An Improved Phase Correlation Method for Stop Detection of Autonomous Driving

ZHELIN YU\textsuperscript{1}, LIDONG ZHU\textsuperscript{1}, (Member, IEEE), AND GUOYU LU\textsuperscript{2}

\textsuperscript{1}National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China
\textsuperscript{2}Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, Rochester, NY 14623, USA

Corresponding author: Zhelin Yu (zhelinyu@std.uestc.edu.cn)

This paper was supported by National Natural Science Foundation of China (No. 61871422), and Chinese Ministry Funds (No. 6140518050316).

ABSTRACT Simultaneous Localization and Mapping (SLAM) is the process by which a mobile robot carrying specific sensors builds a map of the environment and at the same time uses this map to estimate its pose. Currently, SLAM has been proven its value and is a hot topic. However, challenges still exist: when the mobile robot stops in the process of motion and a large number of feature points in the environment move slightly, the feature matching process cannot eliminate the non-stationary feature point pairs. The introduction of a large number of outliers (non-stationary feature point pairs) seriously affects the observation process of SLAM. It directly leads to the estimation errors of mobile robot pose and the 3D features position, and further leads to keyframe trajectory drift. If there is a mechanism that allows the robot to accurately detect itself in a stop state, then the pose and map points could be locked, and the state variables could be optimized to make the system enter the positive succession. Therefore, detecting the stop status of the mobile robot is a significant work for SLAM. In this manuscript, an improved phase correlation method is proposed to solve the problem of stop detection for the autonomous driving vehicle in the dynamic street environment. After experiments, it is revealed that the stop detection is significant for the performance improvement to the state-of-the-art visual SLAM system, and the improved phase correlation method has higher stop detection accuracy than the conventional phase correlation in various scenarios.

INDEX TERMS Stop detection, phase correlation, simultaneous localization and mapping, vanishing point.

I. INTRODUCTION
Simultaneous Localization and Mapping (SLAM) technology, as the main egomotion estimation algorithm for autonomous driving, has been concerned by major research institutions and industries around the world [1]–[3]. ORB-SLAM [4], [5] is one of the most reliable and easy-to-use systems on the modern visual SLAM system. It innovatively uses three threads that run in parallel to realize real-time SLAM: tracking, local mapping, and loop closing. Tacking thread extracts FAST corners and computers the ORB descriptors [6] for the current frame from the camera. It matches with the last keyframe for 2D-2D feature points and estimates the camera’s current pose roughly with polar geometric constraints and the position of map points by the triangulation method. Based on the tracking thread, local mapping thread uses the bundle adjustment to optimize the keyframe currently processed, all the keyframes connected to it in the covisibility graph, and all the map points seen by those keyframes. The loop closure thread takes the last keyframe processed by the local map, tries to detect and close loops, and correct the scale drift [5].

However, when the mobile robot is stopping in a dynamic environment, the real world contains a large number of dynamic objects, such as moving pedestrians and vehicles. In this situation, tracking thread may match a large number of correspondences on dynamic objects, and these outliers are introduced in the process of 2D-2D tracking and 3D-2D matching. It further leads that the drift for odometry and error position estimation for map points are not prone to be corrected in the follow-up local mapping thread and loop closure thread. Therefore, most state-of-the-art approaches that initially designed for conducting SLAM in static environments are not capable of handling severe dynamic scenarios [7].

There is a clear need to devise outliers removal techniques that are able to accurately remove the mismatches for robust
feature matching. Standard visual SLAM achieves this by computing fundamental matrix or homography using Random Sample Consensus (RANSAC), which is a robust outlier exclusion method [8]. However, this method is generally time-consuming and difficult to cope with the situation where there are too many mismatches. There are effective algorithms for mismatch removal in the front-end preprocessing, such as the LPM [9], [44], LMR [10], RFM-SCAN [11], VFC [12], [42], [43], GLPM [40], [41], and GMS [13]. The principle of these algorithms is to maintain the local neighborhood structures of those potential true matches [9]. These algorithms can be very fast and accurate to remove the mismatches caused by similar features, which can accomplish the mismatch removal from thousands of putative correspondences in only a few milliseconds [9]. But these approaches may fail when there are many dynamic objects in front of the robot or the captured image is occluded by a large moving object, because the moving objects can also retain local neighborhood matches. Some superior algorithms use semantic segmentation based on deep learning to recognize pedestrians or vehicles that may be moving in the image to address this issue [7], [14], [15]. But the semantic segmentation result cannot determine whether the objects are truly moving. For example, feature points on pedestrians or vehicles that are truly stationary can still be used for state estimation. Furthermore, the result of semantic segmentation would lose the precise contour of the objects, which will cause problems in the extraction of feature points. In addition, this kind of method is time-consuming, just the semantic segmentation module would take 37.57 ms [7], which makes it generally unable to complete real-time operations.

In order to rule out the impact of these dynamic correspondences on the state estimation when the mobile robot is in the stopping state, we address this problem from a new perspective. If the mobile robot can accurately detect it in the stop state in the front-end processing, then the pose and map points can be estimated in the back-end processing to correct the state variables in time, and ultimately the drift of SLAM will be reduced and its performance will be greatly improved. Therefore, we present a robust stop detection algorithm for mobile robots in the dynamic environment. The application scenario considered in this paper is to detect the stop for the autonomous vehicle in the dynamic street environment, which is one of the most difficult directions in the visual SLAM application filed [14], [17].

Directly, the stop detection problem can be simply addressed with reading the CAN-BUS or wheel speedometer data. However, these data are not readily available to general research institutions, because they are not public and the data formats are different in various car manufacturers. And the resolution of the data is only 1 km/h, which cannot be used to detect ultra-low-speed movement. Usually, autonomous vehicles are equipped with GPS, IMU, and camera. Angular velocity and acceleration measured by IMU cannot be used to detect the stop state due to the bias drift. Meanwhile, the accuracy of GPS device is too low to meet this requirement. For these reasons, this paper considers only using the camera to realize the stop detection from the perspective of image processing.

