An Improved Binocular Visual Odometry for High-Speed Automotive Applications

Yu Huan, Chen Jiabin, Wang Liujun, Xie Ling, Song Chunlei, Wu Qinghe

School of Automation, Beijing Institute of Technology, Beijing 100081
E-mail: yuhuan_bit@126.com

Abstract: In this paper, we present an improved motion estimation method by adding extra information for binocular visual odometry (VO) which is especially suited for improving high-speed pose change estimation. The extra information is obtained by structured object detecting, taking lane line detection as an example. We can get an accurate position information by calculating the interval of each dotted lane line and counting the number of the dotted line which can be fused with the pose information obtained from visual odometry. The outlier rejection of the VO is also improved, making it adapt to highway situation. In the fusion process, a Kalman filter is adopted to estimate the motion and location information for a high speed vehicle. The experimental results show that the approach proposed is valid and can increase the positioning accuracy significantly compared with ordinary visual odometry.

Key Words: Visual Odometry, Lane Line Detection, Kalman Filter

1 INTRODUCTION

Vision system is one of the effective ways for human to perceive the external environment. Eighty percent of the external information is obtained by the visual system. In the field of intelligent robot, researchers launched a plenty of researches and they want robots to think like human. Our judgment of the spatial object position is mainly through the visual system, thus navigation and positioning technology based on computer vision has a very high research value. Nowadays, cameras are widely used for vehicle localization and navigation. The principle of visual odometry is to extract associated features and estimate the translation and rotation of the carrier in consecutive frames [1][2]. When GNSS signal is denied, binocular camera as passive sensor provides a wealth of information and can be a main sensor to substitute the GNSS. Compared with high-precision inertial navigation system, the cost of vision sensor lower. To identify features detected at different time and position is the main difficulty. Algorithms for visual odometry are generally based on features of stationary objects. Therefore, there are many studies on how to remove features stemming from moving objects and how to distinguish inliers and outliers [3][4][5]. The positioning accuracy of Camera based system is susceptive to heavily changing illumination, low brightness, and low frame rate. High-speed vehicles suffer most of those problems [3][6][7], Therefore, in this paper, we introduced an improved visual odometry for high-speed vehicle, which is based on structured object detection and improved outlier rejection.

In this paper, we first introduce the basic theory about binocular vision system and the algorithm used to realize visual odometry. Then the structured object detection and the way we used to get position information are introduced. The third section introduces how the additional information got from structured object is fused with visual odometry data. At the end of the paper, we introduce the localization experiment and result analysis.

2 BINOCULAR VISUAL LOCALIZATION

2.1 Geometrical Model of the Visual system

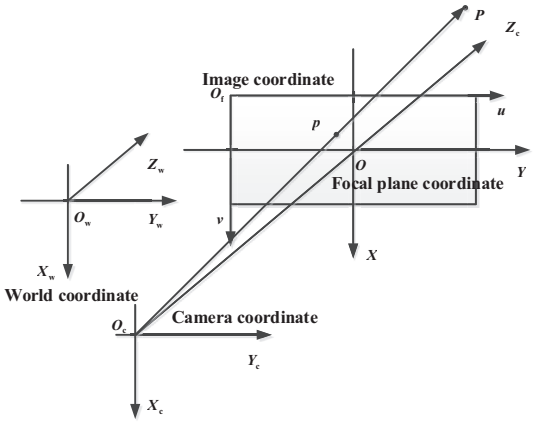


Fig 1. Pinhole model of a camera and coordinate relations.

Pinhole model is selected for the camera model and it is shown in Figure 1. Different coordinate frames are defined in this model. As is shown in Figure1, they are Image coordinate $O_f - uv$, Focal plane coordinate $O - xy$, camera coordinate system $O_c - X_c Y_c Z_c$ and world coordinate system $O_w - X_w Y_w Z_w$.

In binocular visual system, definition of coordinate system is the same as coordinates defined in Figure 1. The ideal binocular visual system is shown in Figure 2.

