https://doi.org/10.2352/EI.2023.35.16.AVM-110

tRANSAC: Dynamic Feature Accumulation across Time for Stable Online RANSAC Model Estimation in Automotive Applications

***Shimiao Li* [1], *Yang Song* [2], *Ruijiang Luo* [1], *Zhongyang Huang* [1], *Chengming Liu* [2]**
[1] ***OmniVision Technologies Singapore Pte. Ltd.***
[2] ***OmniVision Technologies Inc, Santa Clara, USA***
Email: {shimiao.li, yang.song1, ruijiang.luo, zhongyang.huang, cliu}@ovt.com

Abstract

RANdom SAmple Consensus (RANSAC) is widely used in computer vision and automotive related applications. It is an iterative method to estimate parameters of mathematical model from a set of observed data that contains outliers. In computer vision, such observed data is usually a set of features (such as feature points, line segments) extracted from images. In automotive related applications, RANSAC can be used to estimate lane vanishing point, camera view angles, ground plane etc. In such applications, changing content of road scene makes stable online model estimation difficult. In this paper, we propose a framework called tRANSAC to dynamically accumulate features across time so that online RANSAC model estimation can be stably performed. Feature accumulation across time is done in such a dynamic way that when RANSAC tends to perform robustly and stably, accumulated features are discarded fast so that fewer redundant features are used for RANSAC estimation; when RANSAC tends to perform poorly, accumulated features are discarded slowly so that more features can be used for better RANSAC estimation. Experimental results on road scene dataset for vanishing point and camera angle estimation show that the proposed tRANSAC method gives more stable and accurate estimates compared to baseline RANSAC method.

1. Introduction

RANdom SAmple Consensus (RANSAC) is widely used in computer vision and automotive related applications. It is an iterative method to estimate parameters of mathematical model from a set of observed data that contains outliers [1]. In computer vision, such observed data is usually a set of features (such as feature points, line segments) extracted from image. In automotive related applications, RANSAC can be used to estimate lane vanishing point, driving direction vanishing point, camera rotation angles, ground plane, stereo camera fundamental matrix etc. In such applications, observed data as input of RANSAC is usually extracted from road scene images which changes from time to time. As content of road scene changes, at some time point, a large number of features may be extracted from image frame that may lead to accurate model estimation; while at some other point of time, only very few number of features can be extracted from image frame which may lead to poor model estimation. Therefore, if RANSAC estimation is only based on features extracted from each single frame, estimated model may be very unstable due to road scene content change. Although adding temporal filter such as Kalman filter[2] may ease the instability of final estimation to some extent, instability of RANSAC estimation from single frame may still affect accuracy of final estimated model. A novel framework is needed for handling changing content of road scene and sufficiently making use of road scene video stream data for stable and accurate online RANSAC model estimation.

In this paper, we propose a new framework called *tRANSAC* to dynamically accumulate features across time so that online RANSAC model estimation can be stably and accurately performed with large number of features in spite of road scene content change. Feature accumulation across time is done in such a dynamic way that when RANSAC tends to perform robustly, accumulated features are discarded fast so that fewer redundant features are used for RANSAC estimation; when RANSAC tends to perform poorly, accumulated features are discarded slowly so that more features can be used for RANSAC estimation. The above mechanism is realized by adding dynamic accumulated feature pool, model estimation quality measure module, dynamic feature accumulation control module and key frame selection module into baseline RANSAC workflow. Compared to baseline RANSAC workflow, tRANSAC workflow takes advantage of dynamic nature of road scene video stream and overcome vulnerability of online RANSAC model estimation to road scene content change.

2. Related Works

Video stream data contains much richer information compared to single frame image. However, how to make use of such data in practice to improve accuracy of RANSAC model estimation has not been well studied in literature. To our knowledge, there is no previous published work of adapting RANSAC algorithm using temporal adaptively accumulated features for online model estimation from changing automotive video stream data. Although in [3], 3D points are naively accumulated in time for offline ground plane estimation, it does not handle feature accumulation in adaptive way for online model estimation as we propose in this paper. In [4], authors try to accumulate more spatially evenly distributed features for camera pose estimation in a camera network in which all cameras are static. There are also a number of previous works [5, 6, 7] which use Kalman filter or extended Kalman filter (EKF) to improve stability of estimated model parameters using temporal information, however, RANSAC model estimation process is still done using data from single frame. While results in our experiments show that even with the use of Kalman filter, the proposed tRANSAC workflow