With the aim of accurate stop detection, we advocate an improved phase correlation algorithm to extract the pixel movement information of two adjacency frames in the video stream. The traditional phase correlation algorithm [18] is an efficient technique for simple rigid transformation, but it is not suitable for the camera mounted on a vehicle where there are non-rigid transformations containing translation, rotation, scale changing, etc. These transformations give rise to inaccurate phase correlation results. In this paper, we “undo” these transformations by segmenting the image based on the detected vanishing point (VP), computing the local projective transformation, and applying it to the local sub-image. The result will be new synthesized local sub-images in which the objects are shown with their correct geometric shape. Ultimately, phase correlation is applied on each synthesized local sub-image to determine whether the vehicle stops. The flowchart of our method is shown in Fig. 1. The quantitative and qualitative experimental analysis in different scenarios shows that the proposed method can give an effective stop detection.

More concretely, the contributions of this paper can be summarized as follows:

1) This paper presents a new research problem in SLAM: stop detection. When a mobile robot stops in a scene where a large number of feature points move slowly, usually visual SLAM cannot detect the stop state correctly, which may lead to estimation errors for the pose of the robot and the position of the 3D feature points. Based on this, we bring forward the stop detection problem in visual SLAM.

2) We provide an improved phase correlation method to detect the stop state for the autonomous driving vehicle. Firstly, a new VP detection method by the combination of edge lines and optical flow lines is sketched in this paper. And then the VP is used for three-dimensional segmentation. Secondly, we rectify the local sub-images for eliminating the transformation caused by the central projection of a camera. Finally, the phase correlation is applied to the rectified sub-images instead of the whole original images to get the pixel level motion vector, so as to detect the stop state.
3) We apply the VP detection method proposed in this paper on the publicly available dataset, and apply the stop detection method proposed in this paper in various scenarios, including night, rain, dynamic streets, and low-speed to stop. Experiments show that the proposed VP detection method can achieve good performance in terms of both effectiveness and efficiency, and the proposed stop detection algorithm shows good performance in various challenging scenarios.

4) We use the proposed stop detection result to correct the trajectories constructed by state-of-the-art visual SLAM system. Experiments show the corrected trajectories are more accurate and reasonable than the original trajectories. They demonstrate that the stop detection is significant for the performance improvement of the current SLAM system.

The remainder of this paper is organized as follows. Section II describes the related works of VP detection and Fourier phase correlation, followed by preprocessing presented in section III, in which we propose a robust and accurate VP detection method. In section IV, we propose an improved phase correlation method to detect the stop for the autonomous driving vehicle. Section V explains the experiments conducted to demonstrate the effectiveness of the proposed algorithm. Finally, this paper is concluded with some discussions and future works in section VI.

II. RELATED WORK

A. VANISHING POINT

A set of parallel lines in the world project to converging lines on the imaged plane under projective transformation. The intersection point, maybe at infinity, is known as the VP. Because the VP analysis contains the direction information of the straight lines and provides crucial cues for inferring the 3-dimensional geometric structure of a scene, locating the VP correctly has been an active research problem in the field of computer vision [25].

Over the past decades, there has been a large number of works devoted to addressing the problem of VP detection. The majority of VP detection methods can be roughly divided into two categories: edge-based method [26]–[28], and texture-based method [29]–[31]. The edge-based method is to cluster line segments, and accurately estimate a VP from the line cluster. Because of its computation efficiency, this method can be applied to real-time applications. However, the edge-based method requires that the images have enough clear edges. In the scenes which do not contain any strong edges, such as rural and desert areas, it is sensitive to spurious lines and does not perform well. To counter these problems, the texture-based method was proposed. The local texture orientation in each pixel is estimated, and then it is used to vote for the position of the VP. The texture-based method improves the accuracy significantly compared to the edge-based method in various scenes, but its computation time is very high and this method cannot be realized in real time. Furthermore, both of these methods are frame-by-frame detecting VP, which ignores the motion relationship between consecutive frames. These frame-by-frame VP detection methods are susceptible to noise when the image is blurry, such as over- and under-exposed images.

To solve the problems above, we develop a novel algorithm to detect the VP by utilizing the combination of edge lines and optical flow lines. The optical flow lines are essentially the moving trajectory of the vehicle. Assuming the moving direction of the vehicle is parallel to lane markings or road boundaries, consequently, these trajectory lines converge to the VP. Therefore, the VP detection method based on the combination of edge lines and optical flow lines can increase the robustness and accuracy but avoids the corresponding shortfalls of the frame-by-frame VP detection methods.

B. PHASE CORRELATION

Erturk [32] first proposed using the shift property of the Fourier Transform for translational motion estimation, and then applied it to global motion estimation, image stitching and image stabilization. This scheme focuses on the overall correlation of the images. Even if there exist local noise and jitter, it will not affect the overall motion estimation. It is very suitable for the scenario in this paper. Kumar et al. [19] proposed a fast and robust two-dimensional affine global motion estimation algorithm based on phase correlation in the Fourier-Merlin domain and robust least-squares model fitting of the sparse motion vector field. Simultaneously, Xie et al. [20] applied the algorithm to motion estimation of UAV images, and Li et al. [21] applied the algorithm to the local registration of images.

We can draw lessons from the above schemes to get the phase correlation of adjacent frames, and then judge whether the vehicle stops or not. However, these schemes have two limitations. The first limitation is that the scheme can only be applied to simple rigid transformation estimation, but there exists a more complex non-rigid transformation between adjacent frames taken by the camera mounted on the vehicle. And the second limitation is that the scheme judges the transformation of two images based on the results of one phase correlation, which leads to the detection result be greatly affected by the noise in the environment, yielding unreliable information about stop detection. Therefore, the existing phase correlation algorithms cannot be directly used for stop detection. For resolving these problems, we propose using VP for three-dimensional image segmentation. Phase correlation is performed in each one-dimensional sub-image, and the minimum norm value of local motion vectors is selected as the overall moving amount of two adjacent frames. Furthermore, in order to eliminate the influence of noise, we calculate the mean and standard deviation of 10 consecutive frames’ moving amount as the detection result of the current frame. Finally, we use the detection result to determine whether the vehicle stops. The experimental results show that the proposed phase correlation algorithm based on
three-dimensional segmentation has better stop detection performance than the conventional phase correlation algorithm.