This work is supported by National Nature Science Foundation under Grant 91120003

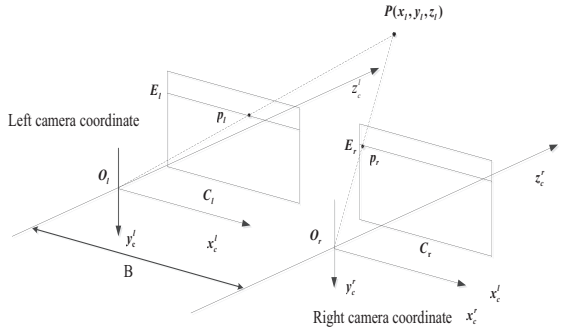


Fig 2. Ideal binocular visual system

In the ideal model, we suppose that the focal length of two cameras is equal and its optical axis is parallel to each other. The optical axis is perpendicular to the camera imaging plane, so the x axes of two camera coordinates are coincident and y axes are parallel. According to triangular principle, the coordinate of point P in Left camera coordinate system can be easily obtained.

$$X_l = \frac{B(u_l - u_0)}{u_l - u_r} = \frac{B(u_l - u_0)}{d} \tag{1}$$

$$Y_1 = \frac{Ba_x(v_l - v_0)}{u_l - u_r} = \frac{Ba_x(v_l - v_0)}{d} \tag{2}$$

$$Z_l = \frac{Ba_x}{u_l - u_r} = \frac{Ba_x}{d} \tag{3}$$

2.2 Motion Estimation Based on Improved RANSAC

There are two steps, feature detecting and matching, before motion estimation.

In the feature points detection process, SURF detector is selected for its good robustness, movement and rotation invariant. We can refer some papers about SURF detector. After the feature detection, Fast Library for Approximate Nearest Neighbors (FLANN) is employed for the first step feature points matching. Muja and Lowe first proposed FLANN algorithm in 2009. Since the images used for motion estimation have been corrected by epipolar, then through parallel epipolar constraint, the wrong matches could be removed and according to the uniqueness of feature matching, if one point corresponds to multi points, this point also should be ignored.

In visual positioning system, the reference frame is selected as the initial state of the left camera coordinate system. The vehicle's motion state of the adjacent time can be described as Figure 3. $\{q_t^i \leftrightarrow q_{t+1}^i\}, i = 1:N$ represents the feature point set, $t+1$ denotes current moment. According to Equation (1), (2) and (3), the spatial coordinates of the corresponding feature point is defined as $\{P_t^i \leftrightarrow P_{t+1}^i\}, i = 1:N$. Fig.6 shows the change of the space position of a space point P . P_k represents the coordinate at previous moment, P_{k+1} represents the coordinate in current moment. The transformation is

$$P_{k+1}^i = RP_k^i + t \tag{4}$$

Where P_{k+1}^i and P_k^i represents the spatial coordinate of the i-th feature point at the present time and previous time, respectively.

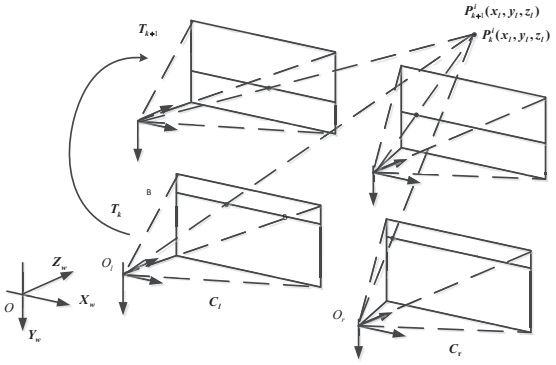


Fig 3. Motion estimation model of a running car.

In the actual scene, there are usually a lot of dynamic objects, so in visual localization algorithm how to remove feature points of dynamic object and other false matching points is key to improve positioning accuracy. In this paper, we employ a method based on RANSAC. The flow chart of the algorithm is described in Figure 4.

