Improving Accuracy for Ego Vehicle Motion Estimation using Epipolar Geometry

Sergiu Nedevschi, Catalin Golban, and Cosmin Mitran

***Abstract* - This paper presents an original method for increasing the accuracy of ego vehicle motion estimation using video data. Our algorithm takes as input a monocular video sequence on which originally combines procedures for feature detection and filtering, optical flow, epipolar geometry and estimation of the rotation from the obtained essential matrix. Imposing a movement constraint on the rotation matrix, we obtain a powerful method for estimating the rotation of the vehicle from frame to frame. Furthermore, the obtained rotation and stereo data are used for computing the translation of the vehicle. The use of stereo data only for translation estimation diminishes the influence of stereo errors on rotation matrix. Experiments have been performed using various urban traffic scenes, with horizontal and vertical curvatures revealing a high degree of accuracy compared to reference measurements.**

I. Introduction

The development of automatic driving assistance tools has been an important research task in the last decades. Various data from an urban scenario can be processed and interpreted in order to provide useful information about other cars in traffic, pedestrians, traffic signs, road or the elements of environment.

Estimating the position of the ego vehicle in an urban scenario is both a desirable and a challenging task. Estimating the motion of the ego vehicle can be used in detecting and tracking various traffic objects, avoiding collisions, computing available paths and so on.

There are several techniques that allow a reliable estimation of the motion of a vehicle. Some of them involve IMUs, others use lasers and others rely on GPS systems. Even if they can be used alone, the limitations of each of these techniques generally necessitate fusing them in order to obtain more reliable results.

An emerging alternative to the above mentioned methods can be estimating the motion of the current car from video input. The video data is preprocessed in order to obtain distinctive environment features, which are then used for computing the ego car motion.

This technique is usually referred to as visual odometry. Originally proposed by Matthies [13], visual odometry is useful for a variety of reasons including the small price of the cameras, the fact that they are small and can be mounted on any vehicle and the increasing of computation power which makes the use of visual odometry even more appealing than the other traditional techniques [14]. Visual odometry can be performed both in a monocular and in a stereo configuration.

This article presents a new method for estimating the motion of a vehicle from a video sequence obtained by camera sensors. A monocular video stream proves to be enough for computing the frame to frame rotation. For better rotation results, we take into account the trajectory the vehicle can follow. This trajectory constraint increases the accuracy of the results. Stereo data are then used for computing the translation. The algorithm we have implemented has proven to give good results in urban scenarios even if the road is straight or it contains curves, regardless how crowded the traffic is.

The proposed method can be the starting point in developing more complex algorithms for separating static from moving objects or for temporally fusing the stereo data.

The paper is structured as follows: In section II we present a summary of related work. Section III presents an overview of our method and a block diagram containing the main component modules. A description of each module and of the algorithm as a whole is provided in section IV. Section V shows some experimental results and makes some comparisons with the results obtained when other sensors are used. Finally, we summarize the current work and present future plans in section VI.

II. Related work

There are methods for estimating the motion of the ego vehicle based on various types of sensors, others than cameras. A laser scanners example is presented in [11].

In robotics, there are several stereo vision based techniques for estimating the ego motion [10]. [5], [15] present methods to detect the camera motion from stereo without making use of the advantages of epipolar geometry.

The increasing available computer power has opened new possibilities to handle real time complex computer vision algorithms. Good results for vehicle ego motion estimation based on monocular video are reported by [4].

Manuscript received on May 18, 2009.

Sergiu Nedevschi is with the Technical University of Cluj-Napoca, Cluj, 400020 Romania (phone: +40-264-401219; fax: +40-264-594835; e-mail: sergiu.nedevschi@ cs.utcluj.ro).

Catalin Golban, is with the Technical University of Cluj-Napoca, Cluj, 400020 Romania. (e-mail: catalin_golban@yahoo.com).

Cosmin Mitran is with the Technical University of Cluj-Napoca, Cluj, 400020 Romania. (e-mail: cosmin_mitran@yahoo.com).

Compared to our approach, they compute moving regions which are propagated in time. Using these regions and the positions of the features in image, they reject those features which have bad correspondences and those features which are detected on moving objects. In our approach, the wrong correspondences between two frames are rejected by using two way optical flow [8] and the moving features are rejected by RANSAC [1] iterations and by the vehicle trajectory constraint.