III. PREPROCESSING

In order to remove the projective distortion from each perspective image in the video stream, the image is preprocessed in this section. Firstly, a new VP detection algorithm is introduced. Secondly, the VP is used for three-dimensional image segmentation. The image is divided into three local sub-images: ground, the left view between sky and ground, and the right view between sky and ground. Finally, different projective transformations are applied to rectify each local sub-image, and the result will be three new synthesized images in which the objects in the plane are shown with their correct geometric shapes.

A. VANISHING POINT DETECTION

It is assumed that the moving direction of the vehicle is parallel to the lane markings and road boundaries, and therefore the optical flow lines converge to the VP. Inspired by this, we propose a novel VP estimation method by combining the edge line detection with optical flow detection. This fast and efficient method consists of four steps: extraction of the region of interest (ROI), extraction and selection of line segments, extraction and selection of optical flow lines, and voting for the VP. The technical details of each step are described below.

1) EXTRACTION OF THE REGION OF INTEREST

As the positions of VP in two consecutive frames are usually not far apart, the position of the VP in the previous frame could provide a powerful clue for the VP estimation in the current frame. Hence, we firstly draw a horizontal line on the current frame, which passes through the position of the VP detected in the previous frame. And then we extract the part below the horizontal line as ROI, which is shown in Fig. 2 (a). If the current processing image is the first frame of the video, ROI takes the whole image.

2) EXTRACTION AND SELECTION OF LINE SEGMENTS

In this subsection, the Canny edge detector is utilized to enhance the edges in ROI firstly, and then perform the Hough transform to extract line segments. Suppose that there are total \( N \) line segments are extracted, and denoted as \( \Omega = \{ l_1, l_2, \ldots, l_N \} \). For convenience, let \( k_i \) represents the slope of \( l_i \), where \( i = 1, 2, \ldots, N \). The extracted line segments are shown in Fig. 2 (b).

Because we rely on the extracted line segments to identify the VP in an image, it should minimize the number of irrelevant or cluttered line segments in \( \Omega \). However, in the real environment, a large number of spurious line segments, which are left by plants, buildings, pedestrians, etc, appear on the images. These spurious line segments not only slow down the speed of VP detection, but also reduce the detection accuracy. Thus, we refine the line segments in \( \Omega \) based on the analysis of their slope, length, and distribution to delete the spurious line segments.

a: LINE SEGMENTS SLOP

We only take count of those line segments whose angle is between 20 degrees and 80 degrees, or between 100 degrees and 160 degrees. We remove approximately horizontal and vertical line segments, because they do not parallel with the driving direction of the vehicle, but converge to the horizontal VP and vertical VP, respectively. We use \( \Omega' = \{ |l_i| \in \Omega, \tan 20^\circ \leq |k_i| \leq \tan 80^\circ \} \) to represent the filtered line segments, and use \( N' \) to denote the line segments’ number in the filtered set \( \Omega' \).

b: LINE SEGMENTS LENGTH

Shorter lines are unconcerned to VP, but related to distortion and misleading. However, for various practical situations, setting a fixed-length threshold to delete short line segments is not a universal method. Hence, we sort the line segments in \( \Omega' \) in descending order by their length. Then a new set \( \Omega'' \) is formed by collecting the top 90% in \( \Omega' \), where \( \Omega'' = \{ l_1, l_2, \ldots, l_{N''} \} \), \( N'' = \left\lfloor 0.9 \cdot N' \right\rfloor \), and \( \left\lfloor \cdot \right\rfloor \) represents the rounding operator.
**c: LINE SEGMENTS DISTRIBUTION**

Based on the principle that the VP’s distance of two adjacent frames is close, the line segments detected in the current frame should be abnormal if they are too far from the VP of the previous frame. For each line segment in \( \Omega'' \), they are arranged in descending order based on the distance between it and the previous frame’s VP. Only the top 90% are taken to form a new set \( \Omega''' = \{l_1, l_2, \ldots, l_{N''}\} \), where \( N'' = \lfloor 0.9 \cdot N \rfloor \).

Finally, Fig. 2 (c) shows the filtered line segments, which will be used to effectively estimate the VP in the following section. By removing the redundant line segments from a given image, the computation time is exponentially reduced but the estimation accuracy is greatly improved, because they are very dependent on the quantity and the quality of line segments for voting.

**3) EXTRACTION AND SELECTION OF OPTICAL FLOW LINES**

Optical flow measurements can be obtained by computing pixels’ motion vectors in continuous image sequences at any time, and then give the instantaneous moving direction of the vehicle. Assuming that the moving direction of a vehicle would be parallel to the road boundaries or lane markings, then the optical flow lines would converge to the VP. Based on this, this sub-section will introduce the extraction and selection of optical flow lines, and then use them to vote for the VP in the next step.

Many different optical flow algorithms have been developed in the last decades. Among them, Lucas-Kanade (LK) et al. [33] and Horn-Schunck (HS) [34] are the most widely used ones. LK optical flow algorithm is chosen in this paper for its relatively simpler calculation. Suppose that optical flow vectors are extracted, which are denoted as \( \Psi = \{\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_M\} \), and shown in Fig. 2 (d). In order to increase the detection accuracy and the algorithm speed, the optical flow vectors which are not related to the VP are removed, and the filtered strategies are described below.

**a: OPTICAL FLOW VECTOR LENGTH**

We limit the magnitude of the extracted optical flow vectors, because the optical flow vectors which are too long or short are usually outliers. We arrange the optical flow vectors in \( \Psi \) in ascending order based on their magnitude, remove the first 10% and the last 10% optical flow vectors, and the remaining middle 80% optical flow vectors in \( \Psi \) form a new set \( \Psi' = \{\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_{M'}\} \), where \( M' = \lfloor 0.8 \cdot M \rfloor \).

**b: OPTICAL FLOW VECTOR DISTRIBUTION**

Just like the strategy of line segment filtering, we remove the optical flow vectors far from the VP of the previous frame. We arrange the optical flow vectors in \( \Psi' \) in ascending order based on the distance between optical flow vectors and the VP of the previous frame, and keep the first 90% of the optical flow vectors in \( \Psi' \) to constitute a new set \( \Psi'' = \{\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_{M''}\} \), where \( M'' = \lfloor 0.9 \cdot M' \rfloor \). In subsequent processing, regardless of the direction of the optical flow vectors, we treat them as line segments to vote for the VP, which are denoted as \( \Psi''' = \{\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_{M''}\} \), and shown in Fig. 2 (e).

