





# TG: Accurate and Efficient RGB-D Feature With Texture and Geometric Information

Xiangrui Zhao , Graduate Student Member, IEEE, Yu Liu , Student Member, IEEE, Zhengbo Wang, Student Member, IEEE, Kanzhi Wu, Gamini Dissanayake , Senior Member, IEEE, and Yong Liu , Member, IEEE

**Abstract**—Feature extraction and matching are the basis of many computer vision problems, such as image retrieval, object recognition, and visual odometry. In this article, we present a novel RGB-D feature with texture and geometric information (TG). It consists of a keypoint detector and a feature descriptor, which is accurate, efficient, and robust to scene variance. In the keypoint detection, we build a simplified Gaussian image pyramid to extract the texture feature. Meanwhile, the gradient of the point cloud is superimposed as the geometric feature. In the feature description, the texture information and spatial information are encoded in relative order to build a discriminative descriptor. We also construct a novel RGB-D benchmark dataset for RGB-D detector and descriptor evaluation under single variation. Comprehensive experiments are carried out to prove the superior performance of the proposed feature compared with state-of-the-art algorithms. The experimental results also demonstrate that our TG can achieve better performance especially on accuracy and the computational efficiency, making it more suitable for the real-time applications, e.g., visual odometry.

**Index Terms**—Feature detection, feature extraction, visual odometry.

## I. INTRODUCTION

THE local feature technology is widely used in many fields, such as object detection, recognition, and geometrical measurement. Although some fields currently have almost been dominated by the deep learning-based methods, there are still

Manuscript received January 4, 2022; revised March 18, 2022; accepted May 2, 2022. Recommended by Technical Editor G. M. Clayton and Senior Editor X. Chen. This work was supported by the National Natural Science Foundation of China under Grant U21A20484. (Xiangrui Zhao and Yu Liu contributed equally to this work.) (Corresponding author: Yong Liu.)

Xiangrui Zhao, Yu Liu, Zhengbo Wang, and Yong Liu are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China (e-mail: xiangruizhao@zju.edu.cn; liuychn@126.com; zhengbowang@zju.edu.cn; yongliu@ipc.zju.edu.cn).

Kanzhi Wu is with CVTE Central Research Institute, Guangzhou 510663, China (e-mail: kanzhi.wu@gmail.com).

Gamini Dissanayake is with the Centre for Autonomous Systems, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: gamini.dissanayake@uts.edu.au).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TMECH.2022.3175812>.

Digital Object Identifier 10.1109/TMECH.2022.3175812

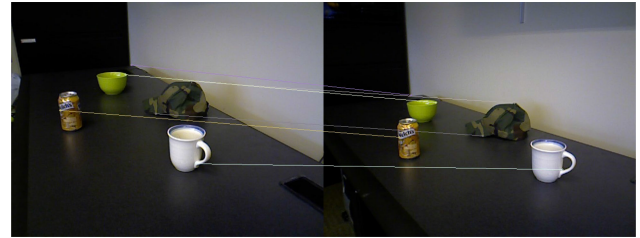
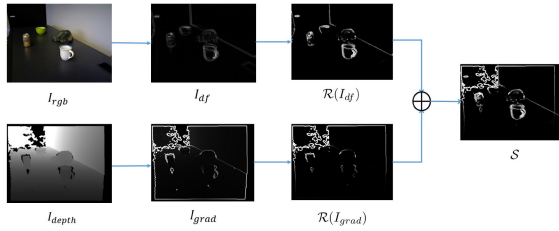


Fig. 1. Example of image matching with the proposed feature TG.

many aspects highly relies on the accuracy and efficiency of the local features, especially in the simultaneous localization and mapping (SLAM) [1], [2], structure from motion [3], and augmented reality (AR) [4], [5].

The local feature problem consists of two main aspects, i.e., keypoint detection and description, which can also be regarded as two operations: extracting keypoints and building feature vectors. Enormous progress has been made in developing robust local features in 2-D image space. Typical examples are scale invariant feature transform (SIFT) [6], speed-up robust feature (SURF) [7], and oriented FAST and rotated BRIEF (ORB) [8], which can achieve excellent performance when texture information is rich.

Due to the complementary nature of RGB image and depth, it has become a trend to fuse the texture information of RGB images with the geometric information of depth images to extract robust features. Binary robust appearance and normal descriptor (BRAND) [9] and our previous work LOIND [10] are typical RGB-D descriptors. But the inefficient use of texture and geometric information (TG) makes them perform unsatisfactorily when the scene changes drastically. Then, we propose RISAS [11] in our previous article, which consists of an RGB-D keypoint detector and a feature descriptor. The detector of RISAS calculates each normal vector's dot product with the primary normal vector from the depth image and combines it with the gray image through the autocorrelation function, respectively, to extract keypoints. The descriptor of RISAS is improved on LOIND, which encodes texture information, spatial distribution information, and plane norm information to compute feature vectors. However, RISAS directly applies the autocorrelation function to the gray image for feature detection, making it sensitive to texture changes and noise.



**Fig. 2.** Pipeline of TG feature detection. The difference of Gaussian image  $I_{dif}$  is generated through a simplified Gaussian image pyramid as the texture feature. On the other hand, the gradient image  $I_{grad}$  is generated by computing the gradient from the depth image. We apply an autocorrelation function  $\mathcal{R}$  on  $I_{dif}$  and  $I_{grad}$ , respectively, to calculate the final score map  $S$ .

This article proposes an RGB-D feature with TG,<sup>1</sup> consisting of a keypoint detector and a discriminative feature descriptor. TG is improved on RISAS [11] and inherits its robustness against scene variation. Moreover, TG improves the localization accuracy at low thresholds and significantly optimizes the efficiency to be used in real-time applications. Fig. 1 shows an example of feature matching. The contributions of this paper are as follows:

- 1) TG detector builds a simplified Gaussian image pyramid and calculates the image pyramid's self-response as the texture feature.
- 2) To speed up the extraction of geometric features, TG calculates the gradient of point cloud by axes, then superimposes the gradient values as the geometric feature.
- 3) To perceive the spatial characteristics of the local patch and integrate the global information into each feature vector, the TG descriptor couples the local normal vector to neighbor point cloud and normalizes feature vectors.
- 4) We propose an RGB-D scene benchmark dataset that can be used to evaluate features under a single variation.