We have used the classical 8 points algorithm for estimating the essential matrix for two consecutive frames in the video sequence. An alternative would have been the 5 points algorithm [2]. Due to the fact that we have hundreds of correspondences between two consecutive frames, we decided to use the simpler 8 points algorithm. Good tutorials for 8 points algorithm and epipolar geometry can be found in [7] and [1].

The segmentation between static and moving points can be done by using an algorithm that detects maximal cliques in graphs [5], [15]. Each two features for which the relative distance from one frame to the other is preserved are connected by edges, and the maximal complete subgraph is retained. The disadvantage of the technique is its sensitiveness to stereo reconstruction errors.

III. Method Overview

The proposed method consists of some steps that are performed for each pair of consecutive frames in the video sequence (Figure 1). The input of the algorithm is represented by the current frame and the previous one, and the outputs are the estimated rotation matrix and the estimated translation vector. First, we determine a set of point features in the previous frame. This can be done using good features to track [12]. We think that a simplified version of SIFT that doesn't compute the key-points descriptors [17] or wavelet based features [16] would have also given good results.

For the features previously determined, we compute their positions in the current frame using the optical flow [8]. Because we need very accurate correspondences between points, we calculate the optical flow from the previous image to the current one and then backwards, from the current image to the previous one. If the distance between the initial positions in the previous frame and the computed ones in the same frame exceeds a predefined threshold, then we reject the corresponding features.

Based on the correspondences filtered by the above mentioned two way optical flow, we compute the fundamental matrix using the epipolar geometry. To achieve better accuracy, we use a RANSAC approach. This will reject bad point correspondences which passed the two way optical flow step. Furthermore, knowing the intrinsic parameters of the camera, we can compute the essential matrix. Using the essential matrix, the rotation of the camera coordinate systems between the 2 consecutive frames and the translation between them up to a scale factor (direction of the translation) are estimated. Additionally, some of the RANSAC results are rejected by a condition which follows the idea that the camera is mounted on the vehicle and hence it has the same trajectory as the vehicle.

We use Kalman filtering to stabilize the results. To determine the exact value of the translation vector, we use the depths of the features points. Those depths are provided by the stereo framework [3] we used. Based on the rotation and on the direction of the translation previously computed, we determine the 3D differences between the coordinates of each selected feature. The average of these differences will represent the estimated translation vector between the consecutive frames.

IV. Practical details

In this section we describe the implementation aspects for the components in the block diagram given in Figure 1.

A. *Features detector*

The first step is to detect the features in one image. For this we prefer the corner-like or edge features that will be easily tracked based on the optical flow algorithm.

Good features to track were first introduced by Shi & Tomasi [12]. According to their paper, good features are those which have two large eigenvalues for second moment matrix (structure tensor), and they can represent corners, salt-and-pepper textures, or any other pattern that can be tracked reliably in successive frames.

B. *Optical flow*

The Lukas-Kanade Sparse optical flow algorithm [8] is based on representing the image as a pyramid. The minimization of the sum of square distances, expanded in Taylor series as a function of flow displacements, is performed starting from the top levels of the pyramid and then propagated to the bottom levels, while iteratively refining the displacement values.

Let's consider that $p_i = \begin{bmatrix} x_i & y_i \end{bmatrix}$ are the features at the previous frame. By running forward the optical flow algorithm, we'll get the correspondences $p_i' = \begin{bmatrix} x_i' & y_i' \end{bmatrix}$ and an array of flags that indicate whether the correspondence is valid or not. For the valid correspondences we perform the optical flow backward. At this stage we will obtain positions $p_i'' = \begin{bmatrix} x_i'' & y_i'' \end{bmatrix}$ and a new array of flags to indicate valid flow vectors. Once those are computed, for each valid correspondence we will impose the following constraint:

$$(x_i'' - x_i)^2 + (y_i'' - y_i)^2 < \delta^2$$

In out experiments we used $\delta = 0.2$ pixels. Figure 2 shows the features that are rejected by this mechanism for a usual traffic scene.

