A Lane Detection Method based on Track Management Approach

Shasha Dong
The Engineer Warfighting Lab
The Engineer Academy of the People's Liberation Army
Xu Zhou City, China
efan1986@163.com

Cheng Peng
The Engineer Warfighting Lab
The Engineer Academy of the People's Liberation Army
Xu Zhou City, China
278456400@qq.com

Abstract—**The lane detection is important for the lane departure warning (LDW) used in advanced driver assistance systems (ADAS). Several approaches for lane detection were suggested in the past. However, there is still one issue about robustness. This paper presents a robust lane detection method based on a two layer Kalman filter. The key idea is to apply methods from the target tracking domain to identify lanes in the image space. We assume each road lane could be seen as one track in 2D coordinates and then use a Kalman filter to update the tracks. Using information from consecutive frames, it is much easy to distinguish between real lane pixels and noise in the image. Our method is based on two phases: an image preprocessing phase to extract areas that potentially represent markings and a tracking phase to identify lanes. In the tracking phase we use two special Kalman filters to estimate tracks according to the track management in spatial dimension.**

The simulation results show that this method exhibits good robustness under various scenarios and meet the real-time requirement.

Keywords—lane detection; robustness; two layer Kalman filter

I. INTRODUCTION

Within the past few years, active safety systems were increasingly introduced into cars to lower the number of accidents. The U.S.Department of Transportation reported 33,963 fatalities in the year of 2009; 59% of all traffic accidents were caused by lane departure. Therefore, road lane detection technology became a hot issue in the present intelligent vehicle research [1]. Therefore, as a key part of advance driver assistance system, more and more researchers pay attention to lane detection methods.

Several algorithms for lane detection were introduced. These can be grouped into two categories: Parameter space transformation based methods and model based methods. Since both these two methods have their own limits, it is important for us to find a more robust method which can keep high performance under variety of environmental conditions.

Parameter space transformation is the most popular method in the lane detection field. First, the image is transformed into a gray image to get the edges of the road lane markings. Second, people use the Hough transformation approach [2] to obtain the straight lane information from the edge image. However, the issue of this method is that the performance about the Hough transformation. As a matter of fact, the algorithms based on Hough transformation are typically only able to detect straight lanes.

Model based methods can also be grouped into two categories. One group is to predefine a set of curvatures as candidate lane models and select the best fitting model per lane [3]. Although this method can overcome the universal lane detection problems due to inaccuracies in edge detection such as shadow of trees or passengers on the road, it still has its limits. It is not possible to combine these models for one lane which means that the algorithm can't get the optimal lane model. The other method is to predefine the road model and then use the Kalman filter to estimate the state [4], [5], [6]. This method assumes that the lane has constant width on a flat plane and works well in most cases. However, it doesn't concern about the light condition and the occlusion. Furthermore, this method requires more parameters than others such as vehicle's gyro information.

This paper uses a tracking approach known from the object tracking domain [7] and considers the image data as the only input, which is much easier to set up the evaluate platform than other methods using vehicle's Gyro information [8] . The contribution of our approach is that it is able to overcome the influence of shadows and other noise in the image.

There are two stages of our method: a preprocessing phase and a tracking phase for lane identification. The first is the preprocessing phase. In this stage the image is cropped to extract the region of interest in the first step. In the next step, the image is processed by a median filter to reduce noise and retain the details. Then Otsu's algorithm is used to identify areas that potentially represent markings. Finally, image erosion is used to remove outliers.

The tracking phase is performed in spatial dimension. In this stage the algorithm tracks lanes by using the information of consecutive frames. Lanes can be grouped into two categories: temporary tracks and confirmed tracks. Different Kalman filters are used for different kind of tracks.

The algorithm is evaluated using test data from both image database [9], [10]. The experiment results indicate that the proposed method yields precise lane marking information from video sequences and still images.

This paper is structured as follows: Sec. II briefly describes the image processing stage. Sec. III introduces more details about the tracking phase. Sec. IV presents experimental results under different scenes. Finally, the paper is concluded in Sec.V.