**4) VOTING FOR THE VANISHING POINT**

We merge the line segments of set \( \Omega'' \) and set \( \Psi''' \) into a new set \( \Upsilon = \{l_1, l_2, \ldots, l_{N''}, \vec{o}_1, \vec{o}_2, \ldots, \vec{o}_{M''}\} \). In order to pursue the robustness, we use RANSAC approach [8] on set \( \Upsilon \) to get the best estimation of the VP. Fig. 2 (f) is a sample result of VP detection.

**B. IMAGE SEGMENTATION**

Note that since the ground and the front in an image are not in the same plane, the projective transformation that must be applied to rectify the front is not the same as the one used for the ground. Therefore, three-dimensional image segmentation is performed in this sub-section for applying different projective transformations to rectify each dimensional local sub-image in subsequent processing.

VP provides a strong cue for identification of the three-dimensional structure of images, because all road boundaries and lanes intersect at a VP. We use the VP as a segment point to segment image in three components. However, when the vehicle’s driving direction is not parallel to the road boundaries or lane markings in a few case, the VP detection algorithm proposed in this paper may not perform well. In order to avoid VP detection error introduced into image segmentation, we limit the segmentation point to a range near the center point of the image in actual use. This means that if the detected VP point is more than 50 pixels away from the center point of the image, we will use the center point instead of the detected VP for image segmentation. The steps of image segmentation are as follows.

1) Connect the segment point and the four vertices of the image, and divide the image into four parts, namely, the top of the image, the bottom of the image, the left side of the image, and the right side of the image. Actually, these four parts correspond approximately to the sky, the ground, left field of view between sky and ground, and right field view between sky and ground, respectively, as shown in Fig. 3 (a). The texture information contained in the sky is not abundant enough, which leads to the noise of phase correlation results. Therefore, we only consider the three parts in the subsequent processing: the ground, the left part, and the right part.

2) For the convenience of projection transformation, we transform these three triangles into three trapezoids respectively. This transformation is usually learned from experience. In this paper, we pick up half of the triangle for the left, half of the triangle for the right, and 80% of the triangle for the ground. The extracted parts are shown in the green trapezoidal parts of Fig. 3 (b).
FIGURE 3. (a) The segment point divides the image into four parts: the sky, the ground, left field of view between sky and ground, and right field view between sky and ground. (b) Transform three triangles into three trapezoids respectively.

C. SUB-IMAGE RECTIFICATION

For eliminating the transformation caused by the central projection of a camera, in this section, we rectify the sub-images according to the projective transformation to make the vertical direction perpendicular to the \( x \)-direction of the image, and lane lines parallel with the \( y \)-direction of the image. The \( x \)- and \( y \)-directions of the image are shown in Fig 3.

Perspective transformation is an invertible mapping to project the image onto a new perspective plane. Assuming \( X \) is a coordinate in the original image, and \( X' \) is a coordinate in rectified image. \( X \) and \( X' \) are related by a projective matrix as \( H \). The core of perspective transformation is to find the perspective matrix \( H \). In this paper, the perspective matrices are obtained with the perspective transformation function of OpenCV [35]. After rectification, the projective distortion of three segmented sub-images are removed, and newly synthesized sub-images are projected to the horizontal and vertical planes respectively, as shown in Fig. 4.

IV. PHASE CORRELATION

After preprocessing, we will apply the phase correlation to the three-segmented and rectified synthesized local sub-images.

The principle of phase correlation is as follows. Assuming we have a reference signal \( f_1(x, y) \) and its shifted version \( f_2(x, y) \):

\[
f_2(x, y) = f_1(x + \Delta x, y + \Delta y),
\]

the offset on the spatial domain can be expressed on the frequency domain as follows:

\[
F_2(u, v) = e^{-j2\pi(ux + vy)} * F_1(u, v),
\]

where \( F_2(u, v) \) and \( F_1(u, v) \) are the Fourier transformations of \( f_2(x, y) \) and \( f_1(x, y) \), respectively. Then, the normalized cross-power spectrum is given by

\[
H(u, v) = \frac{F_1(u, v) \cdot F_2^*(u, v)}{|F_1(u, v)| \cdot |F_2(u, v)|} = e^{-j2\pi(ux + vy)}.
\]

The Dirac delta function will be calculated after inverse Fourier transform on the cross-power spectrum,

\[
IFT(H(u, v)) = \delta(x - \Delta x, y - \Delta y).
\]

Finally, the motion vector \((\Delta x, \Delta y)\) of two consecutive frames is identified as a significant peak in this Dirac function. Fig. 5 is an example.

In this paper, the phase correlation method is applied to corresponding rectified synthesized local sub-images between two consecutive frames. For all three synthesized local
Z. Yu et al.: Improved Phase Correlation Method for Stop Detection of Autonomous Driving

FIGURE 6. (a) The left side of the camera was blocked by the big bus. (b) The right side of the camera was blocked by the truck.

sub-images of an image frame, local motion vectors are estimated from the respective sub-images of the previous frame based on phase correlation. For comparing these local motion vectors, we calculate the norm value of the motion vector \((\Delta x, \Delta y)\), as follows:

\[
Norm = \sqrt{\Delta x^2 + \Delta y^2}.
\] (5)

For each image frame, in total three local motion vectors corresponding to the synthesized local sub-images are determined and the norm value of each vector is calculated. The minimum of the three norm values is assigned to the global inter-frame moving amount. This strategy is designed to reduce the influence of local noise, and the specific analysis is as follows. Considering the realistic scenario shown in Fig. 6, when the autonomous vehicle was in stop state, a large vehicle passed through from the left or right side. This causes a large area of synthesized local sub-image was blocked by the big car. And the norm value corresponding to that part was not close to or even far greater than 0 due to the big car’s movement. This further leads to misjudgment for stop detection. Therefore, we choose the smallest one as the global inter-frame moving amount in order to eliminate the local noise caused by the above situation. The diagram illustrating the full flow of the proposed stop detection method is shown in Fig. 7.