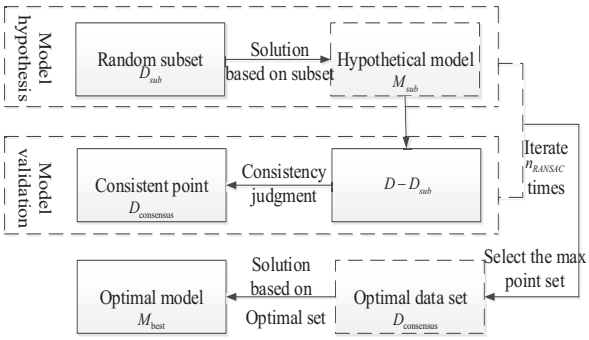


Fig 4. Flow chart of RANSAC-based outlier rejection scheme.

Iteration times can be determined by the following equation:

$$n_{RANSAC} = \frac{\log(1-p)}{\log\left(1-(1-o)^s\right)} \tag{5}$$

Here, s is the minimum feature points needed for calculation. p denotes the probability that one or more sample contains inliers and ε is the assumed percentage of outliers in the data. We get all inliers using Euclidean re-projection error. When the Euclidean re-projection error is lower than a certain threshold, a feature is an inlier. Generally, the threshold is a constant value, while in this paper the value is changed according to the number of feature points and the speed of the car. To get the final ego-motion estimation, we use all inliers of the best sample after the iterative process.

3 STRUCTURED OBJECT DETECTION

When we are driving on the highway, there are many objects we can use to make driving decisions, such as lane

lines, traffic signs, etc. However, in most cases, we do not realize that these structured objects can be used to obtain our position. In this section, we propose a structured object detection method which includes image preprocessing, lane line detection and structured object detection.

3.1 Image preprocessing

Image preprocessing is an important part of image recognition. In high-speed scenarios, real-time is an important aspect that affects the performance of the control. In most cases, the lower part of the image is the road region. To improve the real-time performance, we only process the lower part of the image, and d is determined by different situations, as shown in Figure 4. We can directly do the recognition work in the ROI, which is faster compared to search for the whole image.

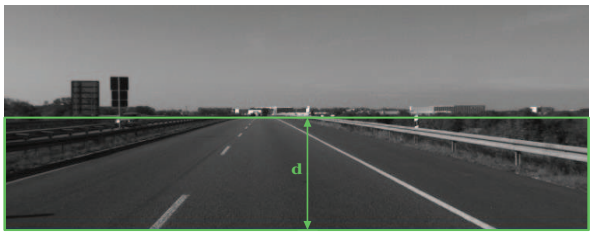


Fig 4. The ROI (region of interest).

The image preprocessing steps include: set the ROI, graying, median blur, image binaryzation. Due to the limited length of the paper, we skipped the first three steps. In image preprocessing, the most important thing is to find the appropriate threshold and separate the object from the background. We propose an improved image binaryzation method. Firstly, using Otsu method to process an image [9], find the optimal threshold. Suppose that the pixels in a gray image be represented in m gray levels (1,2, . . . , m). Let n_i denote the number of pixels at level i , and the total number of pixels $N = \sum_{i=1}^{m} n_i$ .The probability of occurrence of level i is $p_i = n_i / N$, using a threshold T separate the object class C_0 {1,..., T }and the background class C_1 { $T+1$,..., m }. The cumulative probabilities and the mean levels of class C_0 are

$$P_0 = \sum_{i=1}^{T} p_i \tag{6}$$

$$\mu_0 = \sum_{i=1}^{T} \frac{ip_i}{P_0} = \frac{1}{P_0} \sum_{i=1}^{T} ip_i \tag{7}$$

the cumulative probabilities and the mean levels of class C_1 are

$$P_1 = \sum_{i=T+1}^{m} p_i = 1 - P_0 \tag{8}$$

$$\mu_1 = \sum_{i=T+1}^{m} \frac{i \cdot p_i}{P_1} = \frac{1}{P_1} \sum_{i=T+1}^{m} i \cdot p_i \tag{9}$$

