Received: 6 December 2017 | Revised: 4 May 2018 | Accepted: 16 July 2018

DOI: 10.4218/etrij.2017-0298

ORIGINAL ARTICLE

WILEY ETRI Journal

Fast 3D reconstruction method based on UAV photography

Jiang-An Wang | **Huang-Te Ma** | **Chun-Mei Wang** | **Yong-Jie He**

School of Information Engineering, Chang'an University, Xian, Shanxi, China.

Correspondence
Jiang-An Wang, School of Information Engineering, Chang'an University, Xian, Shanxi, China.
Email: wangjiangan@126.com

Funding information
Natural Science Foundation of Shaanxi Province, China (2017JQ6048); Fundamental Research Funds for the Central Universities, China (310824161009), Postdoctoral Science Foundation of China (2015M582589); Guangxi Key Laboratory of Precision Navigation Technology and Application, Guilin University of Electronic Technology, China (DH201711).

3D reconstruction of urban architecture, land, and roads is an important part of building a "digital city." Unmanned aerial vehicles (UAVs) are gradually replacing other platforms, such as satellites and aircraft, in geographical image collection; the reason for this is not only lower cost and higher efficiency, but also higher data accuracy and a larger amount of obtained information. Recent 3D reconstruction algorithms have a high degree of automation, but their computation time is long and the reconstruction models may have many voids. This paper decomposes the object into multiple regional parallel reconstructions using the clustering principle, to reduce the computation time and improve the model quality. It is proposed to detect the planar area under low resolution, and then reduce the number of point clouds in the complex area.

KEYWORDS
3D reconstruction, clustering, feature extraction, PMVS, UAV

1 | INTRODUCTION

The demand for 3D data in geographic information, urban construction, national defense, and other fields is growing [1–3]. Traditional methods cost a lot of manpower, material, and financial resources, and usually incur a large error [4–7]. Satellites are too high from the ground, and the image quality cannot be compared with that of a close range photograph [8,9]. Large aircraft using the high-resolution camera method can produce good quality images, but the costs, landing space, and air route applications make 3D reconstruction very complex [10–14].

The characteristics of small size, low energy consumption, and high mobility make UAV widely used in various fields [15–17]. Compared with the traditional photogrammetry and remote sensing platform, low-altitude flight UAV and low-cost camera combination meet the 3D image accuracy requirements of digital city construction [18]. Using UAV flight formations, many UAVs can be used in a region for data collection. UAV flight platforms have greatly enhanced the efficiency of data collection [19–21]. Therefore, the UAV is the best platform for 3D reconstruction of large scenes at this stage.

At present, 3D reconstruction results based on multi-view images are prone to large voids [22]. Corner and angular object are rebuilt into an arc. Trunk and other smaller cylindrical objects are rebuilt with the phenomenon of displacement, and so on [23,24]. These defects occur primarily because point clouds generated by the 3D reconstruction are not dense and accurate enough, and the details of the scene cannot be highlighted in the point cloud [25].

This paper focuses on UAV 3D reconstruction by the dense reconstruction algorithm. By increasing the number of point clouds and improving the accuracy of point cloud, the voids and reconstruction errors can be reduced, and the reconstruction time can be shortened.

2 | IMPROVEMENT OF 3D RECONSTRUCTION ALGORITHM BASED ON PMVS

The Patch-based Multi-view Stereo (PMVS) algorithm is an algorithm for spreading feature area, as shown in Figure 1. Initial information, such as convex hull, encircling box, and the basic assumptions of the parallel views are not necessary. It only needs to estimate the depth of the reconstructed point cloud and the normal vector that is constrained by the illumination consistency.

The input image is divided into same $\mu \times \mu$ ($\mu = 32$) pixel image blocks. Harris features and DoG features are extracted from the image block, the eigenvalues are sorted, and the pre-β($\beta = 4$) bits with large eigenvalues are selected. After the feature is extracted, the corner points and feature points are obtained. The strongest feature response of each image block is η. When η is smaller, the feature points are denser, and the computational complexity is larger. After the feature points are extracted, the images are matched with each other, and the initial sparse patches are constructed and stored in the grid $C(i, j)$.