## II. RELATED WORKS

### A. 2-D Texture Features

There has been extensive research on feature detectors and descriptors for 2-D RGB images. The most well-known one is SIFT [6]. It contains a difference of Gaussian interest region detector and a descriptor based on the gradient orientation histogram, which is robust to scale and rotation variation. To speed up SIFT, Bay *et al.* [7] proposed SURF, which utilizes integral images, and a Hessian matrix-based measure is used for the detector and a distribution-based descriptor. BRIEF [12] uses a binary string as the feature descriptor, which takes relatively less memory consumption and can be matched fast by Hamming distance with limited computational resources.

2-D features based on deep learning are also proposed. MatchNet [13] consists of a CNN that extracts features from patches and a network with three fully connected layers, which is used to compute the similarity between features. HardNet [14] uses

<sup>1</sup>Both the code and proposed benchmark dataset are available on [Online]. Available: <https://github.com/APRIL-ZJU/TEG>.

a triplet margin loss for metric learning that maximizes the distance between the closest positive and closest negative samples in a batch. DeTone *et al.* [15] and Li *et al.* [16] trained a primary detector on a basic figure to detect corner points, then finetune the detector with real images. To solve the nondifferentiable problem of keypoint extraction during end-to-end training, LF-Net [17] constrains the nondifferentiable part to one branch and trains the network in the other branch. Improved on LF-Net, RF-Net [18] obtains better performance with multiscale feature maps.

### B. 3-D Geometric Features

With the development of low-cost depth sensors, geometric information of the environment can be easily captured, and various 3-D features have been proposed. Zhong [19] proposed intrinsic shape signature (ISS) based on the eigenvalue decomposition of the scatter matrix consisting of the points that belong to the support set of candidates. Another example of a fixed-scale 3-D keypoint detector is keypoint quality (KPQ), which was proposed by Mian *et al.* [20]. Similar to ISS, KPQ is also based on the scatter matrix. A significant difference compared with ISS is that KPQ prunes nondistinctive points using the ratio between the maximum lengths along the first two principal axes.

### C. RGB-D Fusion Features

In some scenarios where texture information is not rich enough, those 2-D features may be invalid. Then, the depth information can be a useful supplementation to improve the discriminability of features. Tombari *et al.* [21] proposed color signature of histogram and orientation (CSHOT) that incorporates RGB information into SHOT descriptor. Nascimento *et al.* [9] proposed BRAND that encodes local information as a binary string to achieve low memory consumption. LOIND [10] encodes both the texture and depth information by orders of both intensities and angles between normal vectors. RISAS detector [10] applies the autocorrelation function to the gray image and the dot product between each normal vector and the main normal vector, respectively, then fuses these two features to extract keypoints.

More recently, some RGB-D features based on deep learning have been proposed. For example, Zeng *et al.* [22] proposed 3-DMatch, an RGB-D descriptor that combines multiple consecutive depth images into one domain named as truncated distance field (TDF), and learns the local geometric information of TDF with a convolutional network. Kehl *et al.* [23] proposed a 3-D object detection algorithm that uses regressed descriptors of locally sampled RGB-D patches for 6-D vote casting. Gupta *et al.* [24] proposed to convert depth images into HHA images [25], which encodes geometric information by three channels (horizontal disparity, height above ground, and angle with gravity). Cheng *et al.* [26] described RGB and HHA images with a convolutional network, then assign the weights of texture feature and geometric feature through a weighted gate.

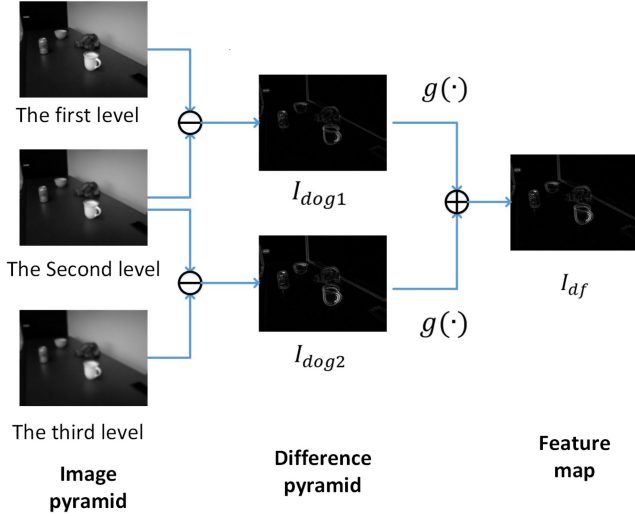


Fig. 3. Process of extracting texture features.

### III. METHOD

#### A. Keypoint Detector

In this part, we will introduce how the proposed detector extracts keypoints with both TG. The flowchart is shown in Fig. 2.

1) *Texture Encoding*: In order to obtain scale invariance, SIFT builds a Gaussian image pyramid by upsampling and downsampling the original image. It is an efficient technique but damages the keypoint accuracy at the same time. We keep the resolution of the image unchanged and convolute it with different Gaussian kernels to build a simplified Gaussian image pyramid. The Gaussian kernel function and its kernel reference  $k$ , standard deviation  $\sigma$  are shown in (1), where  $w$  is the size of window. According to the settings of SIFT, we set  $s = 3$  and  $\alpha = 1.6$ .  $s$  represents the levels of image pyramid.  $\alpha$  is the standard deviation of the first level Gaussian function. Fig. 3 shows the process of extracting texture features. We subtract adjacent images to get two DoG maps

$$\begin{aligned} G(x, y, \sigma) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \\ k &= 2^{1/s} \\ \sigma &= \alpha k^i, i \in 1, 2, 4 \\ w &= 2[4.0\sigma + 0.5] + 1. \end{aligned} \quad (1)$$

We subtract adjacent images to get two DoG maps  $I_{dog1}$  and  $I_{dog2}$ . When detecting keypoints, the changes of edges and corners will gain more attention. As a result, the texture feature map  $I_{df}$  is computed by (2), where  $g(\cdot)$  is the absolute value function.

$$I_{df} = g(I_{dog1}) + g(I_{dog2}). \quad (2)$$

When different Gaussian kernels convolve an image, the two DoG maps have edge features in different receptive fields, which can help TG perceive the neighboring in distinct ranges.