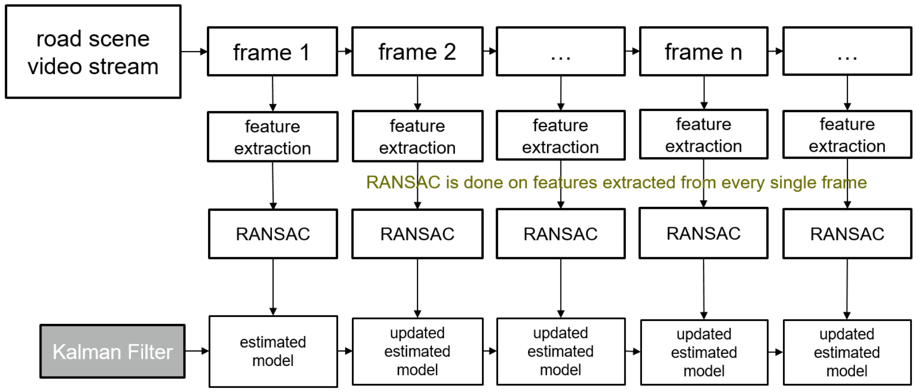

Figure 1. *Baseline workflow.*

still can further improve model estimation accuracy over single frame based RANSAC workflow.

3. Method

In this section, we firstly describe baseline RANSAC workflow for RANSAC model estimation using video stream data. Then we propose tRANSAC workflow and describe in detail its innovative components. We also give some examples to show how tRANSAC workflow improves model estimation robustness compared to baseline workflow.

3.1 RANSAC Model Estimation from Video Stream Data: Baseline Workflow

A simple straightforward workflow of using RANSAC algorithm in online model estimation is to extract features from each single frame, apply RANSAC using the extracted features to estimate model parameters. Then estimated model parameters are input as measurement to Kalman filter [2] so that it gives less noisy estimated model together with knowledge of uncertainty of current estimates (see Figure 1). This simple workflow is widely used in current literature [6, 7].

In the above baseline RANSAC workflow, each RANSAC estimation only uses features exacted from single image frame. In automotive applications, as vehicle drives fast, road scene content may change fast. In some frames, we can extract large number of inlier features without many outliers; while in other frames, we can extract very few inlier features or large number of outlier features which may lead to failed or inaccurate RANSAC estimation.

3.2 Proposed tRANSAC Workflow

To handle the above mentioned instability, we propose new workflow called *tRANSAC* to dynamically accumulate features across time so that RANSAC can be robustly performed with large number of features which are accumulated in a period of time (see Figure 2). In tRANSAC, feature accumulation across time is done in such a dynamic and adaptive way that when RANSAC tends to perform robustly, accumulated features are discarded fast so that fewer redundant features are accumulated and used for RANSAC estimation; when RANSAC tends to perform poorly, accumulated features are discarded slowly so that more features can be used for RANSAC estimation. The above mechanism is realized by adding model estimation quality measure module, dynamic feature accumulation control module, and key frame selection module into the baseline framework. These modules are described in more details in Section 3.3.

Figure 3 shows an example of using optical flow feature to estimate driving direction vanishing point. If only optical flow vectors extracted from current frame are used for RANSAC estimation, a small number of inliers leads to inaccurate vanishing point estimation (D1 in left image of Figure 3). In tRANSAC workflow, large number of optical flow features which are accumulated across time are used and give more accurate RANSAC estimation of vanishing point (D2 in right image of Figure 3).

Figure 4 shows another example of using near-horizontal or near-vertical line segments to estimate camera roll angle for camera view correction. Using baseline RANSAC workflow, we cannot get enough number of line segments to give us confident RANSAC estimation (Figure 4 top left). However, by using tRANSAC workflow, we can get good estimation of roll angle from a large number of accumulated near-vertical line segments in accumulated feature pool (Figure 4 top right). Furthermore, we can perform accurate roll angle correction on the scene image (Figure 4 bottom right).