Each image block takes turns as a reference image $R(P)$. Calculating the angle between the optical axis of $R(P)$ and the optical axis of the other images; when the angle is less than 60°, the image is selected as $I(P)$. Thereby, we select the reference image and the image to be matched. According to the limit constraint condition (the matching point satisfies the distance of the pole line within two pixels), the matching point of each image and the matching image is selected. Then, for each pair of matching points (f, f'), the 3D point coordinates were obtained by the coordinate mapping relationship.

$$\frac{1}{z_c}\cdot\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}\frac{f}{s_x}&r&u_0\\0&\frac{f}{s_y}&v_0\\0&0&1\end{bmatrix}\cdot\begin{bmatrix}\mathbf{R}_{3\times3}&\mathbf{T}_{3\times3}\\0&1\end{bmatrix}\cdot\begin{bmatrix}X_W\\Y_W\\Z_W\\1\end{bmatrix}=\mathbf{K}_{3\times3}\cdot\begin{bmatrix}\mathbf{R}_{3\times3}&\mathbf{T}_{3\times3}\\0&1\end{bmatrix}\cdot\begin{bmatrix}X_W\\Y_W\\Z_W\\1\end{bmatrix}, \tag{1}$$

$$s\cdot\begin{bmatrix}u\\v\\1\end{bmatrix}=\mathbf{M}_{3\times4}\cdot\begin{bmatrix}X_W\\Y_W\\Z_W\\1\end{bmatrix}. \tag{2}$$

$\mathbf{K}_{3\times3}$ represents the camera internal parameter matrix (including five internal parameters). $\mathbf{M}_{3\times4}$ represents the perspective projection matrix. $S = 1/z_c$ is an unknown scale factor. Matrix $\mathbf{P}$ is known in this paper.

Setting the patch as P, the 3D space point $\underline{c}(P)$ is initialized with the patch initial center (f, f'). The patch normal vector $n(P)$ points to the camera optical center of the reference image $R(P)$. By using $c(P)$ and $n(P)$ as variables, the photometric difference value $g(P)$ of the patch is minimized by the conjugate gradient method. The minimum patch is selected as the final patch for this feature point. The photometric consistency function $h(P, I, R(P))$ is calculated, and then $V(P)$ and $g(P)$ are updated by the constraint $V^*(P) = \{I | I \in V(P), h(P, I, R(P)) \leq \alpha\}$. If there are at least γ pictures of little difference in brightness of the picture, the patch is successful. Save patch P in image blocks $V(P)$ and $V^*(P)$, and then update sets $Q_i(x, y)$ and $Q_i^*(x, y)$. Finally, the optimal patch is selected as the final patch with a feature point, and the point is selected as the seed dots. During implementation, the reconstructed patch P is used instead of the original patch in the image block, to reduce the computational complexity and improve program efficiency.

The kernel of patch spread is used to spread the existing patches to neighborhood image block $C_i(x, y)$. Finally, each patch is reconstructed in each image block. The cells are composed of adjacent visible images of patch P. The expression of the image cell is as follows:

$$C(P) = \{C_i(x', y') | p \in Q_i(x, y), |x - x'| + |y - y'| = 1\}. \tag{3}$$

After the seed patch is initialized, the PMVS algorithm spreads the seed patch outwards until a patch is created in all image blocks. To increase the number of patches, the spread adds seed patches to the neighborhood image block; each added patch needs to find its cells. Repeat the previous steps, until all the visible surface in the scene are covered.

When patch P' is spread from seed patch P, the value of patch P is used to initialize $n(P')$, $R(P')$, and $V(P')$. $C(P')$ is initialized with the intersection of the light path in the neighborhood grid center and the plane where the seed patch is located. Then $C(P')$ and $n(P')$ are optimized by the conjugate gradient method. In the optimization process, it is necessary to constrain $C(P')$ on the same straight line, so the projection of $C(P')$in the image is fixed, and correspondence between

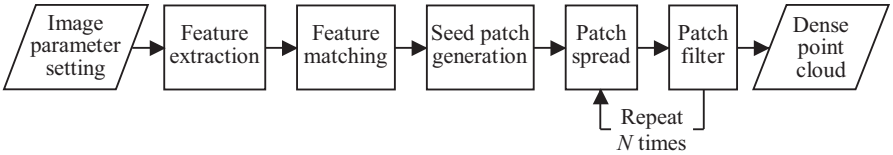


FIGURE 1 PMVS algorithm flowchart

image block $C_i(x, y)$and patch P' is determined. Finally, we need to save the visible image $V(P')$ and update image set $V^*(P)$. After patch spread is completed, the dense point cloud contains incorrect points. Therefore, there are three patch filtering constraints as follows:

1. Visible consistency constraints. The spread patches may fall outside the scene.

$$|V^*(P)|(1 - g^*(P)) < \sum_{P_1 \in U(P)}(1 - g^*(P_i)). \quad (4)$$

   Equation (4) can determine whether the patch is outside or not, where $U(P)$ is a set of patches P_i that are not adjacent to the current patch P, but in the same image cell.
2. Number of visible images. For each patch P, according to depth mapping test, P is visible in $V(P)$, and the number of images in $V(P)$ is calculated. If the number of images is less than the threshold, P is filtered out.
3. Neighborhood constraint. For each patch P, all patches are selected for its cells and the adjacent cells from $V(P)$. If adjacent patches account for less than 0.25 (the threshold is 0.25) of the selected patches, P is filtered out.

3 | IMPROVEMENT OF CLUSTER-BASED PMVS ALGORITHM

Cluster obtains high-resolution images by classifying input images and removing redundant images. The images are clustered into multiple clusters. The large scene 3D reconstruction is decomposed into multiple small area 3D reconstructions, so the 3D reconstruction efficiency is improved by parallel computing.

3.1 | Clustering constraints

1. Density: The definition of density was used to complete the 3D reconstruction with a minimum number of images, and to eliminate redundant images.
   Function $f(P, C)$ represents point P_j in cluster C. SFM points can be reconstructed in at least one cluster, if it satisfied the formula

$$\max(f(P_j, C_k \cap V_j)) \geq \lambda f(P_j, V_j). \quad (5)$$

   V_j is the visible image set of P_j, and λ is the visible image threshold of the current SFM point.
2. Size: To ensure that each cluster can be fully reconstructed, the number of images in a cluster must be enough to reconstruct SFM point, and the number of images cannot exceed the preset threshold.
3. Coverage: To rebuild each cluster, the details of the image must maintain integrity. For any cluster I_k, formula $I_i(C)/I_i(F) \leq \tau$ must be satisfied, where τ is an input threshold.

3.2 | Clustering algorithm

After sparse reconstruction, the sparse point cloud (SFM point set) is huge. Merging adjacent SFM points into one point can effectively reduce the amount of data and increase the speed of operation. According to the second clustering constraint, each SFM point needs to be included by at least one cluster. Therefore, redundancy images can be merged and removed.

When the size of a cluster is large, the cluster will be divided into smaller clusters until the size satisfies the clustering size constraint. The standard segmentation method [26] was used in the clustering algorithm. Set an image as a node, the image nodes with the same region are clustered into a class. The edge weights e_{lm} of image (I_l, I_m) represent the contribution to SFM points reconstruction.

$$e_{lm} = \sum_{P_j \in \theta} lm \frac{f(P_j, \{I_l, I_m\})}{f(P_j, V_j)}, \quad (6)$$

where θ^{lm} is a set of SFM points that are visible for image I_l and I_m. With (6), it is concluded that the greater contribution of an image to the SFM point reconstruction, the less the possibility for the image to be segmented. According to the clustering size constraints, the corresponding image set is designated as a cluster, until all the clusters are within the constraints.