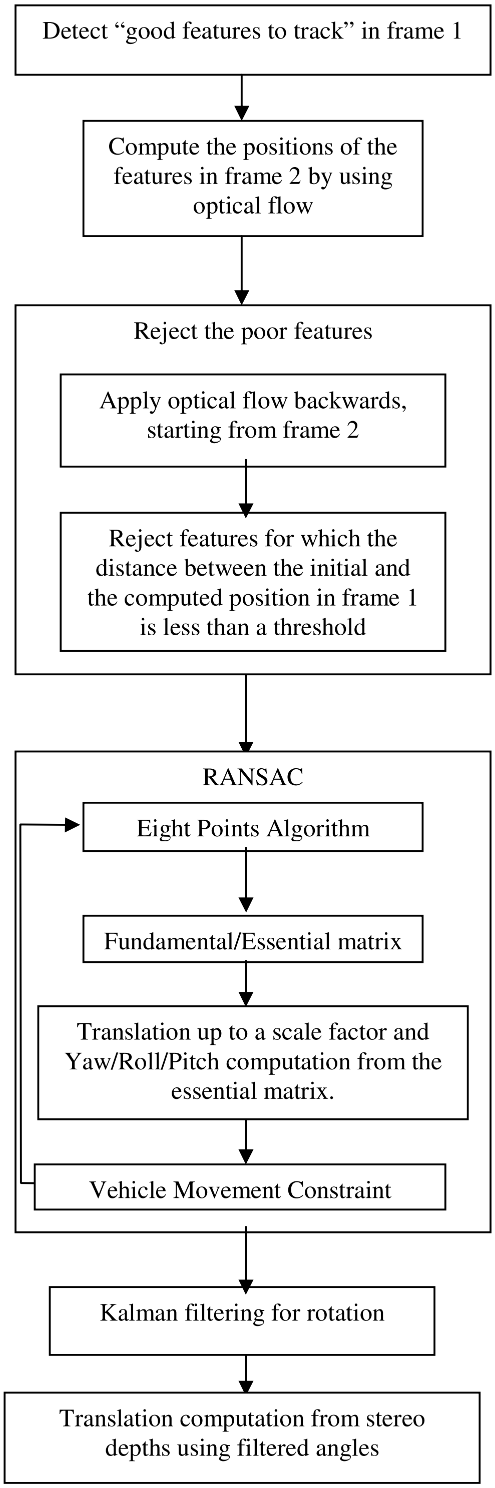
Detect “good features to track” in frame 1
Compute the positions of the features in frame 2 by using optical flow
Reject the poor features
Apply optical flow backwards, starting from frame 2
Reject features for which the distance between the initial and the computed position in frame 1 is less than a threshold
RANSAC
Eight Points Algorithm
Fundamental/Essential matrix
Translation up to a scale factor and Yaw/Roll/Pitch computation from the essential matrix.
Vehicle Movement Constraint
Kalman filtering for rotation
Translation computation from stereo depths using filtered angles

Figure 1 – Steps between two consecutive frames

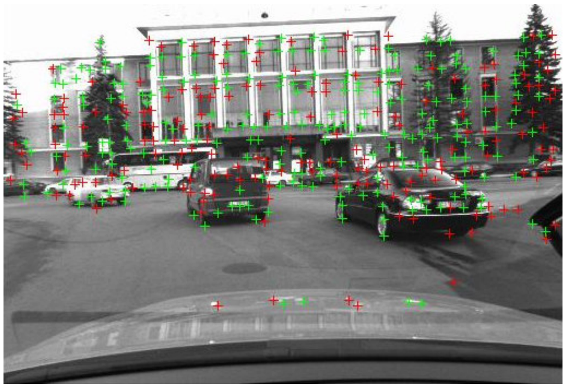

Figure 2 Features marked in red are rejected by the two way optical flow

C. Fundamental and essential matrices estimation. The eight points algorithm.

We can consider that two consecutive frames obtained by the same camera at different times are equivalent with two images of the same scene obtained with two identical cameras in a stereo system. Hence, we can apply the epipolar geometry rules between two consecutive frames.