2) *Geometry Encoding*: First, we transform each point  $(u, v)$  in the depth image  $I_{depth}$  to the 3-D space to build the corresponding point cloud, as shown in (3), where  $(u_0, v_0)$  is the principal point, and  $f_x$  and  $f_y$  are the focal lengths.

$$\begin{cases} z = I_{depth}(u, v) \\ x = \frac{u-u_0}{f_x} z \\ y = \frac{v-v_0}{f_y} z. \end{cases} \quad (3)$$

To acquire the geometric feature of points in 3-D space, a typical operation is to calculate the local plane normal vector for the point cloud. However, calculating normal vectors is time-consuming. On the other hand, the noises of the point cloud will strongly affect the result. Therefore, we propose to utilize the gradient of the point cloud as the geometric feature. When calculating the gradient of point cloud, we can get the following four values:

- 1) the  $X$ -axis gradient in height direction  $I_{dxh}$ ,
- 2) the  $X$ -axis gradient in width direction  $I_{dxw}$ ,
- 3) the  $Y$ -axis gradient in height direction  $I_{dyh}$ , and
- 4) the  $Y$ -axis gradient in width direction  $I_{dyw}$ .

The geometric feature map  $I_{grad}$  is computed by

$$I_{grad} = g(I_{dxh}) + g(I_{dxw}) + g(I_{dyh}) + g(I_{dyw}). \quad (4)$$

Gradient calculation of point cloud only has subtraction, so the process is much faster compared with normal vectors calculation in RISAS.

3) *Feature Fusion*: A similar principle as Harris detector is adopted to compute the response value  $\mathcal{S}(u, v)$ , as shown in (5), where  $(u, v)$  is the keypoint coordinate in image space. The window function  $\omega(x, y)$  is a Gaussian function centered at  $(u, v)$  with a window size of 20.

$$\mathcal{R}(u, v)_I = \sum_{x, y} \omega(x, y) [I(x+u, y+v) - I(x, y)]^2. \quad (5)$$

To better receipt changes in the neighborhood, both texture information and geometric information are utilized in the proposed feature. The score function is defined as a weighted sum, as shown in the following, where  $\tau$  is used to weight these two kinds of information:

$$\mathcal{S} = \tau \mathcal{R}(I_{df}) + \mathcal{R}(I_{grad}) \quad (6)$$

4) *Keypoint Selection*: When selecting keypoints, we first take the local maximum as the candidates with a window of 11 in the score map  $\mathcal{S}(u, v)$ . For all candidates, we perform the following steps:

- 1) pick out points whose scores are larger than  $0.002 \times \mathcal{S}_{max}$ ;
- 2) remove points whose depths are missing.

Therefore, keypoints will be extracted from regions with sufficient TG. For an RGB-D image with a resolution of  $640 \times 480$ , the width of the boundary is set to 30 pixels, and the proposed detector can detect about 400 – 1200 keypoints per image.

#### B. Feature Descriptor

In this section, we will propose the feature extraction of our descriptor from texture and depth information, as well as the process of feature vector construction.



1) *Background Elimination*: To enhance the robustness of our feature, we propose a background elimination algorithm to select the patch where the descriptor is built.

For a point  $k_i$  in image, the corresponding point in point cloud is denoted as  $k'_i$ . The patch centered at  $k_i$  in image is extracted as  $\mathbf{P}^{uv}(k_i)$  and the corresponding patch in point cloud is denoted as  $\mathbf{P}^{xyz}(k'_i)$ . For each point  $k'_j \in \mathbf{P}^{xyz}(k'_i)$ , we remove the outliers with a distance greater than 0.3 m.

2) *Plane Fitting*: The normal vector is a discriminative feature of the point cloud, which is often used in the 3-D features. In the process of feature description, the proposed descriptor encodes the normal vector of the local point cloud as spatial features. We use the normal vector estimation algorithm based on the least squares proposed by Berkmann *et al.* [27].

3) *Scale Estimation*: The scale of the keypoint in the gray image is generally estimated by finding the extreme value in scale space such as SIFT [6] and SURF [7]. However, it can be easily measured using the depth information captured from the modern RGB-D sensors, such as Kinect and Xtion. In this article, we follow the approach in BRAND [9], given in the following, to scale the distance range into the scale range:

$$\begin{cases} s = \max\left(0.2, \frac{3.8-0.4\max(2,d)}{3}\right) \\ r = Rs. \end{cases} \quad (7)$$

We estimate scale  $s$  for a point with its  $d$  and select neighboring feature block with radius  $r$  to build a descriptor.  $R$  is an empirical value and is experimentally evaluated to 20 according to LOIND [10].

#### 4) Feature Extraction:

- 1) *Extract texture feature*: We extract the neighboring gray block  $P_{\text{gray}}$  from the gray image  $I_{\text{gray}}$ .
- 2) *Extract geometric feature*: We calculate the gradient of  $P_{\text{pc}}$  and get the following four items:
  - a) the  $X$ -axis gradient in height direction  $P_{\text{dxh}}$ ,
  - b) the  $X$ -axis gradient in width direction  $P_{\text{dxw}}$ ,
  - c) the  $Y$ -axis gradient in height direction  $P_{\text{dyh}}$ , and
  - d) the  $Y$ -axis gradient in width direction  $P_{\text{dyw}}$ . Then, the geometric feature block is computed by  $P_{\text{grad}} = g(P_{\text{dxh}}) + g(P_{\text{dxw}}) + g(P_{\text{dyh}}) + g(P_{\text{dyw}})$ .
- 3) *Extract Spatial Feature*: We fit a plane for the keypoint. The normal vector is transformed to the coordinate of the keypoint and notated as  $n_w$ . For each point in  $P_{\text{pc}}$ , we multiply its coordinate by  $n_w$  to obtain the space dot product block  $P_{\text{dp}}$ .