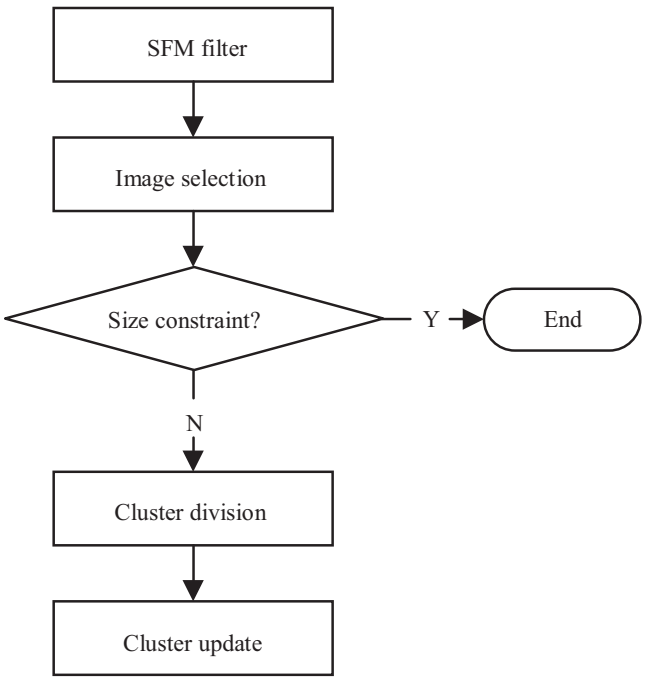


FIGURE 2 Cluster flowchart

FIGURE 3 Input images

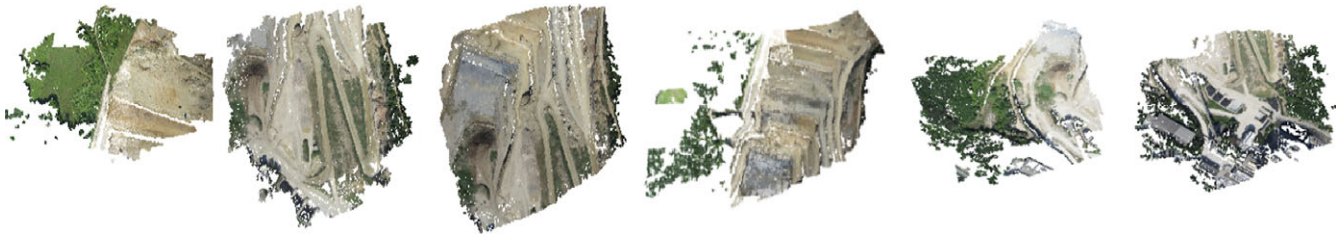


FIGURE 4 Reconstruction model of each region

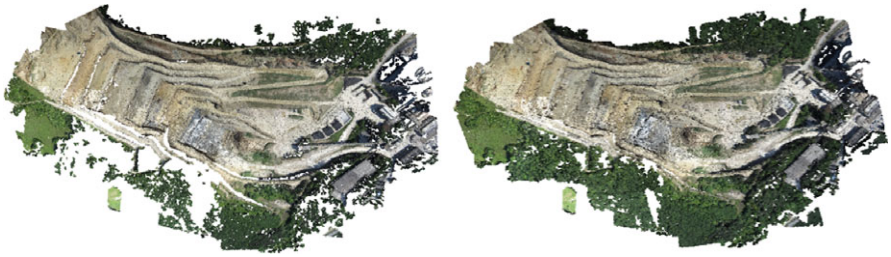


FIGURE 5 Reconstruction model

TABLE 1 Reconstruction time comparison

Scene	Original algorithm reconstruction time/s	Improved average reconstruction time/s				Compared with the original algorithm, the time ratio is saved under the dual thread
		Single thread	Dual thread	Three threads	Four threads	
Quarry	669.7	639.2	581.6	510.3	472.8	13.2%

In the cluster classification step, the coverage constraint is not considered. Therefore, after the first three steps, there are some SFM points that are not covered by any cluster. Add a new image to a cluster that does not satisfy the coverage constraint to ensure that the cluster covers more SFM points. The steps are shown in Figure 2:

1. For the SFM points P_j, which are not covered by any clusters, $C_k = \arg\max f(P_j, C_i)$ is used to determine a cluster with maximum reconstruction contribution.
2. Establish $\{(I \rightarrow C_k), g\}$ with P_j, which means adding the image $I(\in V_j, \in C_k)$ to C_k, where g is the measurement of the validity. g is defined as $f(P_j, C_k \cup \{I\}) - f(P_j, C_k)$, which is the difference in f before and after the image I is added.
3. Create a list of adding image operation in descending order by g.

After the list is created, image I is added to C_k. To reduce computation load, an image can be added only if its g value is more than 0.7 times the maximum in the list. The value of g in the list changes after adding an image, so the list should be updated.

After adding a new image to a cluster, the size constraint may no longer be satisfied. Therefore, classifying and updating of the cluster need to be performed cyclically until all the clusters satisfy these constraints.

4 | EXPERIMENTS

The experiment used a four-rotor UAV and camera designed by us. The resolution of the camera is 20M pixels, and the shooting area is a quarry. The experiment used 128 images as shown in Figure 3.