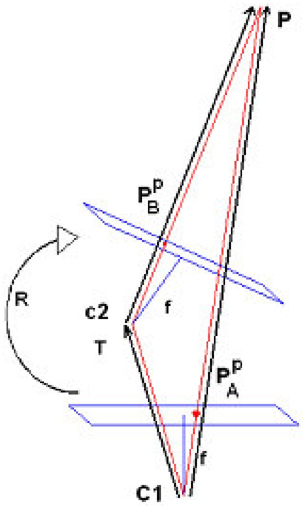

Figure 3 - Epipolar geometry for consecutive frames

Let P be a 3D point in space and the corresponding vectors of coordinates in two consecutive frames be P_A and P_B (Figure 3). P_A contains the coordinates of point P in the camera reference at previous frame and P_B contains the coordinates of point P in the camera reference for the current frame. If we denote by $T = [t_x, t_y, t_z]^T$ the coordinates of the translation vector in the camera reference at the previous frame and by R the rotation matrix between the consecutive frames, we obtain the following relationship [1]:

$$P_B{}^T E P_A = 0 \qquad (1)$$

where $E = RS$ and $S = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$.

The matrix E is called the essential matrix and establishes a mathematical relationship between the coordinates of the same point P represented in the coordinate systems of two consecutive frames.

Denoting by K the matrix containing the intrinsic parameters of the camera, the fundamental matrix is defined by

$$F = K^{-T} E K^{-1} \qquad (2)$$

Based on the fundamental matrix, the epipolar constraint, expressed in pixel coordinates, becomes:

$$(P_B{}^p)^T F P_A{}^p = 0 \qquad (3)$$

where $P_A{}^p$ and $P_B{}^p$ represent the pixel homogenous coordinates of the point P in consecutive frames.

The fundamental matrix can be computed if we have at least 8 points correspondences between two consecutive frames by using the eight points algorithm [1].

In our method, we consider as known the intrinsic parameters of the camera and we propose to use only the video sequence for determining the rotation between consecutive frames.

D. Estimate the rotation from the essential matrix

To infer the rotation matrix between two consecutive frames using the already computed essential matrix, we used the singular value decomposition based method described in [2]. Let $E \sim UDV^T$ be the singular value decomposition of the essential matrix, where U and V are chosen such that $\det(U) > 0$ and $\det(V) > 0$. Then, the translation up to scale and the rotation will be:

$$t \sim [u_{13} \quad u_{23} \quad u_{33}]^T$$

$$R_a = UMV^T \text{ or } R_b = UM^TV^T$$

The value of M is: $M = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

A proof of this result can be found in [6].

The correct rotation matrix is chosen such that the rotation angle around y-axis (yaw in our case) to be minimum. This is valid because the rotation of the vehicle between two consecutive frames is small.

E. The vehicle movement constraint

Approximating the projection of the movement of a vehicle to the xOz plane by an arc of circle was successfully applied for object tracking in [9]. We use the same motion model for the ego vehicle in order to add a supplementary constraint to the essential matrix.

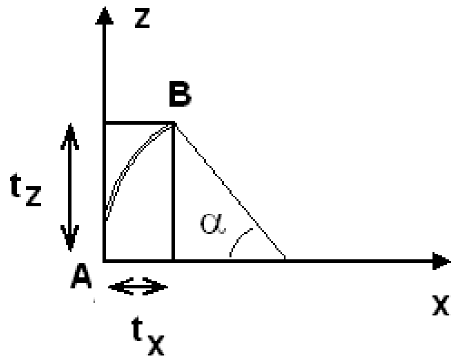

Figure 4 – Vehicle motion in the xOz plan

The relationship between the x and z of the translation and α (yaw angle) is given by the formula:

$$t_z(1-\cos(\alpha)) = t_x \sin(\alpha) \text{ or } t_z \tan(\alpha/2) = t_x$$

When the vehicle moves forward, the yaw angle is 0 and the relation becomes $t_x = 0$. In our experiments we have used the following form:

$$|t_z \tan(\alpha/2) - t_x| < \varepsilon$$

This constraint, applied to reject non-compliant fundamental matrices obtained by the RANSAC iterations because of false matches and moving object in the scene, significantly improves the final result.