5) *Descriptor Construction*: We extract the circular feature blocks through a mask  $M$ . The three square blocks  $P_{\text{gray}}$ ,  $P_{\text{grad}}$ , and  $P_{\text{dp}}$  are multiplied, respectively, by  $M$  to the circular blocks  $P'_{\text{gray}}$ ,  $P'_{\text{grad}}$ , and  $P'_{\text{dp}}$ . Compared with the square block, the circular block avoids considering the direction of feature blocks during encoding and enhances the robustness against rotation variation.

Compared with SIFT, which uses absolute values of features to encode vectors, we use the relative order of features to improve the robustness against noises.

- 1) *Encoding texture information*: We sort values in the gray block  $P'_{\text{gray}}$  in ascending order. The sequence is divided

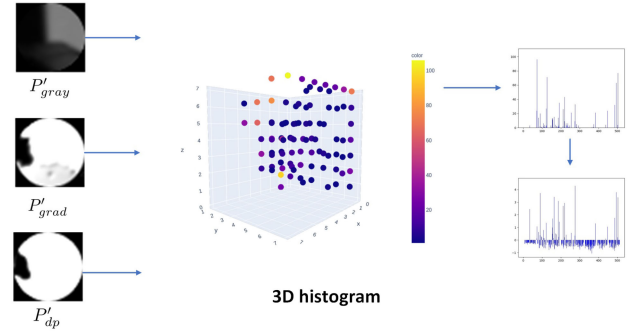


Fig. 4. Process of computing a feature vector. We extract three feature blocks for a keypoint: gray information  $P'_{\text{gray}}$ , geometric information  $P'_{\text{grad}}$ , and spatial information  $P'_{\text{dp}}$ . Their values are divided into  $ng$ ,  $np$ , and  $nd$  bins, respectively, to build a 3-D histogram, which is flattened to a feature vector.

into  $ng$  bins.  $X_i$  represents the number of elements in the  $i$ th bin. We get  $ng$  values from which an  $X$ -axis is established.

- 2) *Encoding geometric information*: We all values in the grad block  $P'_{\text{grad}}$  in ascending order. The sequence is divided into  $np$  bins.  $Y_j$  represents the number of elements in the  $j$ th bin. We get  $np$  values from which a  $Y$ -axis is established.
- 3) *Encoding spatial information*: We all values in the dot product block  $P'_{\text{dp}}$  in ascending order. The sequence is divided into  $nd$  bins.  $Z_k$  represents the number of elements in the  $k$ th bin. We get  $nd$  values from which a  $Z$ -axis is established.

For each keypoint, we use the above three axes ( $X$ ,  $Y$ ,  $Z$ ) to establish a 3-D coordinate system, thus obtaining a feature vector with dimensions of  $ng \times np \times nd$ . Increasing the number of bins can enhance the discrimination of the descriptor, but it will reduce the computational efficiency.

Fig. 4 shows the process of calculating the feature vector for a keypoint ( $ng = np = nd = 8$ ). The more vivid the color of the point, the larger the value there. As can be seen from Fig. 4, most values are 0 so the vector is very sparse. Therefore, after calculating the feature vectors for all keypoints in an image, they are normalized uniformly by dimensionalities to obtain the final feature vector of each keypoint. This strategy integrates the global information of an image into each feature vector, and it turns out to improve the performance of the proposed descriptor.

## IV. RGB-D EVALUATION BENCHMARK DATASET

Most of the existing public available RGB-D datasets are frames taken from videos. As a result, the change between images is a mixture of illumination, rotation, scale, and viewpoint variation. To evaluate the robustness of algorithms against a single variation of the scene, we setup an RGB-D evaluation benchmark dataset that contains the single variation in three scenes with the different richness of texture information and geometric information.

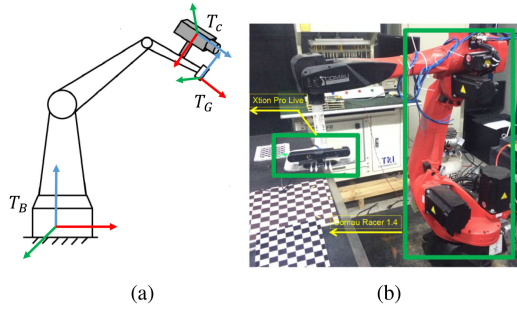


Fig. 5. (a) Schematic (b) and physical picture of the Data collection system.

We mount the Xtion LIVE PRO RGB-D camera on a Comau Racer robotic arm, which can perform six DoF movement so that the pose of each frame can be obtained directly from the controller. Under this circumstance, the environment used to evaluate the RGB-D feature can show a single variation at a time.

1) *Hand-Eye Calibration*: The schematic and physical pictures of the data collection system is shown in Fig. 5, where  $F_C, F_G$ , and  $F_B$  denote the camera coordinate system, gripper coordinate system, and the robot body coordinate system, respectively. During the movement of the robotic arm, the transformation between the robot base and the gripper  $\mathbf{T}_B^G$  can be obtained directly, and the transformation between two consecutive image frame  $\mathbf{T}_{C_i}^{C_{i+1}}$  can be computed by

$$\begin{aligned} \mathbf{T}_{C_i}^{C_{i+1}} &= \mathbf{T}_{C_i}^{G_i} \mathbf{T}_{G_i}^B \mathbf{T}_B^{G_{i+1}} \mathbf{T}_{G_{i+1}}^{C_{i+1}} \\ &= (\mathbf{T}_G^C)^{-1} \mathbf{T}_{G_i}^B \mathbf{T}_B^{G_{i+1}} \mathbf{T}_G^C. \end{aligned} \quad (8)$$

In order to compute the transformation  $\mathbf{T}_{C_i}^{C_{i+1}}$  between frame  $i$  and frame  $i + 1$ , we need to calibrate the relative transformation between camera coordinate system  $\mathbf{T}_C$  and gripper coordinate system  $\mathbf{T}_G$ .

