An experiment on relative rotation estimation from distant points with monocular vision

Catalin Golban, Sergiu Nedevschi

Abstract—**This paper proposes a method to determine the relative rotation between two images acquired with a camera. It is considered that the camera is calibrated, and that the relative motion between the two images is small. The method is appropriate for a camera mounted on a moving vehicle and it is proven that the changes in yaw, pitch and roll angles can be accurately determined in this setup. We propose a RANSAC process that selects the distant points based on a new motion model for image pixels valid only for the points at infinite. Additionally, robustness of the method is increased by considering the fact that image deformations are 0 for the points at infinite.**

I. Introduction

Visual odometry (i.e. detect the motion of the camera from the sequence of frames acquired) algorithms gained increasing popularity in that last 10 years especially because complex algorithms that used to be slow in the past can be optimized to achieve real-time performance even by running on low power CPU architectures.

The initial visual odometry solutions were intended for Mars Exploration Rovers [33] where GPS satellites system is not available but its usage in autonomous vehicles and advanced driver assistance systems is constantly increasing.

Today's cars are more and more equipped with video cameras, used for different tasks such as obstacle detection, lane departure warning, pedestrian detection etc. In this context, it makes sense to use the video data to precisely estimate the motion of the vehicle. For long term localization accurate results are generally obtained by fusing information from sensors. Sensor fusion brings improvements especially in situations when one of the sensors accumulates error over time (wheel encoders, IMU, visual odometry), or when one of the sensors is sensitive to external factors (wheel odometry is sensitive to slippage in uneven terrain, video cameras are sensitive to illumination changes, GPS is sensitive to weather conditions).

The output of the method presented in this paper can be fused with the information from other sensors for improving the localization and can bring into the system important information like roll and pitch angles. Because it is computationally efficient it can be implemented to work real time on low power CPUs.

Manuscript received on April 15, 2013.

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 (e-mail: catalincosminx@gmail.com).

We will also present the mathematical equations for finding the direction of the translation, but the focus of the paper will be strictly on the rotation estimation.

In the context of stereo systems, our major goal is to increase robustness of the rotation estimation of the existing stereo visual odometry methods in extreme situations where for example the illumination changes only for one of the stereo cameras. This is one additional motivation of this work.

The basic idea of the method was inspired from a stereo-based visual odometry algorithm. It was noticed that the terms involving translation motion in the algorithm used in [10], [11], and [12] cancel for points having 0 disparities (points at 'infinite'). Combining this into a RANSAC (**Ran**dom **Sa**mple **C**onsensus) algorithm leads to accurate estimation of the yaw, pitch and roll angles. To increase the robustness we use the fact that image deformations [1] are zero for distant points. The notion of points at infinite will spear quite often in the paper; it refers to the points that are far away from the camera.

II. Related Work

Many monocular methods [8] [9] compute the rotation and the translation up to a scale factor from the essential matrix [2] as shown in [7]. The essential matrix can be computed from feature correspondences using the 8 points algorithm [2] or using the newer 5 points algorithm [13]. Direction of heading only can be computed based on image deformations as shown in [1]. The monocular methods proposed in [6] and [14] use more than two consecutive frames and combine epipolar geometry with 3D triangulation in an iterative algorithm that is also able to resolve the scale factor ambiguity. In [15] a line based monocular method is presented. Reference [4] proposes a visual odometry method based on a rear parking camera while [5] proposes a method that combines the epipolar geometry and the homography induced by the road plane. A comparison between some of the monocular ego-motion approaches is presented in [3].

The background of the stereo method that was the starting point of this study is detailed in [10], [11] and [12].

III. Method Description

A. Overview

Firstly, corresponding feature points between consecutive image frames are determined. A variety of approaches like Harris corners [19], SURF [22] or SIFT [21] features can be used at this step. We use the so called "good features to track" [16] computed in the previous frame and a sparse optical flow algorithm for feature matching [18]. A motion model in the image plane for distant points is then used for

computing the rotation parameters based on the maximum set of inliers that result from performing a certain number of RANSAC iterations [23].