3.3 Dynamic Feature Accumulation

Figure 5 gives clear illustration of dynamic feature accumulation process. In certain time point, after features are extracted from current frame (a in Figure 5), they are added into dynamic accumulated feature pool (c in Figure 5). Then RANSAC is performed using features accumulated in the pool (d in Figure 5). After that estimated model quality measure component (e in Figure 5) measures how good current estimated model is and send the quality measure q to dynamic feature accumulated control component (b in Figure 5)), where q may be a function of RANSAC votes, model residual error or number of inlier features etc. Then dynamic feature accumulation control component (b in Figure 5) takes input q and decides how many or to what extent old features in the dynamic accumulated pool (c in Figure 5) should be retained or discarded. This control component (b in Figure 5) is like water tap and sink. It allows new features from new frame flow into feature pool and let old features slowly run out of feature pool by computing control variable d, which is a monotonically decreasing function of q. The larger d is, the slower old features in the pool are discarded and the more features tend to be retained

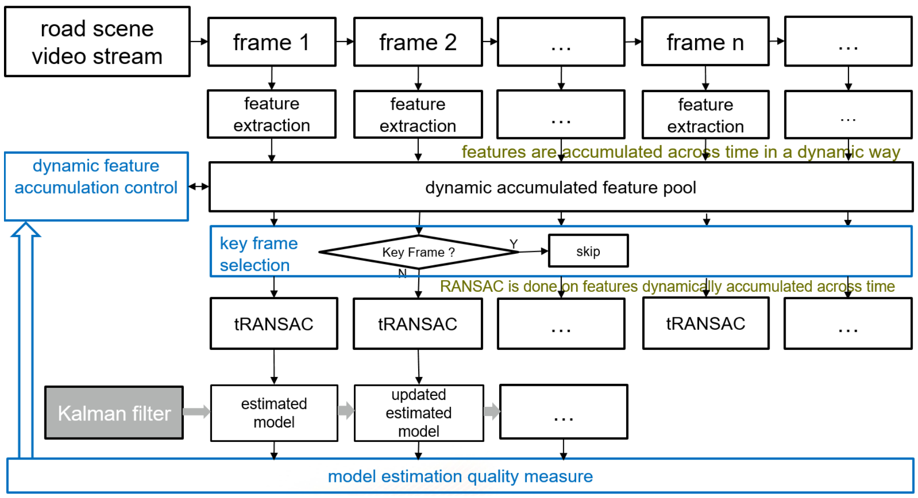

Figure 2. *Proposed tRANSAC workflow.*

in the pool.

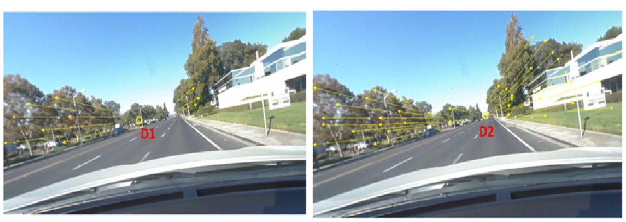

Figure 3. *tRANSAC example 1. Left: RANSAC estimation of lane vanishing point (D1) by optical flow features extracted from single frame. Right: tRANSAC estimation of lane vanishing point (D2) by optical flow features accumulated dynamically across time. (only inlier optical flow vectors are shown for better visualization.)*

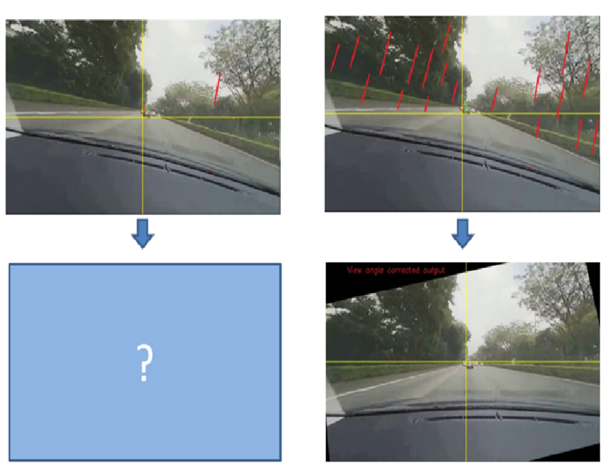

Figure 4. *tRANSAC example 2. Left: No camera roll angle can be estimated by RANSAC as too few features (red near-vertical line segments) can be extracted from single frames. Right: camera roll angle can be estimated by tRANSAC from dynamically accumulated features (red near-vertical line segments) and roll angle can be corrected accurately.*