Estimating the transformation  $\mathbf{T}_G^C$  is known as the hand-eye calibration problem, which has been investigated by Horaud, Dornaika [28], and Daniilidis [29]. In this work, we adopt more recent method proposed by Liang and Mao [30].

2) *Dataset With Single Variation*: Based on the analysis in Section II, those texture/geometric features may be selectively sensitive to different variations. However, the current public available RGB-D datasets are almost mixtures of various variations. They are not suitable to evaluate texture/geometric features under a single variation. Thus, we have constructed an RGB-D benchmark dataset with a single variation for the feature evaluation in our approach. Table I gives the details of our dataset.

As shown in Fig. 6, we have three different sets according to the objects in the environment.

- 1) *Texture*: Textured objects such as snack boxes and books.
- 2) *Geometry*: Geometric objects, such as a sculpture, tellurion, and pot plant.
- 3) *Mixture*: The mixture of textured and geometric objects.

TABLE I  
NUMBER OF IMAGES IN THREE SETS

Variance	Texture	Geometry	Mixture
Illumination	11	8	9
Scale	10	10	10
Viewpoint	11	11	11
3D rotation	16	9	10
In-plane rotation	16	12	12

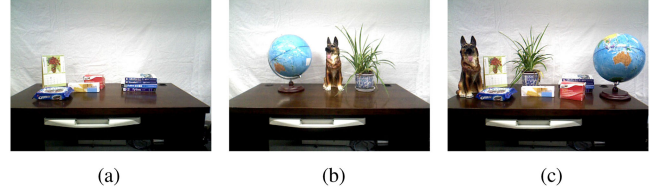


Fig. 6. Example images of our benchmark dataset. (a) *Texture*. (b) *Geometry*. (c) *Mixture*.

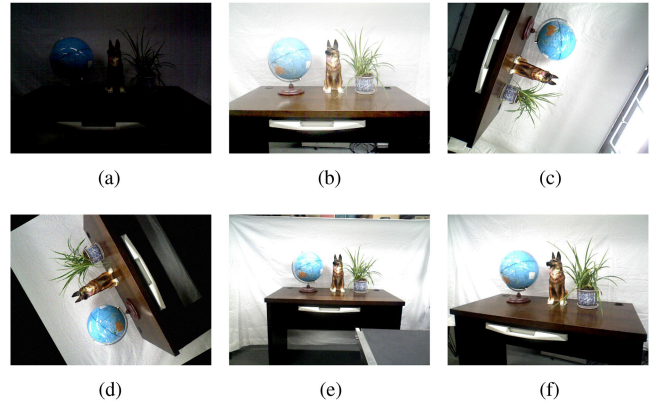


Fig. 7. Example images which show different variation. (a) Natural light illumination (b) Square root illumination (synthetic) (c) 3-D rotation (d) In-plane rotation (e) Scale (f) Viewpoint.

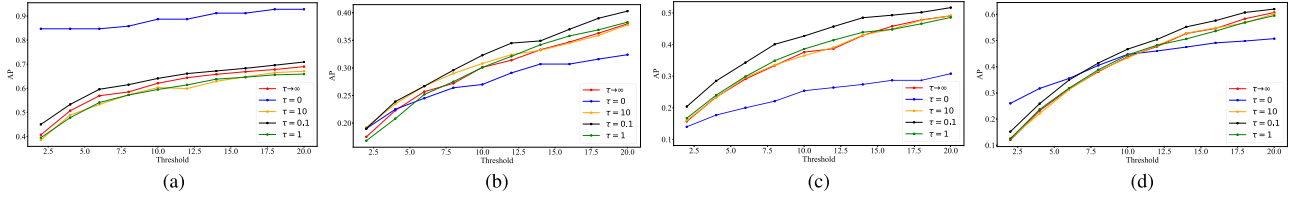
For each of the three sets, we consider the following four common single variation independently, as shown in Fig. 7:

- 1) illumination—synthetic and natural light;
- 2) rotation—in-plane rotation and 3-D rotation;
- 3) scale;
- 4) viewpoint.

## V. EXPERIMENTS AND RESULTS

In this section, we compare the proposed feature TG against state-of-the-art algorithms: Harris detector [31], SIFT [6], ORB [32], RISAS [11], SuperPoint [15], and RFNet [18]. Each consists of a keypoint detector and a feature descriptor except Harris detector. Harris, SIFT, and ORB have been implemented in OpenCV library, while RISAS, SuperPoint, and RFNet have open-source implementations.

Algorithms are evaluated on the built RGB-D evaluation benchmark dataset and the public 3-DMatch RGB-D indoor



**Fig. 8.** Matching performances under single variation with texture information only ( $\tau \rightarrow \infty$ ), geometric information only ( $\tau = 0$ ), more texture information ( $\tau = 10$ ), more geometric information ( $\tau = 0.1$ ), and same proportion of texture information and geometric information ( $\tau = 1$ ). (a) Illumination. (b) Rotation. (c) Scale. (d) Viewpoint.

scene dataset.<sup>2</sup> The later contains many different scenarios, each of which consists of images taken from a video sequence so there are mixed variation between images. Three obviously different scenes are used for evaluation in this article: *rgbd-scenes-v2-scene\_03* (notated as *rgbd03*), *bundlefusion-office1* (notated as *bundle*), and *7-scenes-redkitchen-01* (notated as *7kitchen*).

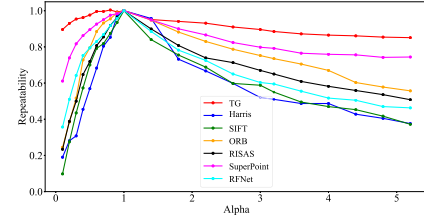
When evaluating the performance of different algorithms, we perform the following operations, respectively.