F. Using RANSAC to find the best essential matrix

The output of the optical flow is a sequence of pairs of pixels, the pixels of a pair representing the same feature in two consecutive frames. Using RANSAC, we apply the eight points algorithm for a several number of iterations) and compute the fundamental matrix which is respected by the highest number of pixel pairs.

This is considered the best fundamental matrix, and it is used for determining the best essential matrix using (9). During the RANSAC iterations, a fundamental matrix which is chosen as the best so far is rejected if it doesn't respect the vehicle movement constraint.

G. Stabilizing the results

In order to stabilize the result, we have used Kalman filtering for each rotation angle separately. The values we obtain for yaw, roll and pitch using the rotation matrix from the previous to the current frame, divided by the time interval between the two frames actually represent the rates of change for the angles. Those rates will be used as measurements for the Kalman process. The state of the system is then represented by the angle rate of change and the acceleration of angle change:

$$x_t = \begin{bmatrix} \dot{\varphi}_t \\ \ddot{\varphi}_t \end{bmatrix}$$

We used a dynamic model with constant acceleration for angle change.

H. Translation

Because we used the dense stereo framework presented in [3], we are able to get the 3D coordinates for most of the points in the image. Based on the 3D coordinates, and knowing the rotation and the direction of translation, we can easily compute the exact value of the translation vector. The following equation holds for each pair of inliers from the RANSAC process:

$$m\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} - R\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

where $[x_1 \quad y_1 \quad z_1]$ represents the corresponding 3D coordinates for the pixel in the previous frame and $[x_2 \quad y_2 \quad z_2]$ represents the corresponding 3D coordinates for the pixel in the current frame. Vector t and matrix R are the translation direction and the rotation matrix obtained using essential matrix. The unknown m is obtained by a weighted average over all inliers. The weights are inverse proportional with the z coordinate of the 3D point because the error reconstruction increases with depth. Based on translation and the time interval between consecutive frames we can compute the speed of the ego vehicle. This speed is smoothed using Kalman filter by considering a constant acceleration motion model for following state vector:

$$x_t = \begin{bmatrix} v_t \\ a_t \end{bmatrix}$$

The acceleration in the above case is the hidden variable of the dynamic system.

V. Results and Evaluation

We used MATLAB to simulate how the algorithm performs in different road scenarios. The number of iterations used for RANSAC is 300. The minimum number of RANSAC iterations can be determined theoretically based on the probability of choosing an inlier ($w \approx 3/5$ in our case), the probability we expect for the algorithm to succeed ($p = 0.99$) and the number of points needed to determine the model (we used nine points for 8 points algorithm) as follows:

$$k = \frac{\log(1-p)}{\log(1-w^n)} = \frac{\log(1-0.99)}{\log\left(1-(3/5)^9\right)} = 271.87$$

The value taken for vehicle motion constraint threshold is $\varepsilon = 10^{-1}$ and it was chosen experimentally.

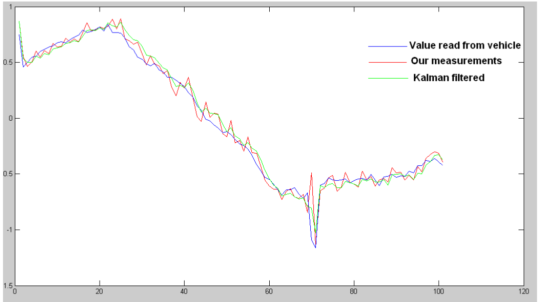

Figure 5 - Yaw estimation for 100 frames

Figure 5 shows the yaw estimate for 100 frames in a double curve (left first and then right). On horizontal axis we have the frame number and on the vertical axis the yaw from the previous frame to the current frame expressed in degrees. The blue line represents the yaw angle computed based on the yaw rate read from the vehicle and the time interval between frames. The red line represents the yaw angle determined with our method. It can be seen that even non smoothed values are very close to the values read from the vehicle. With green we represented the filtered value of the output. The red peek around frame 70 appears because of the changes in illumination caused by the shadows of the buildings. As shown in Figure 5 the algorithm recovers quickly from this error.

Figure 6 shows the roll and pitch estimation for the same set of 100 frames. As seen in the figure they both oscillate close to 0.

