowchart is marked with a dashed frame. The feature points selected by the KLT algorithm may contain a great deal of moving points, and must be filtered out for the minimization computation. To do so, a space position constraint (SPC) is imposed to discriminate between static points and moving points. Afterwards, the Random Sample Consensus (RANSAC) algorithm [21] is employed to further refine the feature point set, i.e., to eliminate the effects of outliers. The remaining feature points are fitted in the objective function and the Levenberg–Marquard (LM) algorithm [22] is used to solve for the six ego-motion parameters by minimizing the objective function.

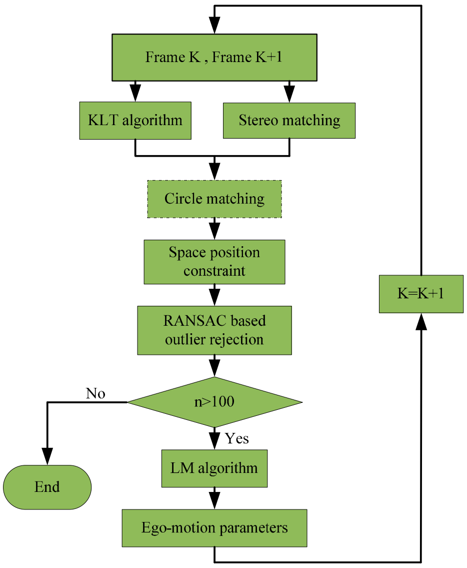


Figure 1. The flowchart of the approach in a tracking cycle.

After the calculation of the ego-motion parameters of the first frame, the remaining points are tracked in the subsequent frames. In theory, solving for six variables requires a minimum of three known static feature points for computation. In this study, the minimum number of the points involved in the computation is set to be 100 to ensure the computation accuracy. It will start the next round of tracking if the feature points drop below this value as the tracking proceeds. The SPC and RANSAC algorithm are applied in every frame. The combination of the tracking, SPC and RANSAC algorithm increases the proportion of the inliers frame by frame and also reduces computation cost.

In this work, the stereo matching is employed to measure 3D position (X, Y, Z) of feature points which are used by the objective function and SPC. In addition, the stereo matching is also used in the circle matching process.

2. Proposed Method

2.1. 3-Dimensional Motion Model and Objective Function

The camera's 3-dimensional motion and planar imaging model is represented in Figure 2. The origin of the world coordinate system $(X,\ Y,\ Z)$ is located at the center of image coordinates $(x,\ y)$, and the Z-axis is directed along the optical axis of the camera. The translational velocity of the camera is $\vec{V} = \left(V_x\ V_y\ V_z\right)$, and the rotational velocity $\vec{W} = \left(W_x\ W_y\ W_z\right)$.

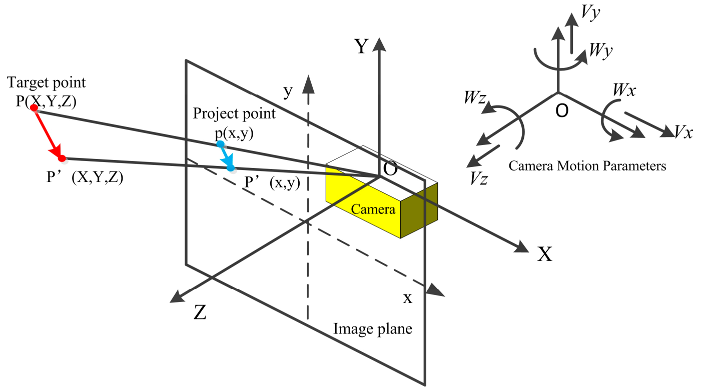


Figure 2. 3D motion and planar imaging model.

Assuming a point $P\,(X,Y,Z)$ in space relatively moves to point $P'\,(X,Y,Z)$ due to the camera's movement, the relation between the point motion and camera motion is as below:

$$\frac{dP}{dt} = -(\vec{V} + \vec{W} \times P) \tag{1}$$

The cost product of the point $P\,(X,Y,Z)$ and the camera's rotational velocity vector can be represented as:

$$\vec{W} \times P = \begin{vmatrix} i & j & k \\ W_x & W_y & W_z \\ X & Y & Z \end{vmatrix} = \left(W_y Z - W_z Y\right) i + \left(W_z X - W_x Z\right) j + \left(W_x Y - W_y X\right) k \tag{2}$$

where (i j k) denotes the unit vector in the direction of the X, Y, Z axis, × refers to the cross-product. Thus, Equation (2) can be rewritten as:

$$\vec{W} \times P = \begin{bmatrix} W_y Z - W_z Y \\ W_z X - W_x Z \\ W_x Y - W_y X \end{bmatrix} \tag{3}$$