A. Starting point

As shown in [12] the values of the coordinates in current frame based on the motion parameters and on the coordinates in the previous frame can be computed as:

$$x_{2,l}^i = f \cdot \frac{x_{1,l}^i - \gamma y_{1,l}^i - \beta f + t_x \cdot \frac{\delta_1^i}{d}}{\beta x_{1,l}^i + \alpha y_{1,l}^i + f + t_z \cdot \frac{\delta_1^i}{d}} \quad (1)$$

$$y_{2,l}^i = f \cdot \frac{\gamma x_{1,l}^i + y_{1,l}^i - \alpha f + t_y \cdot \frac{\delta_1^i}{d}}{\beta x_{1,l}^i + \alpha y_{1,l}^i + f + t_z \cdot \frac{\delta_1^i}{d}} \quad (2)$$

We also have the following system of equations that relates the point coordinates and the motion parameters:

$$A = \begin{bmatrix} x_{2,l}^i y_{1,l}^i & x_{1,l}^i x_{2,l}^i + f^2 & f y_{1,l}^i & \frac{-f\delta_1^i}{d} & 0 & \frac{x_{2,l}^i \delta_1^i}{d} \\ y_{1,l}^i y_{2,l}^i + f^2 & x_{1,l}^i y_{2,l}^i & -f x_{1,l}^i & 0 & \frac{-f\delta_1^i}{d} & \frac{y_{2,l}^i \delta_1^i}{d} \end{bmatrix} \quad (3)$$

$$A \cdot [\alpha \quad \beta \quad \gamma \quad t_x \quad t_y \quad t_z]^T = \begin{bmatrix} f(x_{1,l}^i - x_{2,l}^i) \\ f(y_{1,l}^i - y_{2,l}^i) \end{bmatrix} \quad (4)$$

Images are rectified images and it is assumed that the intrinsic parameters matrix is known. The triplets $(x_{1,l}^i, y_{1,l}^i, \delta_1^i)$ and $(x_{2,l}^i, y_{2,l}^i, \delta_2^i)$ represent a pair of point correspondences between previous frame and current frame expressed in metric units (obtained by multiplying with the inverse of the intrinsic parameters matrix) and their disparities. The motion parameters are $\alpha, \beta, \gamma, t_x, t_y, t_z$; f is the focal distance and d is the baseline of the stereo rig. Thus in the stereo setup, for each feature correspondence we get two equations with 6 unknowns.

B. Adaptation for single camera

The equations above are all valid for stereo a camera mounted on a moving vehicle. First order Taylor expansion of the rotation matrix is used as explained in [10]. The motions that can appear on camera mounted on a moving vehicle are small enough to make the approximation possible.

Looking at the equations (1) - (4), we can take the limits $\delta_1^i \to 0$ and $\delta_2^i \to 0$. Then the equations modify as shown below.

Prediction equations:

$$x_{2,l}^i = f \cdot \frac{x_{1,l}^i - \gamma y_{1,l}^i - \beta f}{\beta x_{1,l}^i + \alpha y_{1,l}^i + f} \quad (5)$$

$$y_{2,l}^i = f \cdot \frac{\gamma x_{1,l}^i + y_{1,l}^i - \alpha f}{\beta x_{1,l}^i + \alpha y_{1,l}^i + f} \quad (6)$$

Then we have the following system of equations that relates the point coordinates and the rotation parameters:

$$A = \begin{bmatrix} x_{2,l}^i y_{1,l}^i & x_{1,l}^i x_{2,l}^i + f^2 & f y_{1,l}^i \\ y_{1,l}^i y_{2,l}^i + f^2 & x_{1,l}^i y_{2,l}^i & -f x_{1,l}^i \end{bmatrix} \quad (7)$$

$$A \cdot [\alpha \quad \beta \quad \gamma]^T = \begin{bmatrix} f(x_{1,l}^i - x_{2,l}^i) \\ f(y_{1,l}^i - y_{2,l}^i) \end{bmatrix} \quad (8)$$

Form one point correspondences we have 2 equations 3 unknowns. From two point correspondences we have 4 equations 3 unknowns, resulting an over determined system that can be solved with the state of the art methods.