II. IMAGE PREPROCESSING PHASE

The suggested approach consists of two phases: the pre-processing phase and the tracking phase. This section gives insights into the preprocessing phase, while the next section will elaborate on the tracking phase.

This section uses image processing methods known from computer vision. The goal of this phase is to improve the estimate precision in tracking phase. In most scenes, the road region and non-road region have obvious boundaries, and the road region is mainly in the lower part of the image [11]. Consequently, we identify the lower part of the image as the region of interest (ROI). In this way, we can satisfy the efficiency and feasibility of the detection method. As shown in Fig. 1, the image is divided vertically into two parts with a fixed ratio (here 1/4 height).

The image is then processed by a median filter to retain the details and remove the noise. Otsu's algorithm is used to calculate an adaptive threshold to segment the image into marking and non-marking areas [12]. Fig.2 shows the segmentation result after the application of Otsu's algorithm.

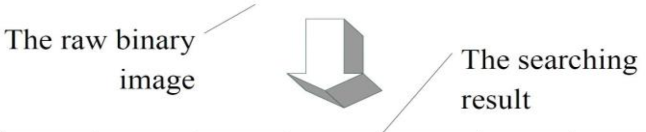
Fig. 1. Application of region of interest (ROI)

Fig. 2. Image after median filtering and application of Otsu's algorithm

After the marking region has been extracted, we apply image erosion to suppress outliers. The lane boundaries are extracted as follows: we start searching for lane pixels at the bottom row, left to right. After the current row is processed, we move on to the next row from bottom to top, line by line.

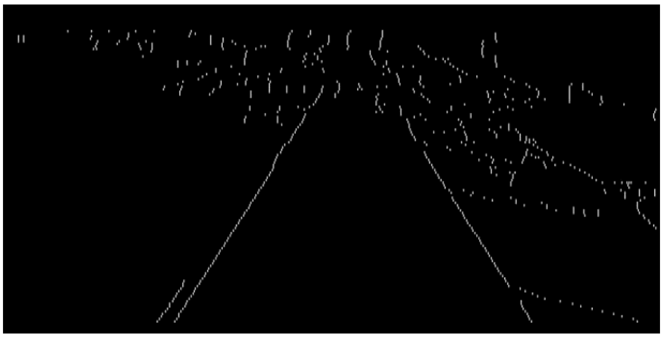
A pixel is identified as a lane pixel if it is part of a set of at least three horizontally connected pixels. The middlemost pixel of each set is recorded as a lane pixel. Fig. 3 illustrates the details of our method.

r/c	1	2	3	4	5	6
4	0	0	1	1	1	0
3	0	1	1	1	1	0
2	0	1	1	1	0	0
1	1	1	1	0	0	0



r/c	1	2	3	4	5	6
4	0	0	0	1	0	0
3	0	0	1	0	0	0
2	0	0	1	0	0	0
1	0	1	0	0	0	0

(a) The method used to identify lane pixels



(b) Result of image preprocessing

Fig. 3. Application of image erosion

III. TRACKING PHASE

A target tracking method is used to distinguish between noise and lane pixels. The tracking algorithm is based on a 2D Cartesian coordinate system. The image is processed row by row and pixels are tracked between the rows. We classify the pixels as noise, as temporary tracks and as confirm tracks. Note that a temporary track will change to confirm track only if it meets certain requirements, which are discussed later.

The tracking method is based on a two layer Kalman filter. The first layer filter is used for temporary tracks and the second is used for confirmed tracks.

We use the first layer Kalman filter to estimate the position and velocity of the pixels in X direction. The X coordinate of the pixels are considered as measurements for this filter. The reason why we don't concern about the position and velocity in Y direction is that the tracking process moves on from bottom to top, line by line. In this case, the position in Y direction is fixed and the related velocity is always equal 1. The second layer Kalman filter is used to estimate the parameters of a quadratic model for each confirmed lane track.

In the following, the tracking phase is explained in more details.

A. Temporary Tracks

Temporary track is estimated by the first layer Kalman filter which can be described as Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in

$$\mathrm{X}_{k+1} = F_k \mathrm{X}_k + \Gamma_k w_k \tag{1}$$

$$z_{k+1} = H_{k+1} \mathrm{X}_{k+1} + v_{k+1} \tag{2}$$

with time index k and state vector x = (X, X˙)T describing the position X and velocity X in X direction in 2D Cartesian Coordinates; F is the state transition matrix and Γ is the process noise gain matrix. The process noise w_k is described by a zero mean, normal distributed random process with covariance Q.