The three dimensional velocity $\left(\frac{dX}{dt}\ \frac{dY}{dt}\ \frac{dZ}{dt}\right)$ of the point can be obtained as below:

$$\begin{aligned} dX/dt &= -\left(V_x + W_y Z - W_z Y\right) \\ dY/dt &= -\left(V_y + W_z X - W_x Z\right) \\ dZ/dt &= -\left(V_z + W_x Y - W_y X\right) \end{aligned} \tag{4}$$

For an ideal pinhole camera model, the image point p(x, y) of the world point P(X, Y, Z) projected in the image plane can be expressed as:

$$x = f\frac{X}{Z}, \quad y = f\frac{Y}{Z} \tag{5}$$

where f denotes the focal length of the stereo camera. The optical flow $\vec{v} = \left[v_x\ v_y\right]^T$ of $P\left(X, Y, Z\right)$ can be obtained by estimating the derivatives along the x-axis and y-axis in 2D image coordinates.

$$\begin{aligned} v_x &= \frac{dx}{dt} = \frac{1}{Z}\left(f\frac{dX}{dt} - x\frac{dZ}{dt}\right) \\ v_y &= \frac{dy}{dt} = \frac{1}{Z}\left(f\frac{dY}{dt} - y\frac{dZ}{dt}\right) \end{aligned} \tag{6}$$

Integrating Equations (4) to (6) yields the following:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = -\begin{bmatrix} \frac{f}{Z} & 0 & -\frac{x}{Z} & -\frac{xy}{f} & \frac{f^2+x^2}{f} & -y \\ 0 & \frac{f}{Z} & -\frac{y}{Z} & -\frac{f^2+y^2}{f} & \frac{xy}{f} & x \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \\ W_x \\ W_y \\ W_z \end{bmatrix} = \begin{bmatrix} \frac{M_V}{Z} & M_W \end{bmatrix} \begin{bmatrix} \vec{V} \\ \vec{W} \end{bmatrix} \tag{7}$$

where $M_V = \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}, M_W = \begin{bmatrix} \frac{xy}{f} & -\frac{f^2+x^2}{f} & y \\ \frac{f^2+y^2}{f} & -\frac{xy}{f} & -x \end{bmatrix}$.

Equation (7) indicates the relationship between optical flow caused by the camera's ego-motion, depth and six parameters of the camera motion. It is evident that the optical flow and the depth of a minimum of three image points must be known to solve for the six unknown variables, i.e., six camera motion parameters. In this study, the KLT algorithm is used to detect the feature points and to measure their optical flows. The depth is obtained from the stereovision method as reported in our previous work [20]. In Reference [20], a stereovision-based method has been proposed for stereo-matching between left and right images, object segmentation according to position and 3D reconstruction for acquisition of world coordinates. In the stereovision system, the world coordinates are set to be coincided with the coordinates of the left camera.

An objective function is built as:

$$E = \sum_{i=1}^{k} \omega_i \|\vec{u} - \vec{v}\|^2 \tag{8}$$

where k is the number of the image points, and ω_i is the weight of the point i determined in the following section, $\vec{v} = \left[v_x\ v_y\right]^T$ is the theoretic optical flow in Equation (7), and$\vec{u} = \left[u_x\ u_y\right]^T$ is the optical flow measured by the KLT algorithm. The camera's motion parameters $\vec{V}$ and $\vec{W}$ can be solved by minimizing Equation (8). The LM algorithm [22] is used to solve the minimization problem. The algorithm is a numerical method to estimate a set of parameters by minimizing an objective function and iteratively refining the parameters. At the first iteration, the LM algorithm works as a gradient method. As it gets near the optimal point, it gradually switches to the Newton-based approach. Good parameter initialization results in a fast and reliable model convergence. In this work, three feature points are initially selected from the point set and put into Equation (8). The computed parameters are taken as the initial value of the LM algorithm. Subsequently, the parameters measured in the current frame are used to initialize the next frame.

2.2. KLT Algorithm and Circle Matching

The KLT algorithm [19] is used to select appropriate image points to solve for the minimization problem and to measure their optical flows. The algorithm extracts the Shi–Tomasi corners from an image by computing eigenvalues and selecting the pixels whose minimum eigenvalue is above a certain value. Upon detection of the feature points, the well-known Lucas–Kanade algorithm is used to measure the optical flows of those points. The KLT algorithm also tracks those points. It takes a feature point and its neighborhood as the feature window, and uses the sum of squared difference (SSD) between the frames in the window as the tracking criteria.

Following with the KLT detection, a circle matching approach has been adopted to ensure that the feature points are matched correctly between consecutive frames. The process of the circle matching is in the order of $a \rightarrow b \rightarrow c \rightarrow d \rightarrow a$ as shown in Figure 3. The correspondence between a–b and c–d is established by the KLT algorithm while the correspondence between b–c and d–a is established by the stereo-matching detailed in [20]. A matching is accepted only if the ending feature point is consistent with the starting feature point. The KLT algorithm is normally applied in a monocular imaging sequence. The circle matching mechanism proposed here makes full use of two pairs of image sequences captured from a stereovision rig.