Figure 6 gives a simple and concrete example of dynamic feature accumulation process. In this example estimated model quality measure component (e in Figure 6) measures current estimated model quality q based on number of RANSAC support votes. If q is larger than a threshold T_1, dynamic feature accumulation control (b in Figure 6) will open sink to let old features flow out of the pool. This is realized by a feature weight decay process, in which the control components (b in Figure 6) sets control vari-

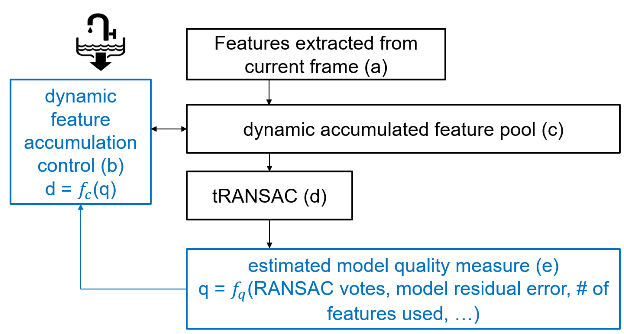

Figure 5. *Dynamic feature accumulation process.*

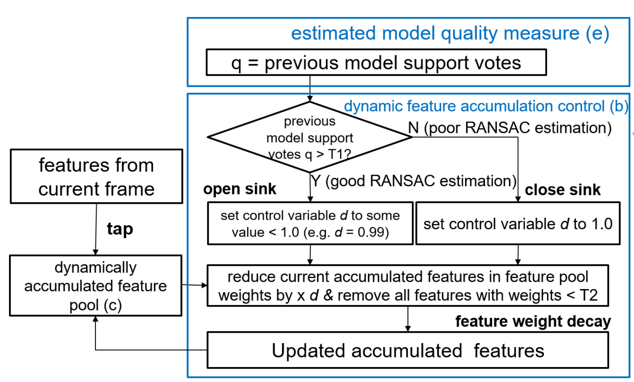

Figure 6. *Dynamic feature accumulation process: a simple and concrete example.*

able d to some value smaller than 1.0 and multiplies weights of current features in the pool (c n Figure 6) by this value. Then all features in the pool with weights smaller than another threshold T_2 are removed from the pool. On the other hand, if q is smaller than T_1, dynamic feature accumulation control will close sink to retain all features in the pool by setting control variable d to be 1.0.

3.4 Key Frame Selection

Figure 7 gives clear illustration of key frame selection process in tRANSAC workflow. At each frame time point, after dynamic feature accumulate pool is updated by process described in Section 3.3, tRANSAC workflow checks the consistency of current feature pool with previous estimated model. If consistency is larger than a threshold, it skips this frame from doing RANSAC, otherwise, it goes to do RANSAC estimation using accumulated features in the pool. It is worth noting that this consistency check process only involves one round of feature voting on the previous estimated model, while RANSAC estimation involves a large number of looped rounds for features to vote out a new estimated model. Therefore, key frame selection process may largely reduce computational cost of unnecessary RANSAC estimation on frames not bringing much new information.

4. Experiments

In this section, we firstly introduce road scene benchmark datasets which were built for testing effectiveness of the proposed method. Section 4.2 briefly describes method of estimating driving direction vanishing point and compares result between using RANSAC and tRANSAC method. Section 4.3 describes method of estimating camera roll angle and compares results between using RANSAC and tRANSAC.

4.1 Benchmark Dataset

To test effectiveness of the proposed tRANSAC workflow, we have collected a number of road scene test video clips in Santa Clara (US), Taipei (TW) and Singapore (SG). The video clips are taken while vehicle is driving forward on road and camera is having certain degree (up to 15) of out of position view angles (yaw / pitch / roll). In different locations, road scene content has different characteristics, which is summarized in Table 1. Figure 8 gives sample scenes from the three datasets.

