Multiview Scene Image Inpainting Based on Conditional Generative Adversarial Networks

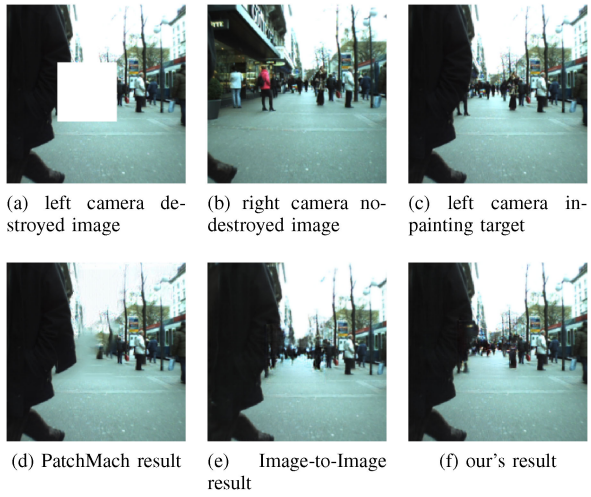
Zefeng Yuan, Hengyu Li, Jingyi Liu, and Jun Luo

***Abstract*—With the help of a multiview system, an unmanned vehicle system can better understand the surrounding environment and choose a more accurate and safer path to avoid obstacles. However, due to the interference of the signal or the loss of part of the signal during the acquisition, processing, compression, transmission and decompression of the video image signal, the local area of the image is abnormal, which affects the perception and decision of the system. This article addresses the problems of inaccurate restored images and noise in the restored images by proposing an image restoration method that is applied to a multicamera system. We utilize different perspective images captured by different cameras to assist and constrain the restoration of the damaged image. This method restores the image by combining sample representations and sample distribution models which respectively based on self-encoder reconstruction loss learning and generative adversarial networks. In this method, the infrastructure is a conditional generative adversarial network, the condition is the images that are from the other perspectives, and the generator is a self-encoder structure with cross-layer connection, group convolution and feature map channel exchanged. This method was carried out on a dataset recorded in Zurich using a pair of cameras mounted on a mobile platform. The experimental results demonstrate that the proposed method is superior to the existing methods in terms of mean L1 Loss, mean L2 Loss and the peak signal to noise ratio (PSNR).**

***Index Terms*—Convolutional neural network, deep learning, generative adversarial networks, image inpainting.**

I. Introduction

With the development of image and video processing technology, visual information has played a key role in the field of automation [1]–[3]. Due to the limited information available from monocular cameras, the multiview system is widely used in navigation [4], panorama [5], occlusion handing and vehicle classification [6], object detection [7]–[9] and tracking [10], [11]. The interference or loss of video image signal during the process of acquisition, compression, translation and decompression will affect the system perception and judgment of the surroundings, which can cause accidents. Image inpainting restores the image based on information such as the texture of the edge of the damaged area, which can show us desired information. This technology has been widely used in many fields, including damaged image restoration [12]–[14], video transmission error repair [15], [16], and image editing [17], [18].



(a) left camera destroyed image (b) right camera no-destroyed image (c) left camera inpainting target

(d) PatchMach result (e) Image-to-Image result (f) our's result

Fig. 1. Qualitative illustration of the different image inpainting methods.

Image restoration techniques can be divided into two types depending on whether the algorithm relies on data other than the damaged image itself. The first type of method relies on information such as the texture structure of the damaged image, and expands and fills the texture structure similar to the pixels around the damaged area to repair the image [12], [19]–[21]. These techniques commonly use patches with similar textures to synthesize the content of the whole region from coarse to fine. Drori *et al.* [19] and Wilczkowiak *et al.* [20] introduced multiple scales and orientations to find better matching patches. Barnes *et al.* [21] used the fast-approximate nearest neighbor algorithm to search the match patches. The first type of method excels at extending high-frequency texture details, and the partially defective target can be recovered. However, when the small target object in the defect area is completely missing, the texture synthesis method cannot work because the remaining areas of the image lack information about the small object. The Fig. 1 shows the image restoration results of three different methods, where Fig. 1(a) is the image to be repaired and Fig. 1(c) is the image of the repair target to be achieved. Fig. 1(d) is the repair result of Bornes *et al.* [21]. Comparing the repair result shown

Manuscript received December 4, 2018; revised June 26, 2019; accepted November 1, 2019. Date of publication November 25, 2019; date of current version May 25, 2020. This work was supported in part by the Natural Science Foundation of China under Grant 61525305 and Grant 61625304, in part by the Shanghai Natural Science Foundation under Grant 17ZR1409700 and Grant 18ZR1415300, and in part by the basic research project of Shanghai Municipal Science and Technology Commission under Grant 16JC1400900. *(Corresponding author: Hengyu Li.)*