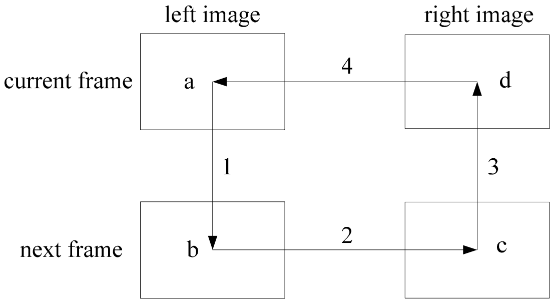


Figure 3. The process of a circle matching.

2.3. Space Position Constraint

To obtain correct motion parameters from Equation (8), all feature points used should ideally be selected from the static points such as background and ground. However, the KLT does not discriminate between static points and moving points. Urban traffic scenarios present many moving actors such as vehicles and pedestrians, therefore, the points selected by the KLT contain a great deal of moving points. In consideration of the fact that the movement of a moving point will significantly differ from the one of a static point, a space position constraint is imposed to discriminate them so that moving points will be filtered out.

The motion of the camera from frame f to f + 1 can be represented by the following 4 × 4 matrix:

$$M_{f(cam)} = \begin{bmatrix} R_f & \vec{V}_f \\ 0 & 1 \end{bmatrix} \tag{9}$$

where $\vec{V}_f$ is the translation vector, R_f is the rotation matrix given by the Rodriguez's formula:

$$R_f = I + [r]_\times \sin\theta + [r]_\times^2 (1 - \cos\theta) \tag{10}$$

where $\theta = ||\vec{W}_f||$, and

$$[\mathrm{r}]_\times = \frac{1}{\theta}\begin{bmatrix} 0 & -\mathrm{W}_{\mathrm{Z(f)}} & \mathrm{W}_{\mathrm{Y(f)}} \\ \mathrm{W}_{\mathrm{Z(f)}} & 0 & -\mathrm{W}_{\mathrm{X(f)}} \\ -\mathrm{W}_{\mathrm{Y(f)}} & \mathrm{W}_{\mathrm{X(f)}} & 0 \end{bmatrix} \tag{11}$$

Assuming that the position of a point P in frame f is $\overrightarrow{\mathrm{P_f}} = \begin{bmatrix} \mathrm{X_f} & \mathrm{Y_f} & \mathrm{Z_f} & 1 \end{bmatrix}^{\mathrm{T}}$, and the position in frame f + 1 is $\overrightarrow{\mathrm{P_{f+1}}} = \begin{bmatrix} \mathrm{X_{f+1}} & \mathrm{Y_{f+1}} & \mathrm{Z_{f+1}} & 1 \end{bmatrix}^{\mathrm{T}}$, the transform from $\overrightarrow{\mathrm{P_f}}$ to $\overrightarrow{\mathrm{P_{f+1}}}$ due to the camera motion can be expressed as below if P is an static point:

$$\overrightarrow{\mathrm{P_{f+1}}} = \mathrm{M_f}\overrightarrow{\mathrm{P_f}} \tag{12}$$

where $\mathrm{M_f}$ is the motion matrix from frame f to frame f + 1 of the point P, which can be obtained by inverting the matrix $\mathrm{M_{f(cam)}}$. Based on the motion smoothness assumption (the frame rate is high enough), the position of point P at frame $\mathrm{f}+1$ can be predicted as $\overrightarrow{\mathrm{P_{f+1_{pre}}}} = \mathrm{M_{f-1}}\overrightarrow{\mathrm{P_f}}$, where $\mathrm{M_{f-1}}$ was determined in frame $\mathrm{f}-1$. It must be noted that the predicted position depends on the camera's ego-motion and is nothing to do with object's motion. On other hand, the position of point P at frame $\mathrm{f}+1$, $\overrightarrow{\mathrm{P_{f+1_{mea}}}}$ can be measured by the stereovision-based method [20]. The difference between the measured position and the predicted position is:

$$\overrightarrow{\mathrm{c_f}} = \overrightarrow{\mathrm{P_{f+1_{mea}}}} - \overrightarrow{\mathrm{P_{f+1_{pre}}}} \tag{13}$$

For a static point, $\overrightarrow{\mathrm{c_f}}$ should be zero if both measured position and predicted position have no errors. For a moving point, $\overrightarrow{\mathrm{c_f}}$ tends to be large due to its own movement. The maximum acceptable difference between the measured position and the predicted position is given as $\overrightarrow{\mathrm{e}} = \begin{bmatrix} \Delta\mathrm{X_{max}} & \Delta\mathrm{Y_{max}} & \Delta\mathrm{Z_{max}} & 1 \end{bmatrix}^{\mathrm{T}}$. If the absolute value of any component of $\overrightarrow{\mathrm{c_f}}$ is larger than $\overrightarrow{\mathrm{e}}$, then the point is discarded. Otherwise the point is assigned a weight as following:

$$\omega = 1 - \frac{||\overrightarrow{\mathrm{c_f}}||^2}{||\overrightarrow{\mathrm{e}}||^2} \tag{14}$$

which is used in Equation (8).