Ultimately, we calculate the mean and standard deviation of moving amount of consecutive 10 frames as the detection results for the last frame in 10 frames, and set the thresholds for the mean and standard deviation based on experiential value, so as to make a stop judgment. The principle of setting the thresholds of mean and standard deviation is elaborated in the experimental part.

V. RESULTS AND ANALYSIS
A. EXPERIMENT ABOUT VANISHING POINT DETECTION

For the purpose of evaluating the performance of the proposed VP detection method in this paper, we performed different detection methods on the Malaga dataset [36], which has a total of 800 images in urban scenarios with size 1024 \(\times\) 768. With the help of 25 students in the laboratory, we manually marked the ground truth of VP for each image. In order to make the marked results as accurate as possible, we first manually reviewed and repaired abnormal values. Secondly,
for the results of each frame, we discarded the maximum and minimum values in the 25 manually marked data. Assuming that these remaining 23 data follow Gaussian distribution, we took their mean value as the final ground truth. The normalized Euclidean distance [22] between the detected result and the ground truth is measured as the error for comparison, which is defined as:

$$\text{NormDist} = \sqrt{(x - x_0)^2 + (y - y_0)^2}, \quad (6)$$

where $(x, y)$ and $(x_0, y_0)$ are 2D column vectors denoting the spatial positions of the detected result and the ground truth respectively, $w$ and $h$ represent the image’s width and the height, respectively. Obviously, the smaller the $\text{NormDist}$, the higher the accuracy of the detected result [39].

For analyzing the performance of the algorithm proposed in this paper, we compare it with the edge-based VP detection algorithm [24] and the texture-based VP detection algorithm [23]. Fig. 8 shows the statistics of $\text{NormDist}$ for these three algorithms. The results of different algorithms are calculated in an 10-bin histogram as shown in Fig. 8, where the horizontal axis denotes $\text{NormDist}$, and the vertical axis gives the number of images in each histogram bin [37]–[39]. We just take count of the $\text{NormDist}$ in the interval $[0, 0.01]$, and place the error into the $\text{NormDist} = 0.01$ when the $\text{NormDist} \geq 0.01$. In fact, the higher the percentage in the left part, the better the performance of the algorithm. From Fig. 8, the algorithm proposed in this paper has a better detection performance and higher accuracy.

Furthermore, based on these histograms, we also compared the performance of the three detection methods by accuracy curve [38] which is shown in Fig. 9. The horizontal axis denotes different norm distance which is between manually marked ground truth position and the detected position, and the vertical axis denotes the accuracy percentage corresponding to different norm distances. The higher the curve at different norm distance, the better the detection result of this method. For example, we can see the detail for the norm distance of 0.008, which true distance is 11.87 pixels according to (6). The accuracy percentage of texture method and edge method are 67.13% and 54.75%, but the accuracy percentage of our method is 83.63%, much higher than texture-based and edge-based VP detection algorithms. From Fig. 9, the method presented in this paper has higher detection accuracy.

Besides, for further comparison, Table 1 gives the average norm distance and pixel distance comparison of the three algorithms. The lower the average distance, the higher the accuracy of the algorithm. Based on the observation of Table 1, our method outperforms the other two methods.

Finally, Table 2 shows the time efficiency comparison of the three VP detection algorithms, which shows that although our method is not the fastest one in the three methods, it can be performed in real time because the shooting time for one frame is 33 milliseconds. And texture-based vanishing point detection is too time-consuming to achieve real-time processing.

In sum, experimental results show that the overall performance of the method in this paper can be in real time and performs significantly better than the other two method, the edge-based VP detection [24] and the texture-based VP detection [23], which indicates that the proposed method can achieve good performance in terms of both the effectiveness and the efficiency.

### B. COMPARISON WITH TRADITIONAL PHASE CORRELATION

To intuitively evaluating the performance of the proposed improved phase correlation algorithm, we synthesize a pair of images that simulate two consecutive frames captured by the camera mounted on a vehicle, as shown in Fig. 10 (a) and Fig. 10 (b), in which Fig. 10 (b) is a synthetic product by advancing Fig. 10 (a) 40 pixels. Experimental results are shown in Fig. 10 (c) and Fig. 10 (d).

The conventional phase correlation method [18] is to directly calculate the phase correlation of the original full image. From Fig. 10 (c), its result does not show a sharp frontal peak, and finally motion vector comes out $(-2.154976, -0.294877)$, which means registration failure. However, by using the algorithm proposed in this paper, local phase correlation results for detecting special

---

**FIGURE 8.** Comparison of VP detection methods in norm distance distribution.

**FIGURE 9.** Comparison of VP detection methods in accuracy curve.

**TABLE 1.** Average error comparison.

| Content             | Edge-based | Texture-based | Our method |
|---------------------|------------|---------------|------------|
| Average norm distance | 0.0342     | 0.0115        | 0.0051     |
| Average pixel distance | 50.87      | 17.20         | 7.71       |

**TABLE 2.** Running time comparison.

| Time (ms) | Edge-based | Texture-based | Our method |
|-----------|------------|---------------|------------|
| Time (ms) | 11.73      | 3082.00       | 22.06      |
FIGURE 10. The performance of the algorithm proposed in this paper is compared to the conventional phase correlation for motion estimating. (a) First frame. (b) Second frame, which is a synthetic product by advancing the first frame 40 pixels. (c) The result of the conventional phase correlation for detecting special translation, and motion vector comes out \((-2.154976, -0.294877)\), which means registration failure. (d) The result of the improved phase correlation for detecting special translation, in which the motion vector of synthesized left sub-image is \((40.050320, 0.012773)\), the motion vector of synthesized bottom sub-image is \((0.000754, 39.998138)\), and the motion vector of synthesized right sub-image is \((-40.006794, 0.006321)\).

C. COMPARISON OF THE STOP DETECTION PERFORMANCE IN DIFFERENT SCENARIOS

In order to further verify the effectiveness of the proposed method in this paper, several video sequences have been recorded in different roads and scenarios with a forward-looking camera mounted on the vehicle windshield, as shown in Fig. 11.