After clustering, the 3D reconstruction of the PMVS algorithm is carried out. The reconstruction model of each region is shown in Figure 4.

The combination of the 3D model of each region is a complete model. The 3D model before and after our improved algorithm processing is shown in Figure 5.

The original algorithm to obtain dense point cloud has large voids. After our improved algorithm processing, the cloud density of the model is improved greatly. The original method produced the large voids in the reconstruction model (dense point cloud) but there are many points in the void area in the sparse point cloud. The clusters are very large in the original method, so it is difficult to match. The proposed method has smaller cluster and it is easier to match, so the void number and area can be reduced in the dense point cloud. Scene boundaries and details generated from the point cloud are more extensive.

The 3D reconstruction time is shortened by using the improved algorithm. The algorithm decomposes the object into multiple regions using multi-thread simultaneous reconstruction, which saves the reconstruction time, as illustrated in Table 1.

5 | CONCLUSIONS

The algorithm has improved the density and accuracy of the point cloud compared with the original algorithm. There is a significant reduction in the number of 3D model voids. Parallel computation is used, and the reconstruction time is relatively reduced. Experiments show that the algorithm proposed in this paper has better results for large scene 3D reconstruction.

ORCID

Jiang-An Wang http://orcid.org/0000-0003-4743-5354

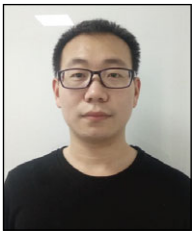
REFERENCES

1. W. Heldens et al., Suitability of remote sensing based surface information for a three-dimensional urban microclimate model, *IEEE Int. Geosci. Remote Sensing Symp.* (IGARSS), Beijing, China, July 10–15, 2016, pp. 7322–7325.
2. D. Marr, *Artificial intelligence – A personal view*, Artif. Intell. **9** (1977), no. 1, 75–108.
3. O. Faugeras and R. Keriven, *Variational principles, surface evolution, PDEs, level set methods, and the stereo problem*, IEEE Trans. Image Process. **7** (1998), no. 3, 336–344.
4. C. Tomasi and T. Kanade, *Shape and motion from image streams under orthography: a factorization method*, Int. J. Comput. Vision **9** (1992), no. 2, 137–154.
5. P. E. Debevec, C. J. Taylor, and J. Malik, Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach, *Proc. Annu. Conf. Comput. Graphics interactive techn.*, New Orleans, LA, USA, Aug. 4–9, 1996, 11–20.
6. R. Koch et al., A geometric approach to light field calibration, *Comput. Anal. Images Patterns: Int. Conf.*, CAIP'99, Ljubljana, Slovenia, Sept. 1–3, 1999, pp. 596–603.
7. S. Agarwal et al., Building Rome in a day, *IEEE Int. Conf. Comput. Vision*, Kyoto, Japan, Sept. 29–Oct. 2, 2009, pp. 72–79.
8. Y. Furukawa and J. Ponce, Accurate, dense, and robust multi-view stereopsis, *IEEE Conf. Comput. Vision Pattern Recogn.*, Minneapolis, MN, USA, June 17–22, 2007, pp. 1–8.
9. Y. Furukawa et al., Towards Internet-scale multi-view stereo, *IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn.*, San Francisco, CA, USA, June 13–18, 2010, pp. 1434–1441.
10. J. Schöning and G. Heidemann, Evaluation of multi-view 3D reconstruction software, *Comput. Anal. Images Patterns: Int. Conf., (CAIP)*, Valletta, Malta, Sept. 2–4, 2015, pp. 450–461.
11. S. Corazza et al., *Markerless motion capture through visual hull, articulated ICP and subject specific model generation*, Int. J. Comput. Vision **87** (2009), no. 1, 156–158.
12. M. Goesele et al., Multi-view stereo for community photo collections, *IEEE Int. Conf. Comput. Vision*, Rio de Janeiro, Brazil, Oct. 14–21, 2007, pp. 1–8.
13. D. Bradley, T. Boubekeur, and W. Heidrich, Accurate multi-view reconstruction using robust binocular stereo and surface meshing, *IEEE Conf. Comput. Vision Pattern Recogn.*, Anchorage, AK, USA, June 23–28, 2008, pp. 1–8.
14. W. Tao, Y. Lei, and P. Mooney, Dense point cloud extraction from UAV captured images in forest area, *Proc. IEEE Int. Conf. Spatial Data Mining Geographical Knowledge Services*, Fuzhou, China, June 29–July 1, 2011, pp. 389–392.
15. W. Burger and M. J. Burge, *Scale-invariant feature transform (SIFT), digital image processing: An algorithmic introduction using java*, Springer, London, UK, 2016, pp. 609–664.
16. S. Agarwal et al., Bundle adjustment in the large, *Eur. Conf. Comput. Vision*, Heraklion, Greece, Sept. 5–11, 2010, pp. 29–42.
17. S. R. Gunn, Edge detection error in the discrete Laplacian of Gaussian, *Proc. Int. Conf. Image Process.*, Chicago, IL, USA, Oct. 7, 1998, pp. 515–519.
18. W. Cai et al., A 3D difference-of-Gaussian-based lesion detector for brain PET, *IEEE Int. Symp. Biomedical Imaging (ISBI)*, Beijing, China, Apr. 29–May 2, 2014, pp. 677–680.