Because the position of a feature point in frame f + 1 is predicated from the motion parameters in frame $\mathrm{f}-1$ ($\overrightarrow{\mathrm{P_{f+1_{pre}}}} = \mathrm{M_{f-1}}\overrightarrow{\mathrm{P_f}}$) and the motion parameters in frame $\mathrm{f}-1$ may be slightly different from the one in frame f due to the camera's acceleration, the basis of setting $\overrightarrow{\mathrm{e}}$ is the maximum possible position difference of static points caused by the camera's acceleration between the consecutive frames. In this study, $\overrightarrow{\mathrm{e}}$ is set to $\begin{bmatrix} 0.04 & 0.04 & 0.08 & 1 \end{bmatrix}^{\mathrm{T}}$ to ensure a static point will not be wrongly discarded.

It should be noted that the above setting of $\overrightarrow{\mathrm{e}}$ does not consider the effect of the error. Actually, the accuracy of the triangulated 3D points decreases with the distance. Consequently, the farther feature points may be wrongly discarded. Thus, a more proper setting of $\overrightarrow{\mathrm{e}}$ should take errors into consideration by modeling the relationship between the errors and the distance. This will be our future work.

In theory (without errors), $\overrightarrow{\mathrm{c_f}}$ is actually nothing to do with the distance of a point and only depends on its own movement. In consideration of the fact that that the errors are relatively small compared to the position change of a moving point, this method should not significantly degrade performance for farther points. Our experiments have also verified this.

2.4. RANSAC Based Outlier Rejection

The RANSAC algorithm is employed to further refine the feature point set, i.e., to eliminate the effects of outliers. The RANSAC algorithm [21] is an effective iterative tool to extract the optimal subset from a large data sample to improve the estimation accuracy. A subset S consisting of three points is selected from the points set Q derived from the space position constraint and used to estimate ego-motion parameters using the LM algorithm. The estimated ego-motion parameters are plugged into Equation (7) to compute the optical flow $\vec{v}$ of other points in Q. The points with $||\vec{u}-\vec{v}||^2$ less than a predefined threshold T are taken as inliers, which constitute a consensus set S* together with the subset S. Repeating this step for M times will generate M consensus set. The six ego-motion parameters are determined by the largest consensus set.

In this work, T is set to 1 pixel based on the experimental empirical value. The number of iterations that is necessary to guarantee a correct solution can be computed as:

$$M = \frac{\log(1-p)}{\log\left(1-(1-\epsilon)^s\right)} \tag{15}$$

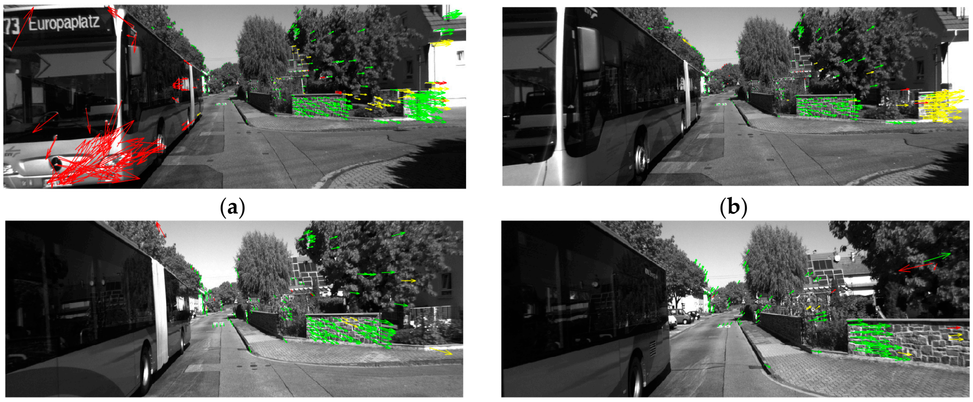
where s is the minimum number of data points needed for estimation, p is the required probability of success and ϵ defines the assumed percentage of outliers in the data set.

3. Experiments and Results

Experiments have been conducted on the public image database KITTI (Karlsruhe Institute Technology and Toyota Technological Institute) [23]. The KITTI database provides a benchmark platform for evaluation and comparison of different ego-motion estimation approaches. Furthermore, the image sequences are annotated with the ground truth of the ego-motion parameters and the depth.