We have manually labelled driving direction vanishing point (Figure 9) and a few horizontal lines and vertical lines for each test clip (Figure 10). For evaluating vanishing point estimation, we use *warped vanishing point (VP) error* which is distance between labelled vanishing point and estimated vanishing point after warping the estimated vanishing point into image centre (see Figure 9). For evaluating roll angle estimation, we use estimated roll angle to do roll angle correction on images. After the correction, we measure angles between all hand labelled horizontal or vertical line segments and horizontal or vertical direction in corrected image (Figure 10 right). Roll angle estimation error of a clip is defined as average value of those angles on all hand labelled line segments in the clip.

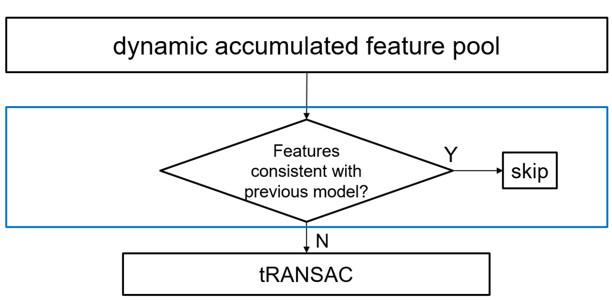

Figure 7. *Key frame selection process.*

Table 1: Benchmark Test Datasets

	scene environments
US dataset (14 clips)	highway, not many tall buildings
TW dataset (15 clips)	city street, tall buildings, crowded traffic
SG dataset (14 clips)	city street, tall building, big tropical trees

4.2 Driving Direction Vanishing Point Estimation Using Optical Flow

We use KLT algorithm [8, 9] to compute optical flow between consecutive frames. Ideally, all optical flow vector lines should go through driving direction vanishing point. We then use baseline workflow and tRANSAC workflow to estimate driving direction vanishing point using extracted optical flow as input. Table 2 gives evaluation result of vanishing point estimation using optical flow on different groups of dataset. Firstly, we compare result of single frame estimation without using Kalman filter. RANSAC or tRANSAC is done on each frame and warped VP error is computed on each frame. We report average value of warped VP error on all frames across all video clips in each dataset. Results in Table 2 column 2 and column 3 show that tRANSAC generates much smaller estimation error compared to RANSAC on single frames. Secondly, when Kalman filter is added in, both RANAC and tRANSAC error drop significantly. However, tRANSAC plus Kalman filter workflow still has significantly smaller error than that of baseline workflow (see Table 2 column 4 and column 5). Finally, adding in key frame selection slightly increases or decreases the error, depending on the dataset (see Table 2 column 6). However, it may save computational cost depending on the platform. We also convert warped vanishing point error in pixel to yaw and pitch error in degree to have some more intuitive idea of accuracy (see Table 2 column 6). Moreover, we define key frame ratio as ratio between number of frames taken as key frame and total number of frames. In Table 2 column 6, we also report key frame ratio on different dataset. It is shown on dataset with richer image features (TW dataset with busy street scenes), key frame ratio is smaller than that on the other two datasets.

4.3 Camera Roll Angle Estimation Using Near-Horizontal and Near-Vertical Line Segments in Road Scene

We use near-horizontal and near-vertical line segments extracted using edgelets computation [10] as input features for

Figure 8. *Sample scenes from US (left), TW (middle) and SG (right) datasets.*

Table 2: Evaluation result of vanishing point (VP) estimation using optical flow

Processing resolution 640x480	Average per frame VP error in pixel (RANSAC)	Average per frame VP error in pixel (tRANSAC)	Average per clip VP error in pixel (RANSAC + Kalman Filter)	Average per clip VP error in pixel (tRANSAC + Kalman Filter)	Average per clip VP error in pixel (tRANSAC with key frame selection + Kalman Filter)
US dataset	50.98	25.91	16.85	11.16	10.53 (yaw/pitch error: 1.82 / 1.87 degree) (key frame ratio: 0.51)
TW dataset	88.08	22.71	16.17	11.46	13.75 (yaw/pitch error: 2.51 / 1.89 degree) (key frame ratio: 0.41)
SG dataset	59.85	15.23	14.32	7.12	8.87 (yaw/pitch error: 1.76 / 1.05 degree) (key frame ratio: 0.58)