1) THRESHOLD SELECTION

For the final comparison, we set the threshold for the mean of detection result as 0.25, and the threshold for the standard deviation as 0.1. That implies when the mean and standard deviation of current frame detection results are lower than their respective thresholds at the same time, we infer that this frame has entered the stop state, otherwise it is in the running state. The selection of threshold is very important, which is directly related to the performance of the algorithm. In this paper, the principle of setting the thresholds is mainly from verification tests in multiple scenarios. We screened multiple scenes videos with stopping frames for verification tests, including urban roads, rural roads, traffic jams on highways, day, night, dusk, rainy day, snowy day, wipers moving, windshield washer fluid spraying, fast stop, slow stop, turn stop, straight stop, uphill stop, downhill stop, and so forth. We compare the stop detection performance with different thresholds with the ground truth manually labeled under various situations. Firstly, we manually marked the start and end frame numbers of the actual stopping interval in these screened videos. Then, we calculated the mean and standard deviation of the detection result for each frame. Finally, in all manually marked stopping frames of verification tests, we recorded the maximum values of the mean and standard deviation of detection results, represented by \(A\) and \(B\). In all marked non-stopping frames, we recorded the minimum values of the mean and standard deviation of detection results, represented by \(C\) and \(D\). The mean and standard deviation thresholds are taken as the intermediate values of \(A\) and \(C\), \(B\) and \(D\), respectively. It is remarkable that the videos for the threshold selection are completely independent of the videos for subsequent simulation experiments.

2) EXPERIMENTAL RESULT

The proposed method was successfully applied in a series of experiments with different scenarios. Three challenging scenarios are specially presented in this subsection to evaluate the performance of stop detection, which are shown in Fig. 12, Fig. 13, and Fig. 14. Each of these figures contains four sub-figures: (a), (b), (c), and (d). Sub-figure (a) shows the vehicle stopping situation in different scenes. Sub-figure (b) gives the stop detection results by using the conventional phase correlation algorithm, where the horizontal axis denotes the frame index of the video sequence, and the vertical axis denotes the norm value of the motion vector for the conventional full image phase correlation method. Sub-figure (c) and sub-figure (d) represent stop detection results by using the algorithm proposed in this paper, in which the horizontal...
The first scenario is the normal case where the traffic on the intersection, where there were vehicles moving around near the field of view. The ground truth of the start and end frame indexes for the stop in this scenario are the 214th frame and the 1193th frame. Fig. 13 (b) is the stop detection result of the traditional phase correlation algorithm. Two issues are found from this sub-figure. Firstly, the maximum value of the result during stopping is even greater than the minimum value during moving. Therefore, it is difficult to set a threshold to determine whether the vehicle is stopping based on the traditional phase correlation algorithm. Secondly, in the two periods during the vehicle stop state, from 400th frame to 600th frame and around the 700th frame, the detection results occur fluctuations. The reason for this phenomenon is that there was a big car passing by at those moments. Fig. 13 (c) indicates that the traditional phase detection method cannot achieve accurate results, which directly leads to false detection.

The second scenario is in the dynamic street scene, as shown in Fig. 13 (a). The vehicle stopped at a small intersection, where there were vehicles moving around near the field of view. The ground truth of the start and end frame indexes for the stop in this scenario are the 214th frame and the 1193th frame. Fig. 13 (b) is the stop detection result of the traditional phase correlation algorithm. Two issues are found from this sub-figure. Firstly, the maximum value of the result during stopping is even greater than the minimum value during moving. Therefore, it is difficult to set a threshold to determine whether the vehicle is stopping based on the traditional phase correlation algorithm. Secondly, in the two periods during the vehicle stop state, from 400th frame to 600th frame and around the 700th frame, the detection results occur fluctuations. The reason for this phenomenon is that there was a big car passing by at those moments. Fig. 13 (b) indicates that the traditional phase detection method cannot achieve accurate results, which directly leads to false detection.
The specific analysis of the detection results is as follows. Firstly, before the vehicle stopped and after it restarted, which is 0th to 214th frames and 1193th to 1300th frames, the detection result has no obvious noise. Secondly, during the vehicle stop state, even there were many cars passing in front of the vehicle, the mean and standard deviation of detection results are steadily below their respective thresholds. Compared to the conventional phase correlation algorithm, our algorithm is able to exclude the influence of surrounding moving objects because after image segmentation, we take the minimum of three synthesized local sub-images’ norm value as the global inter-frame moving amount of the current frame to avoid the noise caused by the local motion. At the same time, it also benefits from the projective transformation of the local sub-image, which increases the local translation of the image. Based on the above analysis, the algorithm proposed in this paper can obtain super stop detection performance in dynamic scenarios.

The third scenario shows a particular situation where the vehicle stopped on a rainy night, as shown in Fig. 14 (a). It can be seen from the sub-figure that the overall image is relatively dark due to the night. And there were a lot of raindrops on the windshield of the vehicle due to the heavy rain and the wiper of the vehicle swung rapidly, which blurred the image. Fig. 14 (b) is the result of applying the traditional phase correlation algorithm. We can observe that the detection results fluctuate greatly both in the stopped state and the motion state, which results in false stop judgment. This result occurs due to the weak image quality at rainy night and the blur caused by wipers. Fig. 14 (c) and Fig. 14 (d) are the stop detection results using the improved phase correlation algorithm. We detect that the start and end frame indexes of the stop are the 243th frame and 666th frame, which are exactly the same as the ground truth. This experiment shows that our scheme has outstanding stop detection performance even the video quality is severely degraded.
From the experimental results in this subsection, it can be concluded that the method presented in this paper has excellent stop detection performance in various scenarios.