- 1) Extract keypoints and compute feature vectors for a pair of images. To reduce the impact of the number of keypoints, only about 400 keypoints with the highest scores are taken into consideration.
- 2) Match feature vectors with nearest neighbor distance ratio matching (the ratio is set to 0.95) [33].
- 3) Estimate the homography between this pair of images through random sample consensus [34] as the ground truth.
- 4) Compare the distance  $\varepsilon_i$  of matched points with the threshold  $\xi$ . A match is correct if  $\varepsilon_i < \xi$ ; otherwise, it is an incorrect one.
- 5) Compute the matching accuracy  $P$ , which is the quotient of the number of correct matches and of all matches.
- 6) Perform the abovementioned steps for each pair of images on a dataset to get the average precision AP.

We first perform experiments to determine the value of  $\tau$ . The experimental results are shown in Fig. 8. The horizontal axis is the matching threshold  $\xi$ , and the vertical axis is the average precision AP. When there is only illumination variation in the scene, the depth information does not change. As a result, detectors with geometric information only ( $\tau = 0$ ) performs significantly better than those with more texture information ( $\tau \rightarrow \infty$  and  $\tau = 10$ ). When there is rotation, scale, or viewpoint variation, both the TG change. Overall,  $\tau = 0.1$  achieves the best matching performance under most single variation, which means that a little geometric information can significantly improve the descriptor's performance.

### A. Detector Evaluation

1) **Illumination Variation:** The main advantage of using geometric information in feature detection is the robustness against illumination variation. In this section, we will present the



**Fig. 9.** Repeatability of different detectors under various illumination conditions.

experimental results of the proposed detector under the various illumination conditions.

The critical criterion in evaluating keypoint detector is *repeatability* [35] of keypoints across different images. Given ground truth transformation  $[\mathbf{R}, \mathbf{t}]$  between two images, keypoints are evaluated using (9). In this experiment, we use image pairs that only contain illumination variation, which means there is no pose transformation between their coordinates, i.e.,  $\mathbf{R}p_j + \mathbf{t} = p_j$ . If  $p_j$  is within the neighborhood of  $p_i$  with radius  $\xi$  (5 pixels),  $p_i$  is regarded as *repeated*

$$\|p_i - (\mathbf{R}p_j + \mathbf{t})\| \leq \xi. \quad (9)$$

Fig. 9 shows the robustness of several detectors against illumination variation. It can be seen that with the brightness increases ( $\alpha < 1$ ) or decreases ( $\alpha > 1$ ), the proposed TG detector will show higher repeatability than others. The main reason is that when texture information varies drastically, the depth image still can help TG detector extract robust keypoints from geometric information. However, the repeatabilities of those algorithms, such as the detector of Harris, SIFT, and RFNet, show significantly decrement when texture information becomes weak.

### B. Robustness Evaluation

#### 1) Single Variation:

- 1) **Illumination Invariance:** Fig. 10(a) shows the performances of different algorithms under single illumination variation. The performance of TG is significantly better than three traditional algorithms: SIFT, ORB, and RISAS. Meanwhile, SuperPoint and RFNet also perform well because deep learning algorithms use data augmentation techniques during training, such as changing the brightness of images to improve illumination robustness.

<sup>2</sup>[Online]. Available: <http://3dmatch.cs.princeton.edu/>



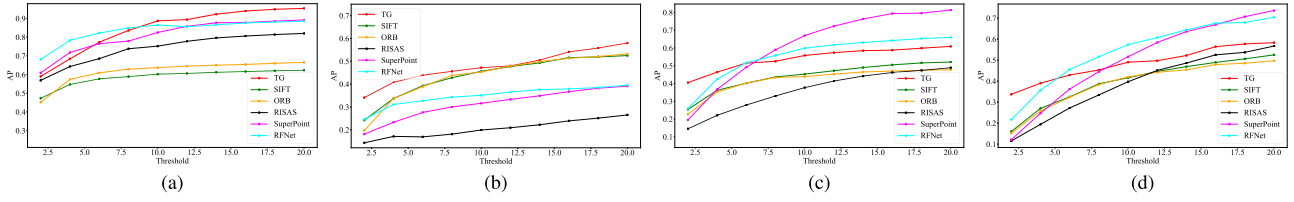


Fig. 10. Matching performances of different algorithms under single variation. (a) Illumination. (b) Rotation. (c) Scale. (d) Viewpoint.

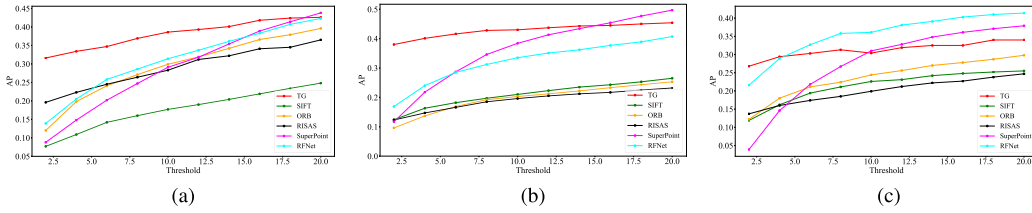


Fig. 11. Matching performances of different algorithms on 3-DMatch datasets. (a) rgbd03 (b) 7kitchen (c) Bundle.

2) *Rotation Invariance*: As shown in Fig. 10(b), the performance of TG is better than other algorithms at each threshold. It is mainly because TG encodes feature in circular blocks rather than square blocks, which can help it effectively resist rotation variation.

3) *Scale Invariance*: Fig. 10(c) shows the performance of different algorithms under single scale variation. TG performs best at low thresholds  $\xi < 5$ , indicating that its localization accuracy is relatively higher than others. Compared with traditional algorithms, TG performs significantly better than SIFT and ORB. One of the main reasons is that TG estimates the scale information of keypoints from depth images. Thereby it can resist the scale variation of the scene. The performance of RISAS is not as good as TG, for our descriptor can encode depth information in a better way. On the other hand, with the increase of the matching threshold, learning-based descriptors show better performances since they have large receptive fields in the network.