The mean level of the total image is

$$\mu = \sum_{i=1}^{m} ip_i = P_0\mu_0 + P_1\mu_1 \tag{10}$$

The between-class variance is

$$\sigma^2(T) = P_0(\mu_0 - \mu)^2 + P_1(\mu_1 - \mu)^2 \tag{11}$$

Thus The optimal threshold is decided by maximizing the between-class variance

$$T^* = \arg\max_{1 \le T \le m} \{\sigma^2(T)\} \tag{12}$$

Secondly, because of the continuity of video frames, the difference between adjacent frames δ is very small. And in most cases, the middle lower part of the image is the road region. Set a ROI box in the middle lower part and figure out the mean value of road region μ_r . If $\delta \ge \delta_{\max}$ ($\delta_{\max}$ is determined by the experiment) or the threshold is smaller than μ_r , we can simply use the threshold of last frame T_p as the threshold of current frame T_c . Finally, thresholding the image and statistical white point probability p_w , if $|p_w| \le |p_{wlast} \pm \varepsilon|$, save the threshold of current frame T_c as T_p for recycle. This method performs well to solve the fails of thresholding of some individual frames. The result binary image is shown in Figure 5.

Fig 5. The result binary image.

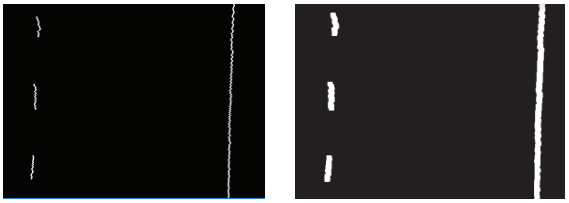
3.2 Lane Line Detection

The lane line detection steps include: perspective transform, Morphological filtering and Hough line detection.

In the image coordinate, the parallel lane line of the world coordinate caused a great deformation. The perspective transformation relation shows in Figure 1. To remove the perspective effect [10][11], we need do perspective transformation to generate a new image. In this image lane lines can be devised as almost vertical bright lines of constant width. The perspective transformation between two kinds of view can be uniquely determined by four pairs of points in the binary image and the new image. The new image is shown in Figure 6.

Fig 6. After the removal of the perspective effect.

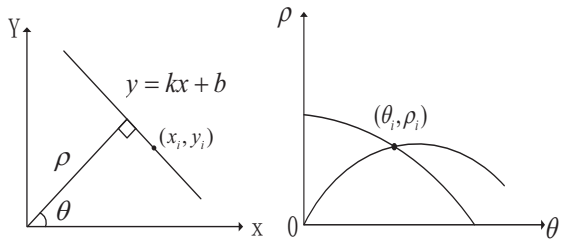
To smooth the lane line edges and make Hough transformation faster, we first calculate the midpoint of white lane line pixels respectively, then use morphological filter to dilate and smooth the lane lines [12][13]. The results are shown in Figure 7.



(a) (b)

Fig 7. (a) the midpoints of lane line pixels; (b) Morphological operation.

The Hough transform is most commonly used for the detection of lane lines. Consider the point (x_i, y_i) in the image space belongs to straight line $y = kx + b$, it can be denoted as $y = (-\cos\theta/\sin\theta)x + (\rho/\sin\theta)$, where ρ is length of a normal from the origin to this line and θ is the orientation of ρ with respect to the x-axis. Reduce it we can get $\rho = x\cos\theta + y\sin\theta$. It indicate that the point (x_i, y_i) map to sinusoids curve in the polar Hough parameter space. The intersection point (θ_i, ρ_i) in the polar space means the Corresponding points (x_i, y_i) in the image space are in the same straight line (see Figure 8(a) and 8(b)). There are many kinds of improved Hough transform method, among them, the probabilistic Hough transform (PPHT) performs well. The performance is often only slightly impaired, thus the execution time can be considerably shortened [14][15]. The detected lane line segments are shown in Figure 9(a).