19. É. Brier and M. Joye, Weierstraß elliptic curves and side-channel attacks, *Int. Workshop Practice Theory Public Key Cryptosystems*, Paris, France, Feb. 12–14, 2002, pp. 335–345.
20. E. Tola, C. Strecha, and P. Fua, *Efficient large-scale multi-view stereo for ultra-high-resolution image sets*, Mach. Vision Applicat. **23** (2012), no 5, 903–920.
21. M. Hofer, M. Maurer, and H. Bischof, Improving sparse 3D models for man-made environments using line-based 3D reconstruction, *Int. Conf. 3D Vision*, Tokyo, Japan, Dec. 8–11, 2014, pp. 535–542.
22. M. Mauro et al., An integer linear programming model for view selection on overlapping camera clusters, *Int. Conf. 3D Vision*, Tokyo, Japan, Dec. 8–11, 2014, pp. 464–471.
23. L. Lou et al., Accurate multi-view stereo 3D reconstruction for cost-effective plant phenotyping, *Image Anal. Recog.: Int. Conf.*, Vilamoura, Portugal, Oct. 22–24, 2014, pp. 349–356.
24. F. Wang and N. An, Accurate multi-view stereopsis fusing DAISY descriptor and scaled-neighbourhood patches, *Adv. Multimedia Inform. Process.: Pacific-Rim Conf. Multimedia*, Xian, China, Sept. 15–16, 2016, pp. 260–270.
25. C. Strecha et al., On benchmarking camera calibration and multi-view stereo for high resolution imagery, *IEEE Conf. Comput. Vision Pattern Recogn.*, Anchorage, AK, USA, June 23–28, 2008, 1–8.
26. N. D. F. Campbell et al., *Automatic 3D object segmentation in multiple views using volumetric graph-cuts*, Image Vision Comput. **28** (2010), no. 1, pp. 14–25.

AUTHOR BIOGRAPHIES

Jiang-An Wang received his BS degree in telecommunication engineering from the School of Telecommunication Engineering, Xidian University, Xi'an, China, in 2003, and his MS and PhD degrees in Electronic Science and Technology from the School of Micro-electronics, Xidian University, Xi'an, China, from 2006 to 2010. Since 2010, he has been with the School of Information Engineering, Chang'an University, Xi'an, China, where he is now a lecturer. His main research interests are navigation and computer vision.



Huang-Te Ma received his BS degree in electronic information science and technology from the School of Information Science and Engineering, Shannong University, Shandong, China, in 2016. Since 2016, he has been studying at the School of Information Engineering, Chang'an University, Xi'an, China. His main research interest is the image processing.

Chun-Mei Wang received her BS degree in electronic information science and technology from the School of Mathematics and Physics, Weinan Normal University, Xi'an, China, in 2017. Since 2017, she has been with the School of Information Engineering, Chang'an University, Xi'an, China, where she is now a student. Her main research interest is computer vision.

Yong-Jie He received his BS degree in electronic information science and technology from the School of Information Science and Engineering, Northwestern Polytechnical University Mingde College University, Xi'an, China, in 2017. Since 2017, he has been studying at the School of Information Engineering, Chang'an University, Xi'an, China. His main research interest is image processing.