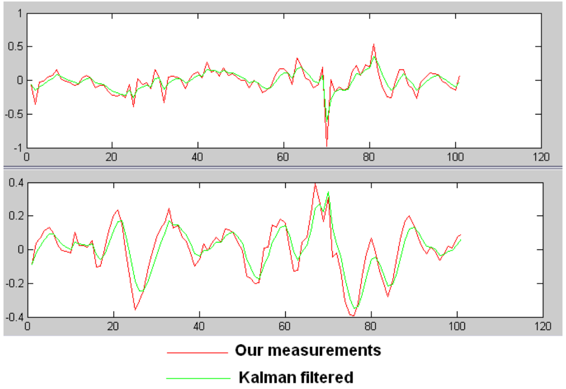

Figure 6 – Roll (upper figure) and pitch (lower figure) for 100 frames.

We have also tested how the approach performs without using the vehicle motion constraint. The results for the first 50 frames used in Figure 5 are represented in Figure 7. The oscillations around the value read from vehicle are bigger, but the graph with red still follows the blue line. This shows that the constraint we proposed increases the accuracy of the rotation estimation.

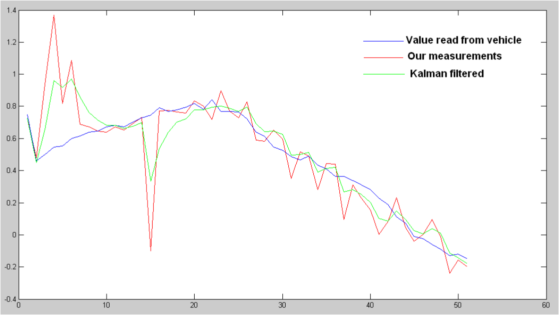

Figure 7 - Yaw estimation for 50 frames without vehicle trajectory constraint.

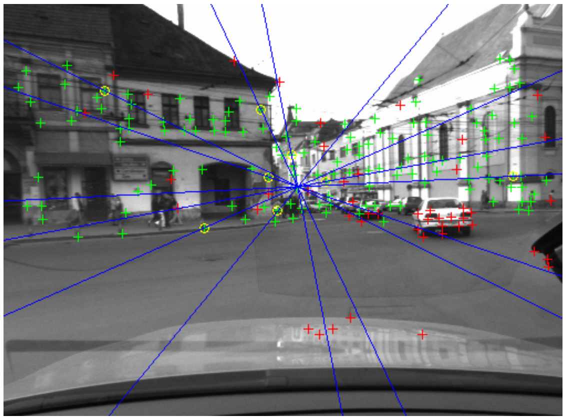

Figure 8 – Motion segmentation example.

Figure 8 shows how the approach can be used for the segmentation of moving objects in the scene from monocular video. The green points are the points used for computing the essential matrix (the points that respect the epipolar constraint). Blue lines are some epipolar lines. With red we marked the outliers reported by the RANSAC algorithm. As can be seen in the figures, some red points correspond to errors and other red points belong to moving objects. The grouping of red points on moving obstacles can be observed especially in curved roads because the two 3D correspondences calculated by optical flow and the camera centers are not in the same plane. We believe that a temporal tracking of the regions where the red points persist can lead to a good approach for motion segmentation in monocular video.

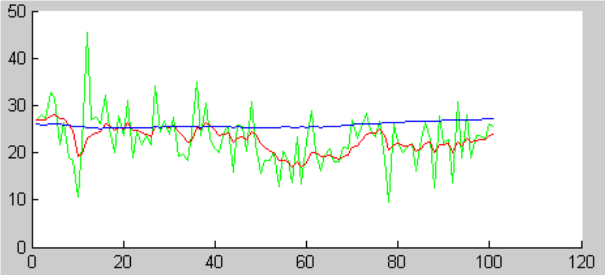

Figure 9 – Translation estimation.

Figure 9 shows how the speed is estimated for 100 frames. Blue represents the speed from the vehicle sensor. Green depicts the estimated speed in km/h, calculated based on the translation detected by our algorithm and the time interval between consecutive frames. Red represents the Kalman filtered speed.