Because it is difficult to determine the points with 0 disparities in a monocular setup a RANSAC based approach can be used to achieve convergence on those points as it will be detailed later in the paper.

C. An alternative proof

Consider a pair of corresponding 3D points: $P_1^i = [X_1^i \quad Y_1^i \quad Z_1^i]$ in the previous camera coordinate frame and $P_2^i = [X_2^i \quad Y_2^i \quad Z_2^i]$ in the current camera coordinate frame. The points are related by the camera rotation and translation as follows:

$$[X_2^i \quad Y_2^i \quad Z_2^i]^T = R \cdot [X_1^i \quad Y_1^i \quad Z_1^i]^T + [t_x \quad t_y \quad t_z]^T \quad (9)$$

Let $(x_{1,l}^i, y_{1,l}^i)$ and $(x_{2,l}^i, y_{2,l}^i)$ be the projections of P_1^i and P_2^i in the image plane. Their values are expressed in metric units. Based on the Pinhole camera model:

$$\frac{f}{Z_1^i} = \frac{x_{1,l}^i}{X_1^i} = \frac{y_{1,l}^i}{Y_1^i} \quad (10)$$

$$\frac{f}{Z_2^i} = \frac{x_{2,l}^i}{X_2^i} = \frac{y_{2,l}^i}{Y_2^i} \quad (11)$$

From (9), (10) and (11) it follows that:

$$Z_2^i \cdot \begin{bmatrix} \frac{x_{2,l}^i}{f} \\ \frac{y_{2,l}^i}{f} \\ 1 \end{bmatrix} = R \cdot Z_1^i \begin{bmatrix} \frac{x_{1,l}^i}{f} \\ \frac{y_{1,l}^i}{f} \\ 1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad \text{which is equivalent to:}$$

$$\frac{Z_2^i}{Z_1^i} \cdot \begin{bmatrix} x_{2,l}^i \\ y_{2,l}^i \\ f \end{bmatrix} = R \cdot \begin{bmatrix} x_{1,l}^i \\ y_{1,l}^i \\ f \end{bmatrix} + \frac{f}{Z_1^i} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (12)$$

Now considering $Z_1^i \to \infty$ and that $\lim_{Z_1^i \to \infty} \frac{Z_2^i}{Z_1^i} = c$, we get:

$$c \cdot \begin{bmatrix} x_{2,l}^i \\ y_{2,l}^i \\ f \end{bmatrix} = R \cdot \begin{bmatrix} x_{1,l}^i \\ y_{1,l}^i \\ f \end{bmatrix} \quad (13)$$

Let r_1, r_2, and r_3 be the lines of the rotation matrix R. By dividing with the last value we get:

$$x_{2,l}^i = f \cdot {r_1}^T \begin{bmatrix} x_{1,l}^i \\ y_{1,l}^i \\ f \end{bmatrix} \left({r_3}^T \begin{bmatrix} x_{1,l}^i \\ y_{1,l}^i \\ f \end{bmatrix} \right)^{-1}$$
$$y_{2,l}^i = f \cdot {r_2}^T \begin{bmatrix} x_{1,l}^i \\ y_{1,l}^i \\ f \end{bmatrix} \left({r_3}^T \begin{bmatrix} x_{1,l}^i \\ y_{1,l}^i \\ f \end{bmatrix} \right)^{-1} \quad (14)$$

If the rotation matrix is approximated with $R = \begin{bmatrix} 1 & -\gamma & -\beta \\ \gamma & 1 & -\alpha \\ \beta & \alpha & 1 \end{bmatrix}$ then we get exactly the same formulas as in (5) and (6).