The measurement is given by the pixel coordinates and modeled with (2), where H is the observation model and vk refers to the zero mean, normal distributed measurement noise with covariance R.

We describe the target motion as a 1D model which represents the trajectory of the target in X direction. The model is given by

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad \Gamma = \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & T \end{bmatrix} \tag{3}$$

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

where T denotes a discrete sampling period. In the case of our algorithm, the sample period reflects the steps of row-wise calculation.

The more detailed and generic description of the tracking phase for temporary tracks is given by [7].

B. Track Management

Based on the pixel velocity and position associated with a temporary track, the tracking process records each temporary track's trajectory in each frame. Using the trajectory information, the mathematical model of each lane can be considered as (4) in 2D Cartesian coordinate [13] and can also be estimated by using RANdom SAmple Consensus (RANSAC) [14], [15] followed by Least Square method [16].

$$y=ax^2+bx+c \tag{4}$$

RANSAC is a robust estimation technique based on the principle of hypotheses generation and verification. The generic RANSAC algorithm robustly fits a model through the most probable data set while rejecting outliers. Least Squares Estimation (LSE) is then used to fit to a quadratic model.

Since the frames are consecutive images, we could also assume that the differences of quadratic parameters for the same lane are negligible. However, the road signs and the noise don't have this feature. Based on this assumption it is much easier to distinguish road sign pixels and real lane pixels. We record each temporary track's quadratic parameter at frame k and then calculate the distance m with the other temporary track's quadratic functions at frame $k+1\left(m = \int \left(y_k(x) - y_{k+1}(x)\right)\right)$.

The algorithm calculate all the temporary tracks in each frame and find the shortest distance as the best fitted track. For the best fitted track if the distance is below a certain threshold in consecutive 5 frames, it is classified as a confirmed track. If not, then the algorithm will delete these temporary tracks. The work is done in the track management, which detects each temporary track in each frame in order to find the potential confirmed track.

C. Confirmed Tracks

Once the track management method changes a temporary track to a confirmed track, the second layer Kalman filter is employed to refine the estimation of the quadratic function parameters and the filter model are given by

$$A_{k+1} = BA_k + b_k \tag{5}$$

$$C_{k+1} = D_{k+1} A_{k+1} + u_{k+1} \tag{6}$$

with

$$A = \begin{bmatrix} a \\ \dot{a} \\ b \\ \dot{b} \\ c \\ \dot{c} \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{8}$$

Where the state vector A describe the quadratic parameters and the first order derivatives of (4), bk and uk describe a zero mean, normal distributed white noise.

From fig. 4 we can see that a lane boundary position (X, Y) is searched according to the predicted quadratic parameter state. Real lane pixels are associated with the ROI which uses a certain size of a gate in perpendicular direction to the predicted lane. If the Euclidean distance of the value

$d = \|(X, Y) - (Xpre, Ypre)\|$ is below a certain threshold, (X, Y) is added to the set of valid points for each lane model. The pixel (X, Y) is the real lane pixel in the image and (Xpre , Ypre) is the predicted pixel according the second layer Kalman filter.

The measurement in (6) is given by the least squares estimate of associated pixels from (4). To reduce the complexity of this association phase, pixels are removed from the ROI after being added to the set of valid points.

If no pixel is associated with a confirmed track within 2 seconds, the confirmed track is discarded.