RANSAC or tRANSAC estimation of roll angle. Roll angle estimation error is computed as we described in Section 4.1. Table 3 shows result of comparison between using baseline workflow and using tRANSAC workflow for roll angle estimation. It is shown that tRANSAC gives numerically smaller errors on all the datasets. Adding key frame selection process increases or decreases slightly on different datasets. Key frame ratio is relatively low in TW dataset due to large number of line features in busy road scene in Taipei. Figure 11 shows sample result of feature accumulation process in tRANSAC roll angle estimation and image after roll angle correction. RANSAC baseline workflow failed to estimate confident roll angle on this video clip due to lack of features in single frames. Average processing time of vanishing point estimation plus roll angle estimation is around 30-40ms per frame on our PC (2.5GHz).

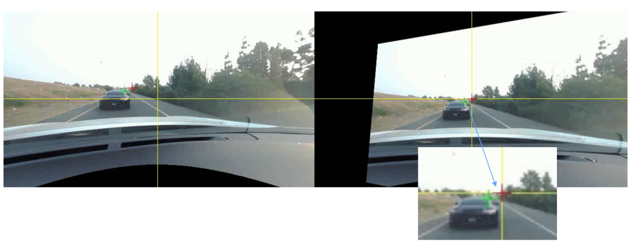

Figure 9. *Warped vanishing point (VP) error. Top left: red color cross - estimated vanishing point, green color cross - hand labelled vanishing point. Top right: image after warping estimated vanishing point to image center. Bottom right: warped vanishing point (VP) error is defined as distance between labelled vanishing point (green color cross) and estimated vanishing point (red color cross) in warped image in which estimated vanishing point lies in image center.*

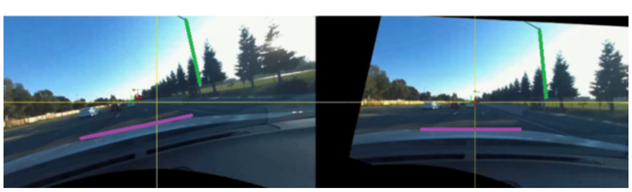

Figure 10. *Left: hand labelled horizontal (purple) and vertical (green) line segments. Right: labelled horizontal (purple) and vertical (green) line segments after roll angle correction.*

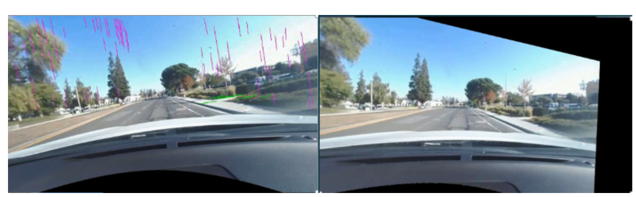

Figure 11. *Left: accumulated near horizontal (green) / vertical (purple) lines in tRANSAC feature pool for roll angle estimation. Right: roll angle correction using roll angle estimated by tRANSAC workflow.*

5. Discussion

It is noted that tRANSAC workflow is based on assumption that model to be estimated is constant within short period of time

Table 3: Evaluation result of roll angle estimation from near horizontal or vertical line segments

Processing resolution 640x480	Estimated roll angle error (in degree) (RANSAC + Kalman filter)	Estimated roll angle error (in degree) (tRANSAC + Kalman filter)	Estimated roll angle error (in degree) (tRANSAC + Kalman filter + key frame selection)
US dataset	5.12	2.13	2.62 (key frame ratio: 0.62)
TW dataset	2.77	1.35	2.04 (key frame ratio: 0.28)
SG dataset	3.05	1.82	1.77 (key frame ratio: 0.59)

(up to a few frames). Since features are in and out dynamic accumulated feature pool adaptively, tRANSAC does not require model to be estimated to be constant in long period of time. It is also worth noting that extended Kalman filter (EKF) itself could be used as powerful online model estimation tool [11] in parallel with RANSAC algorithm. This kind of workflow is usually adopted in robotics community. In future work, we may compare the accuracy and convergence speed of tRANSAC workflow with EKF model estimation workflow.