D. Few notes on translation estimation

From the epipolar constraint the vectors $\begin{bmatrix} x_{2,l}^i \\ y_{2,l}^i \\ f \end{bmatrix}$, $R \begin{bmatrix} x_{1,l}^i \\ y_{1,l}^i \\ f \end{bmatrix}$, and $\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$ are coplanar. Let $\begin{bmatrix} xx_{1,l}^i \\ yy_{1,l}^i \\ ff \end{bmatrix} = R \begin{bmatrix} x_{1,l}^i \\ y_{1,l}^i \\ f \end{bmatrix}$. Note the double letters used as notation. This means that:

$$\begin{bmatrix} x_{2,l}^i & y_{2,l}^i & f \end{bmatrix} \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} xx_{1,l}^i \\ yy_{1,l}^i \\ ff \end{bmatrix} = 0 \quad (15)$$

By algebraic manipulation it leads to:

$$t_x\left(-y_{2,l}^i \cdot ff + yy_{1,l}^i \cdot f\right) + t_y\left(x_{2,l}^i \cdot ff - xx_{1,l}^i \cdot f\right) + t_z\left(-x_{2,l}^i \cdot yy_{1,l}^i + y_{2,l}^i \cdot xx_{1,l}^i\right) = 0 \quad (16)$$

From equation (16) the translation can be determined in two ways:

1. By considering $t_z = 1$, and solving for the other two unknowns from two corresponding 3D points.
2. By considering the equation as a dot product between the translation vector and some other vector *v*, and by doing a cross product between two such vectors in order to compute the translation direction.

The actual translation value can be computed based on the vehicle speed. The translation evaluation is not handled in this paper as it is a more or less straightforward step after rotation is obtained and distance points are eliminated.

E. Image deformations for increased robustness

Given the metric coordinates of two corresponding feature points $F_{1,i} = \left(x_{1,l}^i, y_{1,l}^i\right)$ and $F_{1,j} = \left(x_{1,l}^j, y_{1,l}^j\right)$ in the previous and current frames, we can compute the angle $\varphi_{i,j}^1 = \sphericalangle F_{1,i} O F_{1,j}$, where O is the center of the camera, using the cosine theorem:

$$\varphi_{i,j} = \cos^{-1} \frac{x_{1,l}^i x_{1,l}^j + y_{1,l}^i y_{1,l}^j + 1}{\sqrt{{x_{1,l}^i}^2 + {y_{1,l}^i}^2 + 1} \cdot \sqrt{{x_{1,l}^j}^2 + {y_{1,l}^j}^2 + 1}} \quad (17)$$

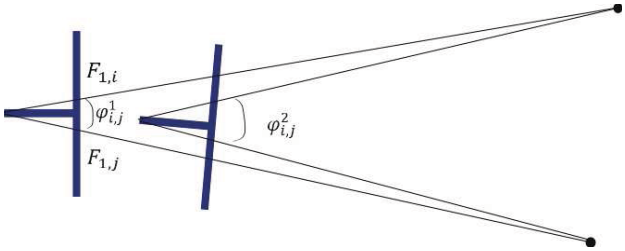

Figure 1. Top view of image deformations; $\theta_{i,j} = \left|\varphi_{i,j}^2 - \varphi_{i,j}^1\right|$

If the camera center stays the same for two consecutive frames (rotation only) then the angle between two corresponding features will remain unchanged. In the same way, if the projections $F_{1,i}$ and $F_{1,j}$ are projections of the points at 'infinite', the angles between features will stay unchanged. Let $\varphi_{i,j}^1$ be the angle between features i and j in the previous frame and $\varphi_{i,j}^2$ the angle in the current frame. The variations of the angles between features as camera moves are called image deformations and were introduced in [1] for computing the direction of heading based on monocular vision.

In our algorithm we will consider the fact that for distant points the image deformations should be zero. More precisely, the value $\theta_{i,j} = \left|\varphi_{i,j}^2 - \varphi_{i,j}^1\right|$ is very close to 0 for the points at infinite.

F. Overall algorithm

Below are the high level steps that are performed for every frame.

Step1: Compute feature correspondences between previous frame and current frame. This can be achieved with optical flow based methods, or with feature matching methods [16], [17], [18], [19]. To achieve illumination invariance the method based on rank transform [20][12] can be used. Bidirectional matches are computed in order to reject poor matches.