D. The Tracking Phase

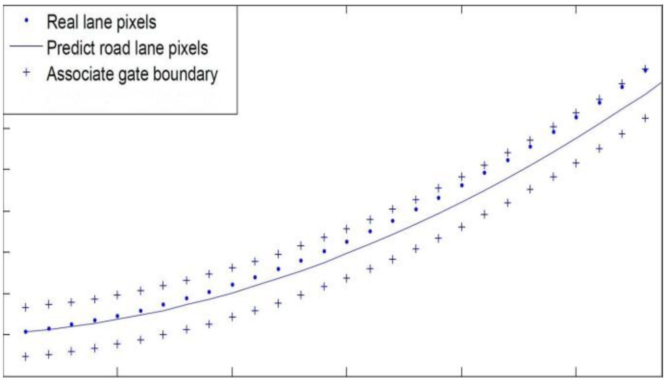


Fig. 4. Application of image erosion

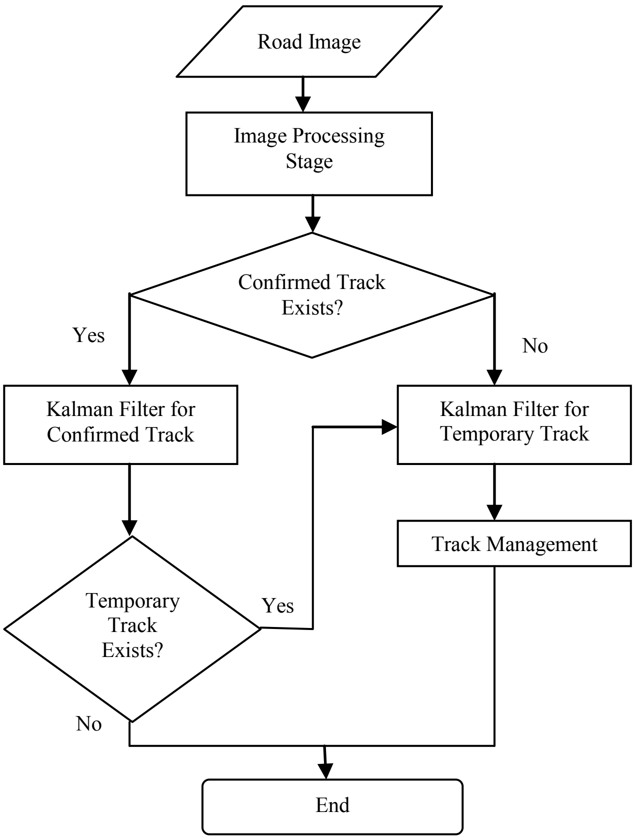


Fig. 5. Data flow of the algorithm

For each consecutive frame, if the confirmed track existed after the image processing phase, the algorithm would first use the second layer Kalman filter to estimate the state of the quadratic function parameters. Then it will delete the related pixels from the image. After doing that, the algorithm uses the first layer Kalman filter to estimate the temporary tracks.

Once the temporary tracks meet the requirements, the track management add them to the set of confirmed tracks.

Fig. 5 shows the flow of our algorithm for one cycle.

IV. EXPERIMENTAL RESULT

Experiments are performed on consecutive frames and still frames, which show the performance of our approach. The consecutive frames include a lot of road signs and other noise such as shadows and bent lanes.

The test data is taken from the CMU Image Database [9] and UIUC Video Database [10]. All video sequences in this work were obtained at 15 frames per second, with a resolution of 480×640 pixels. The average processing time in image preprocessing phase is 70ms per frame and in tracking phase is 15ms per frame (I ntel(R)E8400, 3.0Ghz, 4GRAM). The performance of the algorithm is shown in table I.

TABLE I. PERFORMANCE OF THE ALGORITHM

Video	DR	FR
Unstructured Urban Road	91.35%	5.83%
Structured Highway	98%	0

DR denotes the successful lane detection rate and FR is the false detection rate. Lane markings on the ground are manually marked and recorded in the database. The criteria of successful detection are based on the pixel distance between real lane pixels and the algorithm detection result in one image.

$$DR = \frac{NTP}{NGL}, \mathrm{FR} = 1 - \frac{NTP}{NDR} \quad (9)$$

NGL represents the number of lane markings appearing in the evaluation set. NTP is the number of road lanes which are detected successfully. NDR is the number for the detection result outputted by our algorithm.

A. Results in individual frames for temporary tracks

Fig. 6 shows the results of our algorithm for tracks in individual frames.

These two tracks are considered as temporary tracks and tracked by using the first layer Kalman filter.