6. Conclusion

In summary, compared to baseline RANSAC workflow, the proposed tRANSAC workflow takes advantage of dynamic nature of road scene video stream by accumulating features across time in dynamic way. In such way, vulnerability of RANSAC estimation to road scene content is overcome. Experimental results have proved the effectiveness of using tRANSAC workflow in vanishing point and camera view angle estimation from automotive road scene videos in different environments. Baseline RANSAC workflow is for solving general computer vision problems, while tRANSAC workflow is designed and proved to better solve problems in automotive applications. tRANSAC is suitable for Autonomous Driving Assistant System (ADAS) related applications with better robustness in dynamic changing road environment.

References

[1] J. M. Martinez-Otzeta et al, RANSAC for Robotic Applications: A Survey, Sensors, 23, no.1, pg. 327. (2023).

[2] P. Zarchan and H. Musoff, Fundamentals of Kalman Filtering: A Practical Approach, American Insistute of Aeronautics and Astronautics, Incorporated, 2000.

[3] F. Mufti et al, Spatio-Temporal RANSAC for Robust Estimation of Ground Plane in Video Range Images for Automotive Applications, 11th International IEEE Conference on Intelligent Transportation Systems (ITSC). (2008).

[4] N. Pellicano et al, Robust Wide Baseline Pose Estimation from Video, 23rd International Conference on Pattern Recognition (ICPR). (2016).

[5] J. Jang et al, Camera Orientation Estimation Using Motion-Based Vanishing Point Detection for Advanced Driver-Assistance Systems, IEEE Transactions on Intelligent Transportation Systems, 22, no. 10, pg. 6286-6296. (2020).

[6] J-K Lee and K-J Yoo, Real-time Joint Estimation of Camera Orientation and Vanishing Points, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015).

[7] A. Borkar et al, Robust Lane Detection and Tracking with RANSAC and Kalman Filter, 16th IEEE Conference on Image Processing (ICIP). (2009).

[8] C. Tomasi and T. Kanade, Detection and Tracking of Point Features, Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

[9] J. Shi and C. Tomasi, Good Features to Track, 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (1994).

[10] K. Chaudhury et al, Auto-Rectification of User Photos, 2014 IEEE Conference on Image Processing (ICIP). (2014).

[11] J. Sola et al, Impact of Landmark Parametrization on Monocular EKF-SLAM with Points and Lines, Int. Journal of Computer Vision, 97, no. 3, pg. 339-368. (2012).

Author Biography

Shimiao Li is currently a Senior Staff Algorithm Engineer in OmniVision Technologies Singapore. She received her Ph.D. in Electrical and Computer Engineering from National University of Singapore (2010). She has authored or co-authored a number of technical papers and patents in area of computer vision and image processing. She has also conducted algorithm developments which have been used in medical imaging, mobile phone, video surveillance and ADAS chip products.

Yang Song is currently a Senior Image Scientist at OmniVision Technologies Santa Clara, CA headquarters. She obtained a master degree at The University of Chicago in 2018. She has 5 years' experience in computer vision and image signal processing. Yang conducted several comprehensive research and development in OmniVision that were utilized in different products such as cell phones, security cameras, automotive systems.

Ruijiang Luo received his BS, MS in Electrical and Electronics Engineering from the University of Science and Technology of China (1996, 1999) and MS in Computer Science from the National University of Singapore (2002), respectively. His work has focused on computer vision since then, specifically in video surveillance, and autonomous driving, both in academy and industry. He is now working in OmniVision Technologies, Singapore on computer vision for CMOS sensor.

Zhongyang Huang received his MS in Information Engineering from NTU (Singapore) in 2001 and BS in Biomedical Engineering from SJTU (China) in 1993. From 1994–1999 and 2001–2013, he worked as engineer with KangMing Biomedical Inc in China and R&D leader with Panasonic Singapore respectively. He joined OmniVision Singapore in 2013 and has then managed vision-based R&D analytic projects. He received three PASCAL-VOC winner prizes, (co-)authored 1 book chapter, 24 technical papers and 20+ granted patents.

Chengming Liu received his BS, MS degrees in Electrical Engineering from Fudan University and MS degree in Computer Science from Georgia Institute of Technology. He joined Omnivision Technologies in 2006, where he has been working on CMOS image sensors development. He is currently in charge of the algorithm development team as Vice President in Omnivision Technologies, Santa Clara, United States.