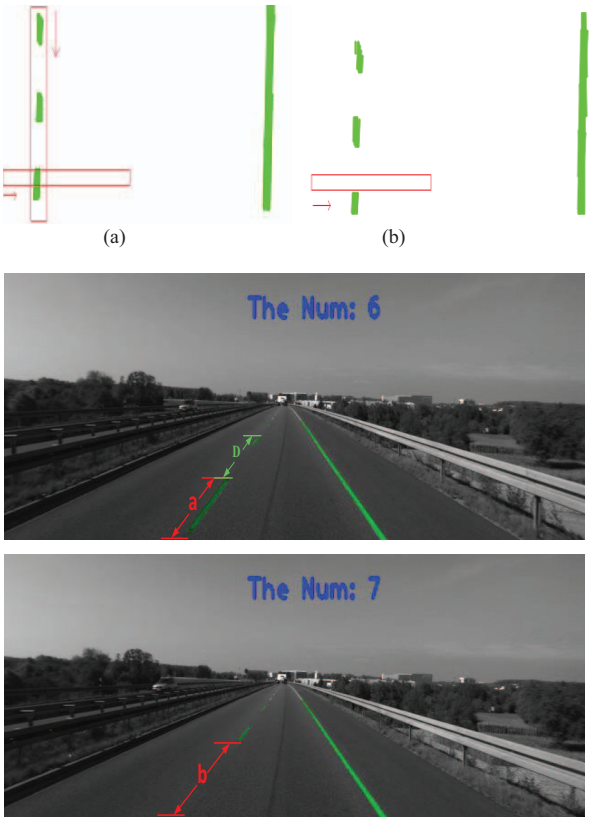
(a) (b)

Fig 8. (a) the image space; (b) the polar space.

3.3 Structured Object Detection

On the highway, there are many structured objects that can be used as visual feature detection. The lane lines are drawn regularly, thus we can count the numbers of passed lane lines to know how far have we go and where we are.

Firstly, we set a series of column search boxes in the picture to find the dotted lane line feature (see Figure 9(a)). Secondly, set a row search box in the corresponding area, if the number of lane line pixels is enough, it indicates that we are passing through a dotted lane line segment (see Figure 9(a)). otherwise, it means that we have already passed or have not passed a dotted lane line segment (see Figure 9(b)). Search for the previous frame and the current frame we can know whether we have passed a dotted lane line segment or not. Then, by counting the number of the passed lane line segment, we can know how far have we go. For example, in Figure 9(c), if the we count one unit, including a blank area and a solid line are, the displacement is D, then according to the calculation described in section 2.1, the position for The Num: 6 image is $6*D+(D-a)$ and the position for The Num:7 image is $7*D+(D-b)$.



(a) (b)

(c)

Fig 9. (a) the previous frame; (b) the current frame; (c) the final result.

4 IMPROVED VISUAL ODOMETRY

To fuse the motion estimation data obtained from visual odometry and structured object detection, a Kalman filter is applied[16][17]. The flow chart of the improved visual odometry is shown in Figure 10. The image data procedure is divided into two part. If visual data is valid, the visual odometry part will calculate all the time. If structured object detected, the additional information will be confused with visual odometry. The procedure of position estimation by structured object is like a wheel odometry, which is described in the last section.

To estimate the motion parameters, we place a stand Kalman filter, which is defined as the following equations:

$$X_k = FX_{k-1} + \varepsilon \tag{13}$$

$$Z_k = H_k X_k + \nu \tag{14}$$

The velocity vector is defined as $V = (R\ \mathrm{t})^T / \Delta t$, R and t are the rotation and transformation matrix, Δt is the time between image frames. If a constant acceleration is assumed, then the matrix in the Kalman filter will be achieved.

$$X_k = (V_k\ \mathrm{a}_k) \tag{15}$$

$$Z_k = {}^{1}\!/\!_{\Delta_t} (R\ \mathrm{t})^T \tag{16}$$

ε, ν represent Gaussian process and measurement noise, respectively.

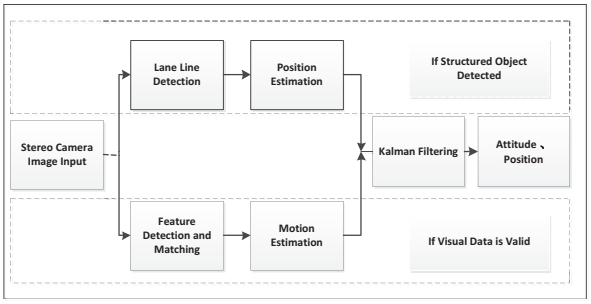


Fig 10. The flow chart of improved visual odometry.