4) *Viewpoint Invariance*: As shown in Fig. 10(d), TG performs best at low thresholds  $\xi < 5$ , indicating that TG's localization accuracy is relatively higher than others under viewpoint variation, which may be especially beneficial for the applications of visual odometry. Compared with hand-craft algorithms, the performance of TG is better than SIFT, ORB, and RISAS. One of the main reasons is that TG estimates the local plane normal vector of a keypoint and couples it to the local neighbor point cloud so that the dot product block has the ability to perceive local spatial features.

2) *Mixed Variation*: To evaluate the proposed feature's versatility, we perform experiments on the public 3DMatch RGB-D indoor scene dataset. We use three different scene sets: *rgbd03*, *bundle*, and *7kitchen*, and randomly select 30 pairs of images from each set. Each pair of images is randomly separated by 30 – 50 frames to ensure that they can have

sufficient changes and similarities. Since images of the 3DMatch dataset are consecutive frames taken by videos, each image pair contains mixed variation. Therefore, we can evaluate the performance of different algorithms under mixed variation.

Fig. 11 shows the matching performances of different algorithms on these three datasets. At low thresholds ( $\xi < 10$ ), TG has the highest accuracy than others, but deep learning-based algorithms, RFNet and SuperPoint, do not perform well. In particular, the performance of SuperPoint is significantly affected by the threshold  $\xi$ .

3) *Analysis*: From the experimental results, we can find the following two phenomena when comparing the proposed feature TG with two deep learning algorithms, SuperPoint and RFNet.

- 1) When there is rotation variation between images, deep learning algorithms' performance is poor. They perceive a square block when convolution, while TG encodes circular feature blocks, which is rotation invariant in theory.
- 2) When the matching threshold is low ( $\xi < 5$ ), TG obtains the highest *AP* in almost all experiments. It is more accurate than others in localization.

### C. Efficiency Analysis

In this section, we analyze the efficiency of different algorithms on the built *mixture* dataset. The evaluation platform is Intel i7 7700K and 16 GB RAM.

As given in Table II, TG takes 31.1 ms to extract keypoints and calculate feature vectors for an image with the C++ implementation. The average processing time for each keypoint is 0.0975 ms. It shows that the TG has relatively high computational efficiency. On the other hand, ORB and SIFT in the OpenCV library are implemented with parallel technology, significantly improving computational efficiency. If parallel technology is used in TG, it will be faster.

**TABLE II**  
COMPUTATIONAL EFFICIENCY OF DIFFERENT ALGORITHMS

Algorithm	Code	Image time (s)	Number of points	Point time (ms)
TG	C++	0.0311	319	0.0975
SIFT	C++	0.0881	400	0.2203
ORB	C++	0.0057	400	0.0143
RISAS	Python	67.729	278	243.19
SuperPoint (CPU)	Python	0.3917	468	0.8356
SuperPoint (GPU)	Python	0.1736	468	0.3709
RfNet (CPU)	Python	2.5859	400	6.4648
RfNet (GPU)	Python	0.2944	400	0.7360

**TABLE III**  
AVERAGE TRAJECTORY ERROR(M) ON THREE DATASETS WITH DIFFERENT DESCRIPTORS

Dataset	ORB	SIFT	TG	SuperPoint*	RfNet
<i>bundle</i>	0.096	0.086	<b>0.078</b>	0.165	0.126
<i>7kitchen</i>	0.145	<b>0.100</b>	0.103	0.103	0.167
<i>rgb03</i>	0.086	0.071	0.060	-	<b>0.057</b>
Mean	0.109	0.086	<b>0.080</b>	-	0.117

\*SuperPoint fails on the *rgb03* sequence.

Bold entity represents the best performance of multiple methods in each sequence.

#### D. Visual Odometry

We employ different descriptors in visual odometry to evaluate their performance in practical applications. When a new frame arrives, we extract feature points, calculate descriptors, and get their depth in the depth map. After feature matching, we perform bundle adjustment to optimize the camera pose.

Table III gives the absolute trajectory error of the estimated trajectories with different algorithms in three datasets. TG reaches the best performance on *bundle* dataset, and overall has the smallest mean error in these three scenarios.

As given in Table II, SIFT and RfNet are slower than TG, so that they are not suitable for real-time applications. In visual odometry, ORB is the most commonly used feature with high computational efficiency. Table III tabulates that the localization error of TG is significantly smaller than ORB. Since TG also has relatively high computational efficiency, it could be an alternative to ORB in current visual odometry systems.

#### VI. CONCLUSION

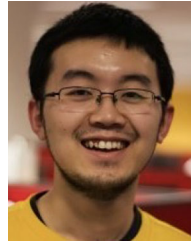
This article proposes an RGB-D fused feature TG that consists of a keypoint detector and a feature descriptor. We use a simplified Gaussian image pyramid to perceive texture information and the DoG pyramid to extract texture features. The gradient of the point cloud is encoded as geometric features, and the autocorrelation function is performed to fuse both TG. Future work will focus on further experimental evaluation and bringing it into more SLAM systems.