As expected, the first layer Kalman filter uses a 1D model can represent the target state in X direction. Fig. 7 illustrates the process of the estimation along the rows for synthetic lane. Fig. 7b and 7c illustrates the position and the velocity of the target along the rows. From Fig. 7a and 7c we can see that while the road lane is straight, the velocity of the target is stable. Once the velocity changes obviously there must be a curve on the road lane.

This feature could also be used to calculate the curvature and the curvature rate by our algorithm in future works.

B. Results in consecutive frames for confirmed tracks

Fig.8 shows the result on binary images. Fig. 8a illustrates the predicted road lane and the estimated road lane. In this figure, red pixels are the estimated road lane and blue pixels are the predicted road lane, white pixels are the result from the image processing phase which including the real lane pixels and noise. Fig. 8b is a local area zoom from fig. 8a which illustrates after the Kalman filtering the estimated lane fits well with the real road pixels.

Fig. 9 shows that the algorithm successfully detects road lanes even within complex scenes.

Fig. 9a illustrates that there are no confirmed tracks in the system. Since there is a big gap on the left lane in Fig. 9a, the right lane changes to confirmed track first. Once the quadratic function of the temporary track doesn't change too much in consecutive frames, the system adds the temporary track to the set of confirmed tracks which can be seen in Fig. 9c. From Fig. 9d we can see that when the vehicle changes lane, the system can not detect the middle lane. There were no real lane pixels in the associated gates and the second layer Kalman filter could not work.

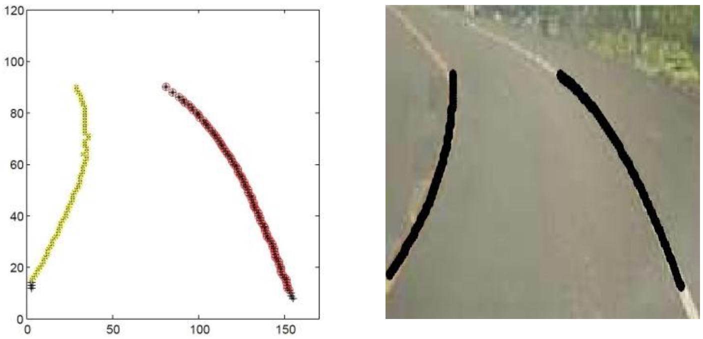
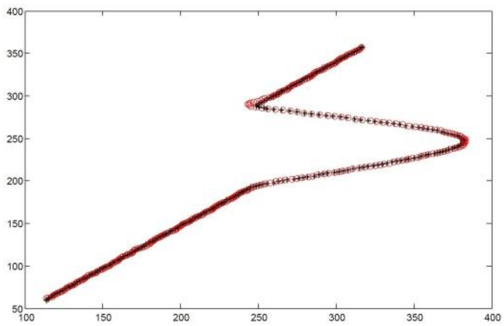
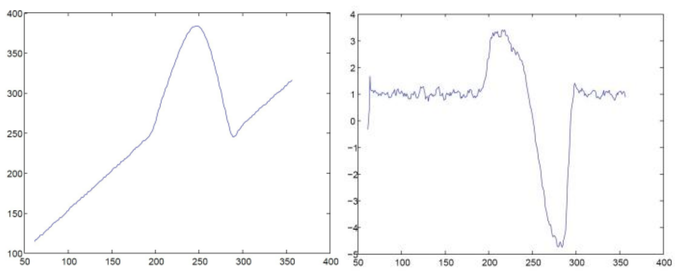


Fig. 6. Lane detection results in single frame

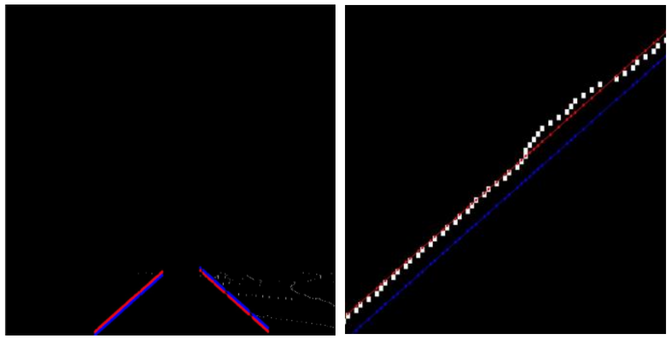


(a) Synthetic lane Model



(b) Position along the rows (c) Velocity along the rows

Fig. 7. Target velocity along the measurement sequence



(a) (b)

Fig. 8. Lane detection results on binary image from image processing phase

(a) (b)

(c) (d)

Fig. 9. Lane detection results in consecutive frames

The results of our experiments indicate that the algorithm performs robust when there are non-lane markings, vehicles and shadows on the road.