EXPERIMENTAL RESULT

In this section, we evaluate the proposed approaches by an off-the-shelf dataset from KITTI Vision Benchmark Suit, which provides dataset with ground truth and collected data from GPS, Gyroscope and Stereo camera with a resolution of 1344x391 and 10 frames/s [18]. Sequence 01 is selected, and the car in this dataset is driving on the highway with high speed, including 1170 frames, and the driving time is about 2 minutes. Figure 11 shows the experiment position and trajectory in a Google Map. The positon of the star represents the starting point, and the red line denotes the moving path. The image of this place

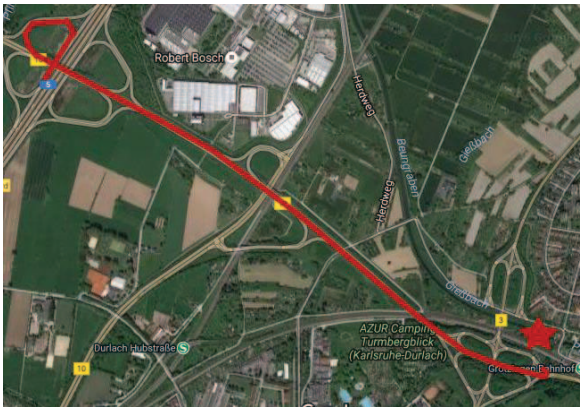


Fig 11. Experimental area on Google Map.

We firstly analyzed the feature points detected and matched, and the number of inliers selected according to the method introduced section 2.2. The results for sequence 00 are shown in Figure 12. Blue line denotes the number of matched feature points between consecutive frames and the red represents the number of inliers selected. Compared with campus and residential dataset, the number of points in the figure are very small.

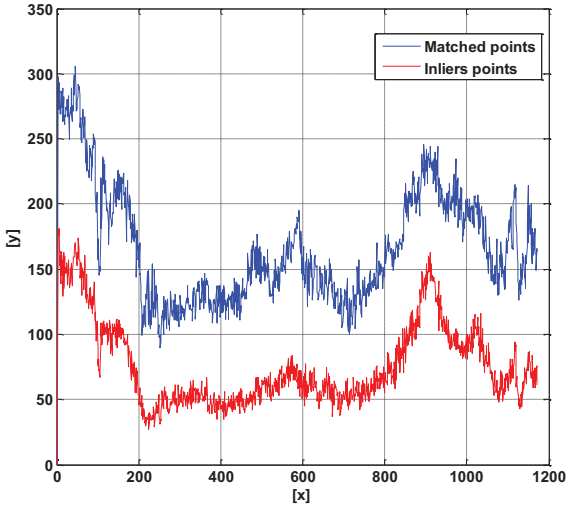


Fig 12. Feature points detection results.

when a car is driving on a highway, we can not detect lane line all the time. So only if lane line are detected, extra information can be calculated. In sequence 00, there are several sections can be useful. The valid sections are Frame 46 to 156, Frame 214 to 872, and Frame 1114 to 1169. The localization result is represented in Figure 12. The curves in Figure 13 are very similar with the red line in Figure 11.

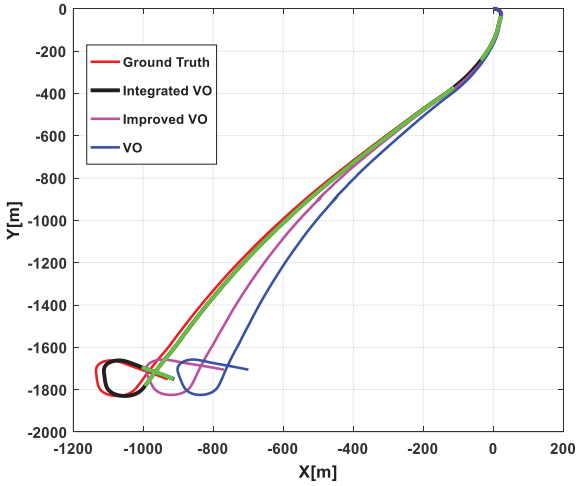


Fig 13. Localization results of different algorithms.