Two typical urban traffic scenarios with moving objects, are selected as examples to evaluate the approach. The image resolution is 1226 × 370 pixels. The first scenario involves one oncoming bus, and the equipped vehicle moves in a longitudinal direction. In the second scenario, the vehicle is turning right in a bend and the moving object is one car that drives from left to right.

Figure 4 shows the results of one tracking cycle (frames 2624–2632) within scenario 1 after the circle matching process. The arrows represent the optical flow of the feature points. The green arrows represent the points selected for computation, the red ones are the points rejected by the SPC, and the yellow ones are the points rejected by the RANSAC algorithm. As shown in Figure 4a, all the moving feature points on the bus have been rejected by the SPC in frame 2624. Table 1 lists the number of the green points and the proportion of the inliers for the four frames. As seen in Table 1, the proportion of the inliers is increasing with the effect of the SPC and RANSAC algorithm. The minimum number of points used for the computation is 104, which is bigger than the threshold of 100.



(**a**) (**b**)

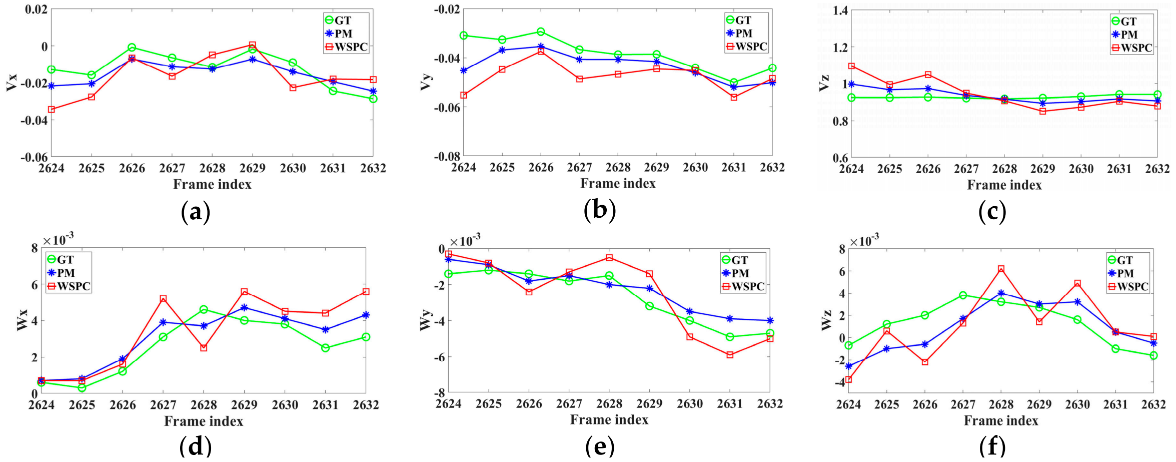
(**c**) (**d**)

Figure 4. The feature points tracking results in scenario 1. (**a**) Frame 2624; (**b**) Frame 2625; (**c**) Frame 2628; (**d**) Frame 2632. (The green arrows represent the points selected for computation, the red ones are the points rejected by the SPC, and the yellow ones are the points rejected by the RANSAC algorithm.)

Table 1. Number of the green points and proportion of the inliers.

	Frame 2624	Frame 2625	Frame 2628	Frame 2632
green points	282	200	168	104
proportion of the inliers	91.49%	93.50%	95.24%	95.19%

The ego-motion parameters of frame 2624 to frame 2632 are shown in Figure 5. The parameters are expressed in meters per frame (for translation) and in degrees per frame (for rotation). The results were obtained by the ground truth (GT), the proposed method (PM) and the proposed method without the SPC (WSPC) respectively. It can be seen that the proposed method generates smoother curves than the one without the SPC, and they are closer to the ground truth in the most frames.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 5. The results of ego-motion parameter estimation in scenario 1. (**a**) V_x; (**b**) V_y; (**c**) V_z; (**d**) W_x; (**e**) W_y; (**f**) W_z.

The ego-motion estimation error is expressed by the average translation error (ATE) and the average rotational error (ARE) [23]. The comparison results of PM and WSPC for scenario 1 are shown in Table 2. It can be seen that the PM method is much better than the WSPC method.

Table 2. Comparison of ego-motion estimation error in scenario 1.

	PM	WSPC
ATE	3.46%	8.51%
ARE	0.0028 deg/m	0.0037 deg/m

Figure 6 shows the results of one tracking cycle (frames 69–89) within scenario 2 after the circle matching process. As shown in Figure 6a, all the moving feature points on the car have been rejected by the SPC although the equipped vehicle is making a turn. Table 3 lists the number of the green points and the proportion of the inliers.