V. CONCLUSION

It is very hard to detect the lane robustly under different scenes. The varying road signs, shadows and clutters from other objects cause the results erring from the real status. In this paper, an approach of lane detection is presented. In comparison to the related works, the algorithm only needs the image data to detect the road lanes from the video stream. The evaluation results show that the algorithm achieves high accuracy and robustness under different scenes and meets real time requirements. This paper used two layer Kalman filters to estimate the temporary lane tracks and the confirmed tracks. The benefit of this approach is the possibility to distinguish the clutters and the real lane pixels from the image, which is also the key issue of the lane detection field.

Future improvements of the algorithm should include other sensors such as GPS, laser and Gyro sensors to improve the estimation precision and robustness.

REFERENCES

[1] X.Dai, A. Kummert, S. B. Park, and D. Neisius, "A warning algorithm for lane departure warning system," in Intelligent vehicles Symposium, 2009 IEEE, 2009, pp. 431 –435.

[2] A. Saudi, J. Teo, M. H. A. Hijazi, and J. Sulaiman, "Fast lane detection with randomized hough transform," in Information Technology, 2008. ITSim 2008. International Symposium on, vol. 4, 2008, pp. 1 –5.

[3] Z. Kim, "Robust lane detection and tracking in challenging scenarios," Intelligent Transportation Systems, IEEE Transactions on, vol. 9, no. 1, pp. 16 –26, 2008.

[4] A. Takahashi and Y. Ninomiya, "Model-based lane recognition," in Intelligent Vehicles Symposium, 1996., Proceedings of the 1996 IEEE, Sept. 1996, pp. 201 –206.

[5] M. Bertozzi and A. Broggi, "Gold: a parallel real-time stereo vision system for generic obstacle and lane detection," Image Processing, IEEE Transactions on, vol. 7, no. 1, pp. 62 –81, Jan. 1998.

[6] P. Coulombeau and C. Laurgeau, "Vehicle yaw, pitch, roll and 3d lane shape recovery by vision," in Intelligent Vehicle Symposium, 2002. IEEE, vol. 2, june 2002, pp. 619 – 625 vol.2.

[7] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, Estimation with application to tracking and navigation. Wiley, 2001.

[8] J. McCall and M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," Intelligent Transportation Systems, IEEE Transactions on, vol. 7, no. 1, pp. 20 –37, 2006.

[9] CMU/VASC Image Database, "http://vasc.ri.cmu.edu/idb/html/road".

[10] UIUC Image Video Database, "http://reason.cs.uiuc.edu/krr/.".

[11] W.N.Lu and Z.K.Shi, "Synchronous detection of the lane marking and road boundary on monocular vision," Chinese Journal of Sensors and Actuators, vol. 20, pp. 1171 – 1175, May 2007.

[12] N.Otsu, "A threshold selection method from gray-level histograms," Systems, Man and Cybernetics, IEEE Transactions on, vol. 9, no. 1, pp. 62 –66, 1979.

[13] W. Liu, H. Zhang, B. Duan, H. Yuan, and H. Zhao, "Vision-based real-time lane marking detection and tracking," in Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on, 2008, pp. 49 –54.

[14] S. Choi, T. Kim, and W. Yu, "Robust video stabilization to outlier motion using adaptive ransac," in Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, oct. 2009, pp. 1897 –1902.

[15] O. Chum and J. Matas, "Optimal randomized ransac," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 30, no. 8, pp. 1472 –1482, aug. 2008.

[16] C. L. Lawson and R. J. Hanson, Solving Least Squares Problems. Englewood Cliffs. Prentice-Hall, 1974.