In the results figure, the red line denotes the Ground Truth, which is provided by GPS and is the reference line. The magenta line represents an improved VO, which uses the improved RANSAC described in section 2.2, and the blue

line is obtained by ordinary RANSAC method. The line composed of black and green is calculated by fusing the position data obtained from lane line detection and improved VO. When lane line is detected, the trajectory is the green part, and when it is denied, the trajectory is the black part.

If the positioning accuracy is assessed by the position error between the calculated and truth. The best positioning error is 12.3 m obtained from the integrated VO algorithm proposed in this paper.

5 CONLUSION

The experiment result denotes that if a car is running on the highway, like the scene in Figure 9, ordinary visual odometry does not work efficiently. However, if extra information could be added to VO, the positioning accuracy will be improved remarkably. In this paper, we only detect the lane line, but in a real scenario there are more structured objects that could be extracted, such as the trees, static cars on the roadside and so forth. Therefore the method proposed has great engineering and practical significance.

REFERENCES

[1] Persson, Mikael, et al. "Robust stereo visual odometry from monocular techniques." 2015 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2015.

[2] Badino, Hernán, Akihiro Yamamoto, and Takeo Kanade. "Visual odometry by multi-frame feature integration." Proceedings of the IEEE International Conference on Computer Vision Workshops. 2013.

[3] Buczko, Martin, and Volker Willert. "How to distinguish inliers from outliers in visual odometry for high-speed automotive applications." Intelligent Vehicles Symposium (IV), 2016 IEEE. IEEE, 2016.

[4] Cvišić, Igor, and Ivan Petrović. "Stereo odometry based on careful feature selection and tracking." Mobile Robots (ECMR), 2015 European Conference on. IEEE, 2015.

[5] Krešo, Ivan, and Siniša Šegvic. "Improving the Egomotion Estimation by Correcting the Calibration Bias." 10th International Conference on Computer Vision Theory and Applications. 2015.

[6] Kitt, Bernd, Andreas Geiger, and Henning Lategahn. "Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme."Intelligent Vehicles Symposium. 2010.

[7] Deigmoeller, Joerg, and Julian Eggert. "Stereo Visual Odometry Without Temporal Filtering." German Conference on Pattern Recognition. Springer International Publishing, 2016.

[8] Chen, Chenyi, et al. "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving." (2015):2722-2730.

[9] Xu, Xiangyang, et al. "Characteristic analysis of Otsu threshold and its applications." Pattern Recognition Letters 32.7(2011):956-961.

[10] Liu, Ziqiong, S. Wang, and X. Ding. ROI perspective transform based road marking detection and recognition. 2012.

[11] Broggi, Alberto, M. Bertozzi, and A. Fascioli. "Architectural Issues on Vision-Based Automatic Vehicle Guidance." Real-Time Imaging6.4(2000):313-324.

[12] Serra, Jean, and L. Vincent. "An overview of morphological filtering."Circuits, Systems, and Signal Processing 11.1(1992):47-108.

[13] Soille, Pierre, and H. Talbot. "Directional morphological filtering." IEEE Transactions on Pattern Analysis & Machine Intelligence 23.11(2001):1313-1329.

[14] Kiryati, N., Y. Eldar, and A. M. Bruckstein. "A probabilistic Hough transform." Pattern Recognition 24.4(1991):303-316.

[15] Matas, J., C. Galambos, and J. Kittler. "Robust Detection of Lines Using the Progressive Probabilistic Hough Transform." Computer Vision & Image Understanding 78.1(2000):119-137.

[16] Tardif, Jean-Philippe, et al. "A new approach to vision-aided inertial navigation." Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. IEEE, 2010.

[17] Huai, Jianzhu, C. K. Toth, and D. A. Grejner-Brzezinska. "Stereo-inertial odometry using nonlinear optimization." International Technical Meeting of the Satellite Division of the Institute of Navigation 2015.

[18] Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.