Step2: Select a random pair of point correspondences and check that they are distant points based of deformations. If deformation is not close to 0 then repeat **Step2**.

Step3: Compute the rotation from random pairs of points.

Step4: For the rotation computed at **Step3** find how many other points move according to that rotation. The rotation producing the maximum number of inliers is considered to be the best rotation between the pair of frames.

IV. Evaluation and Results

Following sections show different tests that were performed for evaluating and validating the theoretical facts.

A. Image deformations example

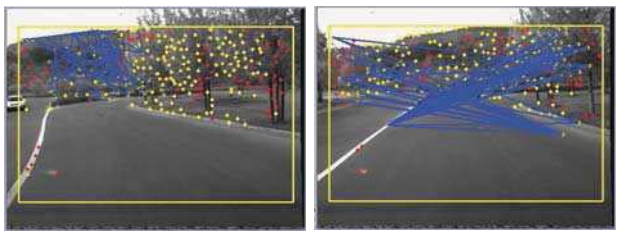

Figure 2. Image deformations. Left image indicates the pairs of points producing the 50 smallest deformation values. Rght image indicates the pairs of points producing the 50 highest deformation values.

As experiment we computed the deformations for all the pairs of point correspondences. The point closer to the camera will produce the highest deformation values while the distant points will tend to produce the smallest deformation values. Figure 2 illustrates it. The features marked with yellow are considered for rotation estimation while the features marked with read were rejected because the matches were not reliable. If the pair of points considered for certain RANSAC iteration produce a deformation higher than a threshold then it does not make sense to compute re maximum consensus set. We use a value of 0.01 degrees as threshold. This condition significantly increases the execution speed. Another important application of image deformations may be the decision if distant points are available in the image. However, this is not just a matter of applying a threshold for the deformation values and is out of the scope of this paper.

B. RANSAC convergence on distant points

Figure 3 shows with green the inliers resulting after the RANSAC iterations. Using a model that is valid only for the points at infinite, the maximum set of inliers will be detected as being far away from the camera. We believe that this method for determining the distant points can be helpful for other applications like image segmentation or background subtraction. Also it is visible that the green inliers tend to group in clusters (not necessary one cluster). If this does not happen then it may be an indication that the scene does not contain distant points.

Figure 3. Inliers after RANSAC iterations

C. How features distance affects the result?

Because the work was done on a stereo framework we evaluated the measurement error that appears if we don't consider for rotation estimation the features having the depth greater than a certain threshold. This experiment indicates how the algorithm would perform when distant points are not available. Real world situations where points at infinite are not available can appear especially in urban scenarios when making turns.

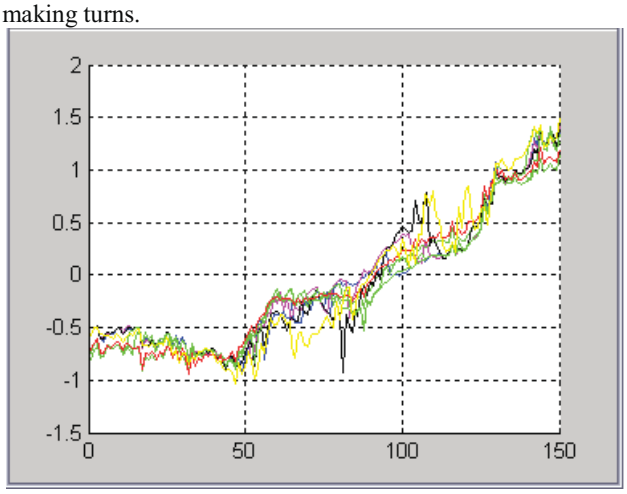

Figure 4. Yaw value in degrees (vertical axis) coputed for 150 frames (horisontal axis) and for different depth thresholds. Red indicates the yaw read from the vehicle. For the meaning of the other colors see Table 1

As expected, the error increases if the threshold decreases as shown in Table 1.