Figure 10 shows the top view representation (right side) for the frame in the left side in the coordinate system of the first frame. It is visible how the front vehicles advance in time. With further processing, this approach can be used for motion estimation based on stereo data. The representation is made in the road plane (XOZ) in a rectangular area of 14 (7 in the left of the camera and 7 in the right) meters on the X axis and 30 meters on the Z axis. The medium height in rectangular areas of 20x20 centimeters is represented.

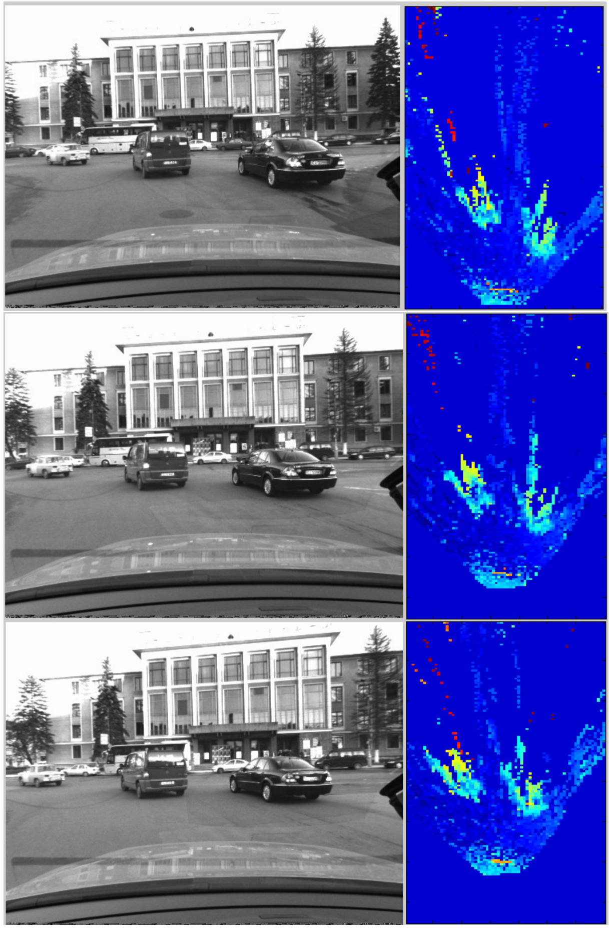

Figure 10 – 3D reconstructed data representation in a global coordinate system.

Figure 11 shows how the 3D points obtained based on stereo reconstruction can be fused for multiple frames. For the points covered by red in the left images, we have the 3D correspondents, and on the right their depths are represented. With uniform blue predominant in the upper and lower part of the image, the points for which we don't have reconstruction are represented. The upper figures represent the start frame and the lower figures represent the fused information for 5 frames.

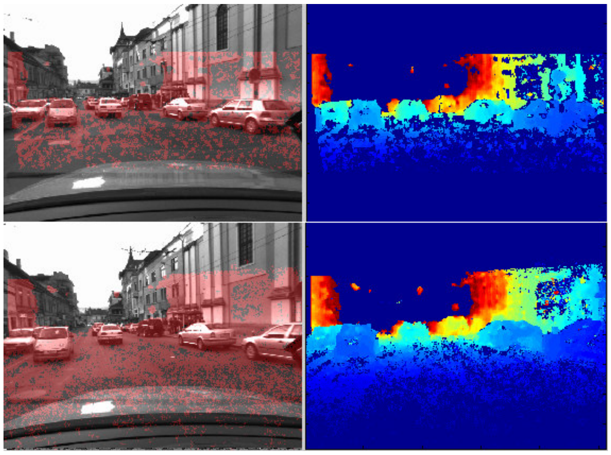

Figure 11 Temporal fusion of 3D data.

It can be observed that by fusing the 3D information in the lower figure, we have a better view of the 3D world. The left side of Figure 12 represents three dimensionally the upper part of Figure 11 and the right side of Figure 12 represents the lower part of Figure 11.