#### REFERENCES

- [1] Y. Chen, S. Huang, and R. Fitch, "Active SLAM for mobile robots with area coverage and obstacle avoidance," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 3, pp. 1182–1192, Jun. 2020.
- [2] V. Kubelka, M. Reinstein, and T. Svoboda, "Tracked robot odometry for obstacle traversal in sensory deprived environment," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 6, pp. 2745–2755, Dec. 2019.
- [3] H. Kim and B. Lee, "Robust 3-D object reconstruction based on camera clustering with geodesic distance," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 2, pp. 889–891, Apr. 2019.
- [4] L. Jin, H. Zhang, and C. Ye, "Camera intrinsic parameters estimation by visual-inertial odometry for a mobile phone with application to assisted navigation," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 4, pp. 1803–1811, Aug. 2020.
- [5] Z. Lin *et al.*, "ARei: Augmented-reality-assisted touchless teleoperated robot for endoluminal intervention," *IEEE/ASME Trans. Mechatronics*, early access, Sep. 2021, doi: [10.1109/TMECH.2021.3105536](https://doi.org/10.1109/TMECH.2021.3105536).
- [6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [7] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.
- [8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.
- [9] E. Nascimento, G. Oliveira, M. Campos, A. Vieira, and W. Schwartz, "Brand: A robust appearance and depth descriptor for RGB-D images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 1720–1726.
- [10] G. Feng, Y. Liu, and Y. Liao, "LOIND: An illumination and scale invariant RGB-D descriptor," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 1893–1898.
- [11] K. Wu, X. Li, R. Ranasinghe, G. Dissanayake, and Y. Liu, "Risas: A novel rotation, illumination, scale invariant appearance and shape feature," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 4008–4015.
- [12] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 778–792.
- [13] T. L. Xufeng Han and Y. Jia, "MatchNet: Unifying feature and metric learning for patch-based matching," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3279–3286.
- [14] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4826–4837.
- [15] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 224–236.
- [16] S. Li, S. Liu, Q. Zhao, and Q. Xia, "Quantized self-supervised local feature for real-time robot indirect VSLAM," *IEEE/ASME Trans. Mechatronics*, early access, Jun. 2021, doi: [10.1109/TMECH.2021.3085326](https://doi.org/10.1109/TMECH.2021.3085326).
- [17] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "Lf-net: Learning local features from images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6234–6244.
- [18] X. Shen *et al.*, "RF-Net: An end-to-end image matching network based on receptive field," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8132–8140.
- [19] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops*, 2009, pp. 689–696.
- [20] A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes," *Int. J. Comput. Vis.*, vol. 89, no. 2–3, pp. 348–361, 2010.
- [21] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3d feature matching," in *Proc. IEEE Int. Conf. Image Process.*, 2011, pp. 809–812.
- [22] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3Dmatch: Learning local geometric descriptors from RGB-D reconstruction," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 199–208.
- [23] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab, "Deep learning of local rgb-d patches for 3D object detection and 6d pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 205–220.
- [24] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik, "Aligning 3D models to RGB-D images of cluttered scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4731–4740.



- [25] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 345–360.
- [26] Y. Cheng, R. Cai, Z. Li, X. Zhao, and K. Huang, "Locality-sensitive deconvolution networks with gated fusion for RGB-D indoor semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3029–3037.
- [27] J. Berkmann and T. Caelli, "Computation of surface geometry and segmentation using covariance techniques," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 11, pp. 1114–1116, Nov. 1994.
- [28] R. Horaud and F. Dornaika, "Hand-eye calibration," *Int. J. Robot. Res.*, vol. 14, no. 3, pp. 195–210, 1995.
- [29] K. Daniilidis, "Hand-eye calibration using dual quaternions," *Int. J. Robot. Res.*, vol. 18, no. 3, pp. 286–298, 1999.
- [30] R.-H. Liang and J.-F. Mao, "Hand-eye calibration with a new linear decomposition algorithm," *J. Zhejiang Univ. Sci. A*, vol. 9, no. 10, pp. 1363–1368, 2008.
- [31] C. G. Harris *et al.*, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 10–5244.
- [32] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.
- [33] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
- [34] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," in *Readings in Computer Vision*. New York, NY, USA: Elsevier, 1987, pp. 726–740.
- [35] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *Int. J. Comput. Vis.*, vol. 37, no. 2, pp. 151–172, 2000.



algorithms.

**Kanzhi Wu** received the B.S. and M.S. degrees in vision-based navigation from Northwestern Polytechnical University, Xi'an, China, in 2010 and 2013, respectively, and the Ph.D. degree in robotics from center for autonomous system from Center for Autonomous Systems, University of Technology Sydney, Ultimo, NSW, Australia, in 2017, under the supervision of Prof. Gamini Dissanayake.

His current research interests include SLAM, sensor fusion for state estimation, and planning



**Gamini Dissanayake** (Senior Member, IEEE) received the B.Sc.(Eng.) degree in mechanical/production engineering from the University of Peradeniya, Peradeniya, Sri Lanka, in 1977, the M.Sc. degree in machine tool technology, and the Ph.D. degree in mechanical engineering from the University of Birmingham, Birmingham, U.K., in 1981 and 1985, respectively.

He is the James N Kirby Distinguished Professor of mechanical and mechatronic engineering with the University of Technology Sydney, Ultimo, NSW, Australia. His research interests include SLAM, navigation systems, dynamics, and control of mechanical systems, cargo handling, and path planning.

**Yong Liu** (Member, IEEE) received the B.S. degree in computer science and engineering and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2001 and 2007, respectively.

He is currently a Professor with the Department of Control Science and Engineering, Zhejiang University. His current research interests include machine learning, robotics vision, information processing, and granular computing.



**Xiangrui Zhao** (Graduate Student Member, IEEE) received the B.S. degree in automation from the Huazhong University of Science and Technology, Wuhan, China, in 2018. He is currently working toward the Ph.D. degree in control science and engineering with the Department of Control Science and Engineering, Institute of Cyber Systems and Control, Zhejiang University, Hangzhou, China.

His current research interests include robotics vision and SLAM systems.



**Yu Liu** (Student Member, IEEE) received the B.S. degree in automation from China Agricultural University, Beijing, China, in 2017 and the M.S. degree in control engineering from Zhejiang University, Hangzhou, China, in 2020.

She is currently with Huawei Technologies Company, Ltd. Group, Shenzhen, China. Her current research interests include visual inertial odometry and multiple sensor fusion.



**Zhengbo Wang** (Student Member, IEEE) received the B.S. degree in automation from Harbin Engineering University, Harbin, China, in 2019. He is currently working toward the M.S. degree in control science and engineering with Zhejiang University, Hangzhou, China.

His current research interests include robotics vision and SLAM systems.