D. STOP DETECTION FOR SLAM PERFORMANCE ANALYSIS

We use two sequences in KITTI Datasets [45], [46], 2011_09_26_drive_0029_sync in raw data and sequences-07 in odometry data, which contain the stopping frames in dynamic scenarios, to show the performance improvement of the proposed stop detection method for the dynamic visual SLAM, and these scenarios are shown in Fig. 15. To simplify the notation, we use odometry-07 to denote sequences-07 in odometry data, and these scenarios are shown in Fig. 15. To simplify the notation, we use odometry-07 to denote sequences-07 in odometry data, and raw-29 to denote 2011_09_26_drive_0029_sync in raw data. The odometry-07 contains stereo images and ground truth states. The raw-29 contains stereo images, synchronized IMU and DGPS measurements, and ground truth states. We run the kitti_odom_test mode and kitti_gps_test mode of VINS-Fusion [47] on odometry-07 and raw-29 respectively, and the trajectories are shown as Fig. 16. VINS-Fusion is an extension of VINS-Mono [48], [49], which is an optimization-based multi-sensor state estimator, and achieves accurate self-localization for autonomous driving, and unmanned aerial vehicle. VINS-Fusion supports multiple visual-inertial sensor types (mono camera and IMU, stereo cameras and IMU, and even stereo cameras only) [47]. In this experiment, kitti_odom_test mode and kitti_gps_test mode of VINS-Fusion are used, the former is a visual odometry mode from stereo cameras only, and the latter is a tightly-coupled fusion mode from stereo cameras, and DGPS.

We use the stop detection algorithm proposed in this paper to process the trajectories as follows.

1) The stop detection method is used to detect the stop frames in the sequence, and the start frame and the end frame in the stopping interval are recorded as the $k^{th}$ frame and the $(k+m)^{th}$ frame, respectively;

2) The trajectory coordinate corresponding to $i^{th}$ frame in the sequence is defined as $P_i$, and the trajectory offset before stopping and after restarting is calculated, which is denoted as $\Delta P = P_{k+m} - P_k$;

3) For the trajectory from $P_k$ to $P_{k+m}$, because we detect that the vehicle is at a stopping state, we construct a new trajectory as follow: $\{P_j\}' = P_j$, $j = k, k + 1, \ldots, k + m$;

4) For the trajectory after $P_{k+m}$, in order to eliminate the accumulated error during the stopping period of the original trajectory, we construct the new trajectory: $\{P_j\}' = P_j - \Delta P$, $j = k + m + 1, k + m + 2, \ldots$;

5) Finally, we construct the corrected trajectory as $\{P_i\}' = \{P_i\}_i = 1, 2, \ldots, k - 1, j = k, k + 1, \ldots$.

For the odometry-07 and raw-29, the original trajectory, the corrected trajectory, and ground truth are shown in Fig. 16. Since these are the global display of the overall trajectories, we see that the original and corrected trajectories almost overlap during stopping and after restarting in Fig. 16. In the subsequent experimental analysis, we will give the partially enlarged view of that area to show the detailed difference. Besides, in the lower right part of Fig. 16 (b), the original trajectory has a large jitter compared to the ground truth at the beginning. This situation occurs because kitti_gps_test was running for raw-29, that is, the system needs to fuse the data of the stereo cameras and DGPS measurements. Thus there exist jitter in the convergence process of initialization when the system starts. In the subsequent experimental analysis, we do not take this initializing trajectory into consideration.

1) STOP DETECTION PERFORMANCE ANALYSIS

The real stop frame indexes of odometry-07 and raw-29 are marked manually. For odometry-07, its stopping interval is from the 665th frame to 719th frame. For raw-29, its stopping interval is from the 206th frame to 261th frame. The stop detection algorithm proposed in this paper detects the stopping interval of the two sequences are from the 665th frame to 719th frame, and the 206th frame to 261th frame, respectively, which is completely consistent with the result of manual marking.

2) THE COMPARISON BETWEEN THE ORIGINAL TRAJECTORY AND THE GROUND TRUTH

Fig. 17 shows the partially enlarged view of the ground truth and VINS-Fusion original trajectory in the stopping

FIGURE 15. Dynamic scenarios in KITTI Datasets. The vehicle is stopping at a dynamic intersection where there are vehicles passing by. (a) The odometry-07. (b) The raw-29.

FIGURE 16. Results of the VINS-Fusion. (a) Trajectory in odometry-07. (b) Trajectory in raw-29.
and its neighbor area. We can see that the ground truth has obvious and abnormal jittering effect during the stop and its neighbor areas. We have a brief analysis of the reasons for this phenomenon. The ground truth comes from the inertial navigation fusion of the DGPS and IMU measurements. However, when the vehicle is at the low-speed or stopping state, the vibration of the vehicle engine and the bias of IMU cause the signal-to-noise ratio of the IMU measurement to be very low. This eventually leads to the trajectory of inertial navigation fusion occur the jittering effect. Therefore, the ground truth coming from IMU during the vehicle stop state has no meaningful reference.

In addition, due to the cumulative error of the VINS-Fusion, the original trajectories corresponding to the odometry-07 and raw-29 deviate from the ground truth, which is far bigger than the deviation caused by not adding stop detection. Therefore, in order to effectively and clearly show the performance improvement of SLAM brought by the stop detection method, we compare the original trajectory and the corrected trajectory in the subsequent analysis, so as to explain the improvement by our method on VINS-Fusion.

3) THE COMPARISON BETWEEN THE ORIGINAL TRAJECTORY AND THE CORRECTED TRAJECTORY
We mainly pay attention to the difference between the original trajectory and the corrected trajectory after the vehicle stops. Fig. 18 shows the two trajectories comparison of the $X-Y$, $Y-Z$, and $X-Z$ planes during the stop and its neighbor areas. From Fig. 18, the original trajectory is composed of the blue line, green line, and brown line. During the vehicle stop state, the green lines in three planes occur obvious jittering effect. Each green line randomly moves in a square area with a side length of 3 cm. This phenomenon is contrary to the objective fact that the vehicle is in a stopping state. And the result shows that the state-of-the-art SLAM system can not deal with the vehicle stopping state well in the dynamic scene. Another perspective, the corrected trajectory is composed of the blue line and purple line. The corrected trajectory is locked at a point during the stop and the overall corrected trajectory is very smooth during the stop and its neighbor areas, which is consistent with the actual trajectory of the vehicle. These experimental results show that the stop detection algorithm proposed in this paper is significant for the performance improvement of the current visual SLAM system.