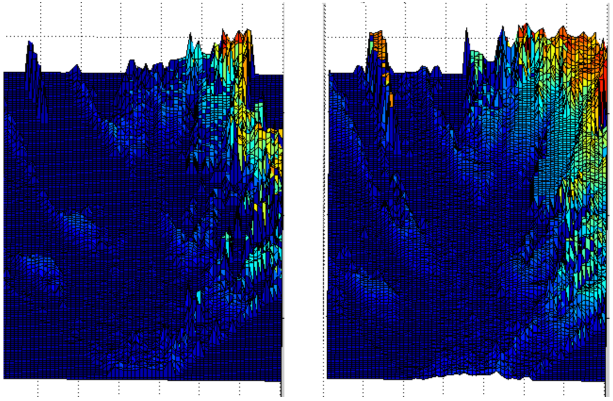

Figure 12 Three-dimensional representation of temporal fusion

From our estimations a C++ implementation with additional optimizations and profiling would lead to a real-time (20fps) or close to real time solution.

VI. Conclusions and Future Work

We described a way of combining pixel features from images, optical flow, epipolar geometry, RANSAC and Kalman filtering in order to achieve accurate motion estimation from video data. To estimate the 3D rotation matrix for consecutive frames we need only a single camera. This fact increases the estimation accuracy because we don't need to handle stereo reconstruction errors. Since translation can be determined only up to a scale factor based on monocular video, we used the 3D coordinates associated to some feature points in image for estimating the translation.

We have also shown how the estimated transform between frames could be used for motion segmentation, temporal fusion or representation of stereo data in a global coordinate frame.

Compared to other methods in the literature, we used the fact that motion is not detected between two arbitrary frames, but between two successive frames taken from a camera mounted on the vehicle. Because the projection of the vehicle motion in the xOz plane can be very well approximated by an arc of circle, we added this as an additional constraint for the essential matrix. This constraint helps the RANSAC process to infer the correct essential matrix even if there are many outliers and many features belong to objects in motion.

Some problems can occur when significant changes in illumination appear because optical flow will give poor results in that case. Improving this will be one of the following steps in this research direction.

VII. References

[1] E. Trucco, A. Verri, "Introductory Techniques for 3-D Computer Vision", Prentice Hall, 1998

[2] David Nister, "An Efficient Solution to the Five-Point Relative Pose Problem", CVPR, 2003, Volume 2, pp 195-202

[3] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, M. M. Meinecke, T. Graf, T. B. To, M. A. Obojski, "A Sensor for Urban Driving Assistance Systems Based on Dense Stereovision", IV 2007, Istanbul, Turkey, pp. 278-286.

[4] K. Yamaguchi, T. Kato, Y. Ninomiya, "Vehicle Ego-Motion Estimation and Moving Object Detection using a Monocular Camera", ICPR, 2006

[5] Andrew Howard, "Real-time stereo visual odometry for autonomous ground vehicles", International Conference on Intelligent Robots and Systems, 2008

[6] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, ISBN 0-521-62304-9, 2000.

[7] Berthold K.P. Horn, "Recovering Baseline and Orientation from Essential Matrix", 1990

[8] Jean-Yves Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker. Description of the algorithm", Intel Corporation, 2000.

[9] M. Maehlisch, W. Ritter, K. C.J. Dietmayer, "Feature level video and lidar sensor fusion for ACC stop&go using Joint Integrated Probabilistic Data Association".

[10] A. Milella, R. Siegwart, "Stereo-Based Ego Motion Estimation Using Pixel Tracking and Iterative Closest Point", ICVS, 2006

[11] S. Ono, H. Kawasaki, K. Hirahara, M. Kagesawa, K. Ikeuchi, "Ego Motion Estimation for Efficient City Modeling by Using Epipolar Plane Range Image Analysis", ITSWC, 2003

[12] Jianbo Shi and Carlo Tomasi, "Good features to track", Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn., pp 593-600, 1999.

[13] L.H. Matthies, "Dynamic Stereo Vision", Carnegie Mellon University, Ph.D. thesis, 1989.

[14] R. Roberts, H. Nguyen, N. Krishnamurthi, T. Balch, "Memory-Based Learning for Visual Odometry", ICRA, 2008, pp 47-52.

[15] H. Hirschmuller, P. Innocent and J. Garibaldi, "Fast, Unconstrained Camera Motion Estimation from Stereo without Tracking and Robust Statistics", ICARCV, 2002

[16] S. G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. VOL II . NO. 7. JULY 1989, pp 674-693

[17] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints". International Journal of Computer Vision, 2004, pp 91–110.