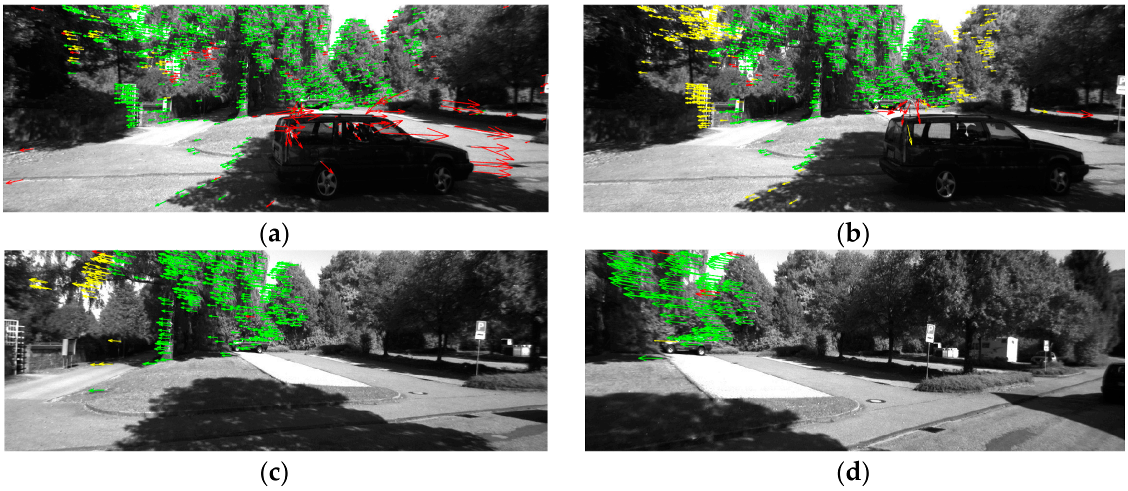
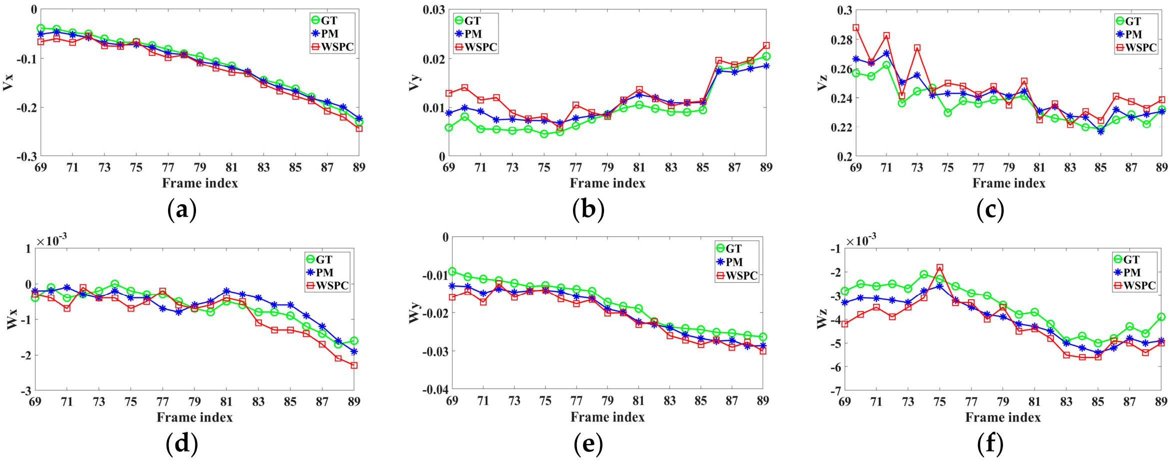

(a) (b) (c) (d)

Figure 6. The feature points tracking results in scenario 2. (**a**) Frame 69; (**b**) Frame 70; (**c**) Frame 79; (**d**) Frame 89.

Table 3. Number of the green points and proportion of the inliers.

	Frame 69	Frame 70	Frame 79	Frame 89
green points	1037	732	372	128
proportion of the inliers	92.57%	94.54%	95.16%	94.53%

The ego-motion parameters of frame 69 to frame 89 are shown in Figure 7. For most parameters, the subsequent frames generate more accurate results than the former frames due to the tracking strategy. The comparison of the ego-motion estimation error of PM and WSPC for scenario 2 is shown in Table 4.



(a) (b) (c) (d) (e) (f)

Figure 7. The results of ego-motion parameter estimation in scenario 2. (**a**) V_x; (**b**) V_y; (**c**) V_z; (**d**) W_x; (**e**) W_y; (**f**) W_z.

Table 4. Comparison of ego-motion estimation error for scenario 2.

	PM	WSPC
ATE	3.25%	5.49%
ARE	0.0089 deg/m	0.0151 deg/m

4. Evaluation and Comparison

The proposed approach has been compared to the approaches presented in References [10,15]. Reference [10] is known as the VISO2-S method, and Reference [15] is highly ranked as S-PTAM in the KITTI benchmark. The reasons for comparison with these works are: (1) they both are stereovision-based approaches; (2) they both have similar attributes with our method, i.e., solving for the ego-motion parameters by establishing an objective function and accordingly minimizing the objective function using a set of selected feature points.