E. TIME CONSUMING ANALYSIS
For more accuracy processing, we only calculate the VP location every 1 second, and in this second, all frames are segmented with the same VP. We used 100,000 frames to do the verification experiments, which is under the 2.8 GHz Intel Core i7 platform and contains these three scenarios mentioned above. The average time of single-frame processing using the conventional full image phase correlation method is 3.98 millisecond, and processing time for the proposed method is 7.14 millisecond. The average time to calculate the VP location for a single frame is 22.06 millisecond.

In addition, if the computing power of the platform is extremely limited, we can take the center point of the image instead of the detected VP position to segment the images, which will also maintain good accuracy, and satisfy real time requirements.

VI. CONCLUSION
In this paper, we propose a new research problem of stop detection for visual SLAM, which is critical for vehicle state and egomotion estimations, and then develop an improved
phase correlation method for stop detection of autonomous driving. Specifically, a new VP detection algorithm is proposed, which combines edge detector and optical flow detector. And based on the VP detection result, we develop a projection transformation method on the sub-images per frame, and then apply the Fourier transform to determine whether the vehicle is stopping or not. Experiment results have shown that the method has extraordinary stop detection performance in different situations, like overtaking traffic, and rainy night. For future research, we will further improve the accuracy and robustness of the algorithm, and add the stop detection algorithm to the state estimation framework of visual SLAM.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: Part I,” IEEE Robot. Autom. Mag., vol. 13, no. 2, pp. 98–108, Jun. 2006.
[2] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (SLAM): Part II,” IEEE Robot. Autom. Mag., vol. 13, no. 3, pp. 108–117, Sep. 2006.
[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” IEEE Trans. Robot., vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: A versatile and accurate monocular SLAM system,” IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[5] R. Mur-Artal and J. D. Tardos, “ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras,” IEEE Trans. Robot., vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in Proc. Int. Conf. Comput. Vis., Nov. 2011, pp. 2564–2571.
[7] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, “DS-SLAM: A semantic visual SLAM towards dynamic environments,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 1168–1174.
[8] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” Commun. ACM, vol. 24, no. 6, pp. 381–395, 1981.
[9] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, “Locality preserving matching,” Int. J. Comput. Vis., vol. 127, no. 5, pp. 512–531, 2019.
[10] J. Ma, X. Jiang, J. Jiang, J. Zhao, and X. Guo, “LMR: Learning a two-class classifier for mismatch removal,” IEEE Trans. Image Process., vol. 28, no. 8, pp. 4045–4059, Aug. 2019.
[11] X. Jiang, J. Ma, J. Jiang, and X. Guo, “Robust feature matching using spatial clustering with heavy outliers,” IEEE Trans. Image Process., vol. 29, pp. 736–746, 2020.
[12] J. Ma, J. Zhao, J. Tian, A. L. Yuille, and Z. Tu, “Robust point matching via vector field consensus,” IEEE Trans. Image Process., vol. 23, no. 4, pp. 1706–1721, Apr. 2014.
[13] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, “GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 4181–4190.
[14] G. Lu and X.-I. Wong, “Taking me to the correct place: Vision-based localization for autonomous vehicles,” in Proc. IEEE Int. Conf. Intell. Image Process. (ICIP), Sep. 2019, pp. 2966–2970.
[15] X. Lu, H. Wang, S. Tang, H. Huang, and C. Li, “DM-SLAM: Monocular SLAM in dynamic environments,” Preprints.org, Jan. 2020, doi: 10.20944/preprints202001.0123.v1.
[16] D. Titterton, J. L. Weston, and J. Weston, Strapdown Inertial Navigation Technology, vol. 17. Edison, NJ, USA: IET, 2004.
[17] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, “Simultaneous localization and mapping: A survey of current trends in autonomous driving,” IEEE Trans. Intell. Vehicles, vol. 2, no. 3, pp. 194–220, Sep. 2017.
[18] M. Drackmuller, “Phase correlation method for the alignment of total solar eclipse images,” Astrophysical J., vol. 706, no. 2, pp. 1605–1608, Dec. 2009.
[44] J. Ma, J. Zhao, H. Guo, J. Jiang, H. Zhou, and Y. Gao, “Locality preserving matching,” in Proc. 26th Int. Joint Conf. Artif. Intell., 2017, pp. 4492–4498.

[45] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 3354–3361.

[46] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” Int. J. Robot. Res., vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[47] T. Qin, J. Pan, S. Cao, and S. Shen, “A general optimization-based framework for local odometry estimation with multiple sensors,” 2019, arXiv:1901.03638. [Online]. Available: http://arxiv.org/abs/1901.03638

[48] T. Qin, P. Li, and S. Shen, “VINS-mono: A robust and versatile monocular visual-inertial state estimator,” IEEE Trans. Robot., vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[49] T. Qin and S. Shen, “Online temporal calibration for monocular visual-inertial systems,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 3662–3669.

ZHELIN YU received the B.Sc. degree in information and computing science from the University of Electronic Science and Technology of China, Chengdu, China, in 2015, where she is currently pursuing the Ph.D. degree in communication and information system. Her current research interests include simultaneous localization and mapping (SLAM), monocular SLAM, and visual-inertial systems.

LIDONG ZHU (Member, IEEE) received the B.S. and M.S. degrees from Sichuan University, Chengdu, China, in 1990 and 1999, respectively, and the Ph.D. degree in communication and information systems from the University of Electronic Science and Technology of China, Chengdu, in 2003. From 2009 to 2010, he was a Visiting Scholar with the Department of Electronic Engineering, The University of Hong Kong. He is currently a Professor with the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China. His research interests include satellite communications, blind source separation, and communication countermeasures.

GUOYU LU received the B.S. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, the M.S. degree from the University of Trento, the M.S. degree from RWTH Aachen University, and the Ph.D. degree from the University of Delaware. He was a Research Scientist in autonomous driving with Ford Research and a Computer Vision Engineer with the ESPN Advanced Technology Group. He was a Visiting Scholar with the KEDRI Group, Auckland University of Technology. He was an Intern with Siemens Corporate Research and Bosch Research. He is currently an Assistant Professor with the Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology. His research interests include computer vision, machine/deep learning, multimedia, biomedical imaging, and robotics.