TABLE I. ESTIMATION ERROR FOR DIFFERENT DEPTHS THRESHOLDS

Distance	*Error*	*Color on graph*
Infinite	0.0503	Green
120m	0.1139	Green
80m	0.1369	Green
60m	0.1466	Blue
50m	0.1519	Magenta
40m	0.1666	Black
30m	0.2017	Yellow

Note that this numbers are strictly related to our camera setup and different intrinsic parameters may lead to different error values. The error will also depend on the quality of the feature matching, illumination changes, vehicle speed etc.

C. Comparison with stereo method results

We made a quick comparison with the results of the linear stereo method presented in [10]. Quantitatively it does not perform as good as the stereo method but the results are quite promising for a monocular setup.

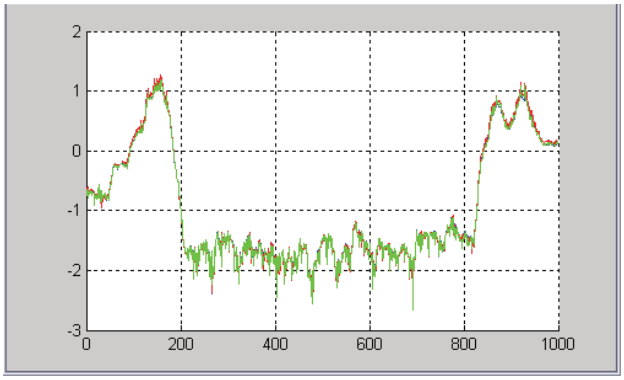

Figure 5. Coparison of yaw angle (degrees, horisontal axis) for 1000 frames (vertical axis). Yaw from vehicle is represented in red, the yaw computed with the stereo method in blue and the yaw computed with the proposed method in green.

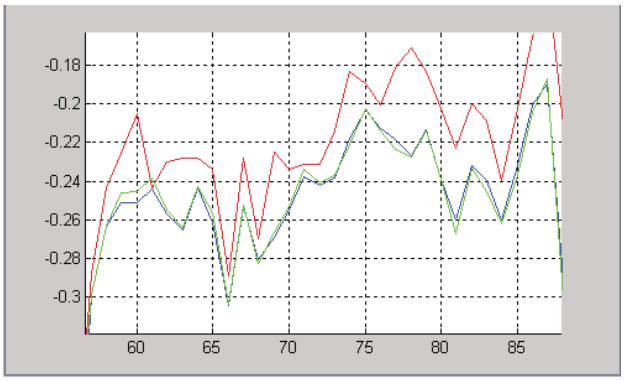

Figure 6. Zoom on Figure 5 with the same notations.

D. *Pitch estimation example*

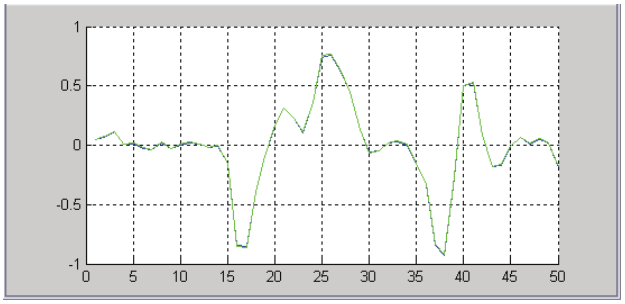

Figure 7. Pitch variation. Blue is the pitch coputed with the method presented in [10] and green is computed using the present method. The value is plotted in degrees.

Figure 7 shows the pitch variation when passing over the speed bump from the bottom left corner of the Figure 3.

V. Conclusion

We proposed a method for computing the rotation based on the distant points from monocular vision. Results show how the method determines the distant points based on RANSAC. Robustness and execution speed are increased by considering the fact that the image deformations for points at infinite are 0.

We also showed how the error depends on the distance. It immediately follows that the method only works with an acceptable error delta if we consider feature correspondences that are far enough from the camera.