4.1. Robustness

One of the performance measures of a visual odometry is the proportion of the number of inliers $\mathrm{n_{in}}$ accounting for the number of feature points used for the computation $\mathrm{n_p}$, which can be expressed as the following:

$$\mathrm{P_{in}} = \frac{\mathrm{n_{in}}}{\mathrm{n_p}} \times 100\% \tag{16}$$

If $\mathrm{P_{in}}$ is lower than 20%, ego-motion estimation may fail. In addition, the minimum value of $\mathrm{n_p}$ must be bigger than a certain number to ensure the computation accuracy. In this study, we set this number to 50. The robustness of an odometry can be evaluated by the ratio of the frames with $\mathrm{P_{in}} > 20\%$ and $\mathrm{n_p} > 50$ to the total frames within an image sequence:

$$\mathrm{P_s} = \left(\frac{\mathrm{N_{P_{in}\ >20\%\ \cap\ n_p>50}}}{\mathrm{N_f}} \right) \times 100\% \tag{17}$$

where $\mathrm{N_{P_{in}>20\%\ \cap\ n_p>50}}$ is the number of frames with $\mathrm{P_{in}}\ > 20\%$ and $\mathrm{n_p} > 50$, and $\mathrm{N_f}$ is the total number of frames. The comparison result using 11 KITTI videos is shown in Table 5.

Table 5. Comparison of robustness estimation.

	VISO2-S [10]	S-PTAM [15]	Our Method
$\mathrm{P_s}$	98.27%	98.74%	99.31%

It can be seen that the proposed method has the greatest robustness. This is mainly because our method integrates the space position constraint together with the RANSAC algorithm into the tracking process to increase the proportion of the inliers and makes sure that the number of feature points used for the computation is bigger than 100.

4.2. Absolute Trajectory Error

The absolute trajectory error proposed by Sturm et al. [24] reflects the tracking accuracy, and therefore is used for evaluation and comparison. It is calculated from the root mean squared error (RMSE) of all frames within a trajectory. The comparison result is shown in Table 6. Compared to the mean value of the 11 sequences, our method is better than the VISO2-S method and is slightly worse than the S-PTAM method. It can be noted that our method shows the best performance on sequence 03 and 05, which was collected from urban traffic. All methods have a poor performance on sequence 01, which was collected from a highway scenario with a high driving speed, resulting in a great difference between consecutive frames. The last column of the table lists the average values omitting sequence 01. It can be seen that our method performs best at this time.

Table 6. Absolute trajectory error RMSE in m.

Sequence	VISO2-S [10]	S-PTAM [15]	Our Method
00	29.54	7.66	13.47
01	66.39	203.37	227.51
02	34.41	19.81	11.35
03	1.72	10.13	1.08
04	0.83	1.03	0.96
05	21.62	2.72	1.73
06	11.21	4.10	3.04
07	4.36	1.78	5.84
08	47.84	4.93	9.48
09	89.65	7.15	5.89
10	49.71	1.96	3.16
mean	32.48	24.06	25.77
mean(w/o 01)	29.09	6.13	5.60

5. Conclusions

This paper presents a robust and precise approach for the measurement of 6-DoF ego-motion using binocular cameras. The method integrates stereovision with optical flow to build an objective function fitted with a set of feature points and accordingly solves for the ego-motion parameters by minimizing the objective function. The approach presented here is tested on the KITTI benchmark database and compared with other approaches. The experimental results demonstrate that the approach gives a robustness of 99.31% on outlier removal. The mean of the absolute trajectory error is 25.77 m for all eleven sequences provided by the KITTI. An improvement can be made on setting a more proper threshold $\vec{e}$ of the SPC by modeling the relationship between the position errors and the distance.

The proposed method follows a traditional procedure, i.e., solving for the ego-motion parameters by establishing an objective function and accordingly minimizing the objective function using a set of selected feature points. However, the contributions of this work can be found as follows: (1) The objective function was built in a unique way. The objective function fitted with a set of feature points is created in the image plane by establishing the mathematical relationship between optical flow, depth and camera ego-motion parameters through the camera's 3-dimensional motion and planar imaging model. It has significant advantages in avoiding stereo reconstruction errors over the typical 3D to 3D formulation; (2) The combination of the space position constraint with the tracking mechanism and the RANSAC algorithm effectively rejects outliers caused by moving objects and therefore greatly improves the potion of inliers, which makes the approach especially useful in an urban context; (3) The circle matching method fully makes use of two pairs of image sequences captured from a stereovision rig and effectively removes mismatching points caused by the KLT feature detection and stereo matching algorithms. In summary, the proposed approach improves the performance of visual odometry and makes it suitable to be used in an urban environment.