Detecting the frames that don't have distant points represents one of the main future improvements for the method and many alternatives can be evaluated:

-based on deformations

-depth probability propagation for each pixel

-triangulation based on consecutive frames

In the context of the stereo ego motion estimation, it is very useful to select features close to the camera for translation computation, and features far away from the camera for improving rotation estimation. This kind of features classification will be evaluated for improving the existing point based stereo visual odometry methods.

Also, we believe that additionally to the stereo based visual odometry algorithms, by computing the rotation for both left and right images used in stereo we can detect and diagnose extreme error situations that can appear in real scenarios.

Acknowledgement

This paper is written within PAN-Robots project. PAN–Robots is funded by the European Commission, under the 7th Framework Programme Grant Agreement n. 314193. The partners of the consortium thank the European Commission for supporting the work of this project.

References

[1] Carlo Tomasi, Jianbo Shi, "Direction of heading from image deformations", CVPR 1993, pp. 422-427

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

[3] T.Y. Tian, C. Tomasi, D.J. Heeger, "Comparison of approaches to egomotion computation", CVPR 1996, pp. 315-320

[4] Steven Lovegrove, Andrew J. Davison, Javier Ibanez-Guzman, "Accurate Visual Odometry from a Rear Parking Camera", IV 2011, pp. 788 – 793

[5] Takahiro Azuma, Shigeki Sugimoto, Masatoshi Okutomi, "Egomotion Estimation Using Planar and Non-Planar Constraints", IV 2011, pp. 855-862

[6] D. Nistér, O. Naroditsky, and J. Bergen, Visual odometry, CVPR 2004, Volume 1, pp: 652-659.

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

[8] S. Nedevschi, C.Golban, C.Mitran, "Improving accuracy for Ego vehicle motion estimation using epipolar geometry", ITSC 2009, pp: 1-7.

[9] K. Yamaguchi, T. Kato, Y. Ninomiya, "Vehicle Ego-Motion Estimation and Moving Object Detection using a Monocular Camera", ICPR 2006, pp: 610 - 613.

[10] Catalin Golban, Sergiu Nedevschi, "Linear vs. non linear minimization in stereo visual odometry", IV 2011, pp. 888-894

[11] Istvan Szakats, Catalin Golban, Sergiu Nedevschi, "Fast vision based ego-motion estimation from stereo sequences — A GPU approach", ITSC 2011, pp. 538-543

[12] Catalin Golban, Istvan Szakats, Sergiu Nedevschi, "Stereo based visual odometry in difficult traffic scenes ". IV 2012

[13] D. Nister. An Efficient Solution to the Five-Point Relative Pose Problem, IEEE Conference on Computer Vision and Pattern Recognition, Volume 2, pp. 195-202, 2003.
[14] David Nistér, Oleg Naroditsky, and James Bergen, "Visual Odometry for Ground Vehicle Applications", Journal of Field Robotics, Volume 23, 2006.
[15] Ali Elqursh, Ahmed Elgammal, "Line-Based Relative Pose Estimation", CVPR 2011, pp. 3049-3056
[16] Jianbo Shi and Carlo Tomasi, "Good features to track", Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn., pp. 593-600, 1999.
[17] Bruce D. Lucas, Takeo Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", Proceedings of Imaging Understanding Workshop, 1981, pp. 121-130.
[18] Jean-Yves Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker. Description of the algorithm", Intel Corporation, 2000.
[19] C. Harris and M. Stephens. "A combined corner and edge detector", In Proc. Alvey Vision Conference, pages 147–151, 1988.
[20] Ramin Zabih, John Woodfill, "Non-parametric Local Transforms for Computing Visual Correspondence", European Conference on Computer Vision, 1994, pp. 151-158
[21] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints". International Journal of Computer Vision, 2004, pp 91–110.
[22] H. Bay, A. Ess, T. Tuytelaars, , and L. V. Gool, "Speeded-up robust features (surf)," Computer Vision and Image Understanding (CVIU), vol. 110, pp. 346–359, 2008.
[23] Charles V. Stewart, "Robust Parameter Estimation in Computer Vision", SIAM REVIEW, 1999, Vol. 41, No. 3, pp. 513–537