Acknowledgments: This work was sponsored by Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20133120110006), the National Natural Science Foundation of China (Grant No. 61374197).

Author Contributions: Yingping Huang initiated the project, proposed the method and wrote the paper; Wenyan Ci designed and implemented the algorithms, conducted experiments, analyzed the data and wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ligorio, G.; Sabatini, A.M. Extended Kalman filter-based methods for pose estimation using visual, inertial and magnetic sensors: Comparative analysis and performance evaluation. *Sensors* **2013**, *13*, 1919–1941. [CrossRef] [PubMed]

2. Scaramuzza, D.; Fraundorfer, F. Visual odometry (part I: The first 30 years and fundamentals). *IEEE Robot. Autom. Mag.* **2011**, *18*, 80–92. [CrossRef]
3. Nister, D.; Naroditsky, O.; Bergen, J. Visual odometry. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July 2004; pp. 652–659.
4. Ke, Q.; Kanade, T. Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03), Madison, WI, USA, 18–20 June 2003; pp. 390–397.
5. Matthies, L.; Shafer, S. Error modeling in stereo navigation. *IEEE J. Robot. Automat.* **1987**, *3*, 239–248. [CrossRef]
6. Milella, A.; Siegwart, R. Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In Proceedings of the 4th IEEE International Conference on Computer Vision Systems, New York, NY, USA, 4–7 January 2006.
7. Ni, K.; Dellaert, F. Stereo Tracking and Three-Point/One-Point Algorithms-A Robust Approach in Visual Odometry. In Proceedings of the 2006 International Conference on Image Processing, Atlanta, GA, USA, 8–11 October 2006; pp. 2777–2780.
8. Kong, X.; Wu, W.; Zhang, L.; Wang, Y. Tightly-coupled stereo visual-inertial navigation using point and line features. *Sensors* **2015**, *15*, 12816–12833. [CrossRef] [PubMed]
9. Comport, A.I.; Malis, E.; Rives, P. Real-time quadrifocal visual odometry. *Int. J. Robot. Res.* **2010**, *29*, 245–266. [CrossRef]
10. Geiger, A.; Ziegler, J.; Stiller, C. Stereoscan: Dense 3D reconstruction in real-time. In Proceedings of the IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, 5–9 June 2011; pp. 963–968.
11. Talukder, A.; Matthies, L. Real-time detection of moving objects from moving vehicles using dense stereo and optical flow. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, 28 September–2 October 2004; pp. 3718–3725.
12. Kitt, B.; Ranft, B.; Lategahn, H. Detection and tracking of independently moving objects in urban environments. In Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC), Funchal, Portugal, 19–22 September 2010; pp. 1396–1401.
13. Kreso, I.; Segvic, S. Improving the Egomotion Estimation by Correcting the Calibration Bias. In Proceedings of the 10th International Conference on Computer Vision Theory and Applications, Berlin, Germany, 11–14 March 2015; pp. 347–356.
14. Fanfani, M.; Bellavia, F.; Colombo, C. Accurate keyframe selection and keypoint tracking for robust visual odometry. *Mach. Vis. Appl.* **2016**, *27*, 833–844. [CrossRef]
15. Pire, T.; Fischer, T.; Civera, J.; Cristoforis, P.; Berlles, J. Stereo Parallel Tracking and Mapping for robot localization. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 1373–1378.
16. Musleh, B.; Martin, D.; De, L.; Armingol, J.M. Visual ego motion estimation in urban environments based on U-V. In Proceedings of the 2012 Intelligent Vehicles Symposium, Alcala de Henares, Spain, 3–7 June 2012; pp. 444–449.
17. Min, Q.; Huang, Y. Motion Detection Using Binocular Image Flow in Dynamic Scenes. *EURASIP J. Adv. Signal Process.* **2016**, *49*. [CrossRef]
18. He, H.; Li, Y.; Guan, Y.; Tan, J. Wearable Ego-Motion Tracking for Blind Navigation in Indoor Environments. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 1181–1190. [CrossRef]
19. Shi, J.B.; Tomasi, C. Good features to track. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600.
20. Huang, Y.; Fu, S.; Thompson, C. Stereovision-based objects segmentation for automotive applications. *EURASIP J. Adv. Signal Process.* **2005**, *14*, 2322–2329. [CrossRef]
21. Yousif, K.; Bab-Hadiashar, A.; Hoseinnezhad, R. An overview to visual odometry and visual SLAM: Applications to mobile robotics. *Intell. Ind. Syst.* **2015**, *1*, 289–311. [CrossRef]
22. Marquardt, D.W. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math.* **1963**, *11*, 431–441. [CrossRef]

23. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
24. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, 7–12 October 2012; pp. 573–580.