The authors are with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China (e-mail: yuanzefengde@163.com; lihengyu@shu.edu.cn; jingyiliu1991@foxmail.com; luojun@shu.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIV.2019.2955907

in 1(d) with the repair result shown in 1(c), we find that the missing part of the black coat is recovered, and the missing tiny characters are filled with the surrounding road and sky around the missing area; this target information is lost in the repair result. The second set of approaches solves this problem in a data-driven manner. This type of method can be divided into traditional methods and deep network-based methods. The core idea of the early approach is to find the image blocks that match the damaged image from the image dataset to fill the damaged area. For example, Hays *et al.* [22] involved a cut-paste formulation using nearest neighbors from a dataset of millions of images to find the matched image block. The repair effect of this method only makes people feel that the image is not damaged, while it ignores the content that the original image should have. In addition, this method is not effective for the local missing repair of objects. When the object is partially missing, it is difficult to repair the missing object portion and align the entire object content at the seam. Based on deep learning, image restoration methods [23]–[25] use the self-encoder to learn the content and semantic expression of the scene and use the generated confrontation network to learn the distribution model of the scene. Combined with the reconstruction loss of the autoencoder and the generative adversarial loss of the generative adversarial network (GAN) to train the network, it can generate a reasonable reconstructed image from the input damaged image.

Autoencoders [26], [27] encode an image to a low-dimensional "bottleneck" and decode it by reconstructing the high-dimensional image from the "bottleneck". The purpose of doing this is to obtain the compact feature representation of the scene. Denoising autoencoders [28] reconstruct the image from the corrupted status to learn more robust features. A denoising autoencoder encodes and decodes the damaged image to reconstruct the original image. The repair result image obtained in this way is blurry. The generating model (G) and the discriminant (D) constitute a generative adversarial network [29]. The discriminant model identifies the real data and the generated data, and the goal of the generated model is to generate more realistic data to deceive the discriminant model test. Through the operation, the generated model learns the true distribution of the real data. Deepak Pathak *et al.* pioneered the use of autoencoders and GANs for image restoration and proposed the context-encode method [23], The method combines the semantic expression of the content learned by the encoder reconstruction loss and the data distribution learned by the generative adversarial loss to generate the sharpness repair images, which eliminates the effect of the encoder smoothing blur. As shown in 2(a), a encoder is used as the generator (G) of GAN, the damaged image ($\tilde{x}$) is inputted, and the complete image ($G(\tilde{x})$) is reconstructed through an encoding and decoding pipeline. Li *et al.* [24] used the same idea to do address missing completions. However, because of the GAN structure used in the method, the generated images are randomly with the learned distribution of the true data, and the process of generating the image is too free. The image content generated by this method has obvious traces of artificial modification, and sometimes, the generated content is quite different from the target. Mehdi *et al.* [30] proposed

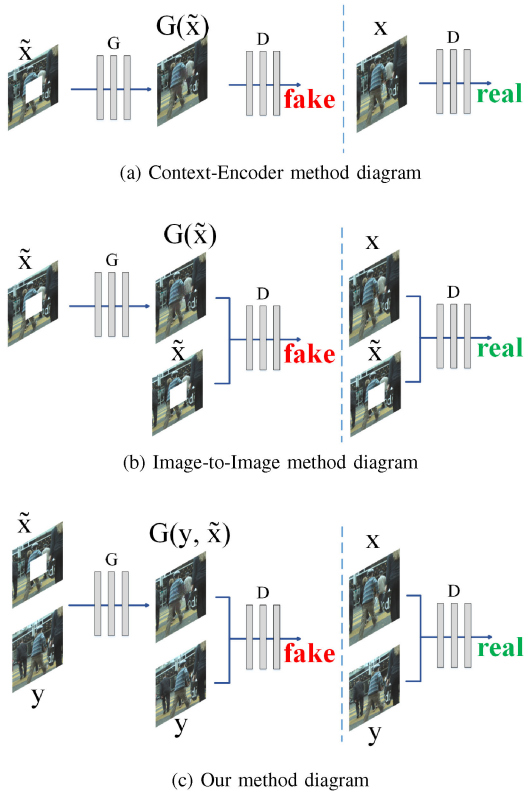


Fig. 2. Compare different image inpainting methods' diagram.

conditional generative adversarial networks (CGAN), which constrain the generation process by adding condition variables to the generator and discriminator so that the generated adversity network can generate pictures that meet the specified conditions. Isola *et al.* exploited the idea of CGAN, improved the context-encoder method, and proposed the image-to-image method [25]. As shown in Fig. 2(b), the damaged image ($\tilde{x}$) is introduced in the method as a condition into the discriminating process so that the discriminator discriminates between a true (damaged image ($\tilde{x}$), the original image (x)) or fake (damaged image ($\tilde{x}$), repairs the resulting image ($G(\tilde{x})$)) image pair, which reaches a certain constraint to the repair result. Because the damaged image is exactly the same as the image content of the repair target in the nondamaged area, image-to-image adds skip layers from the encoder to the decoder in the autoencoder structure (similar to UNet [31]), sharing the hierarchical features of the image in the nonimpaired area so that the repair task is concentrated in the damaged area, reducing the complexity of reconstruction. Another improvement is that Image-to-Image uses the "PatchGAN" classifier in the discriminator, which only penalizes structures at the scale of the image patches. The PatchGAN architecture was first applied in [32] to capture local style statistics. Fig. 1(e) is the repair result of the Image-to-Image method, compared the repair Fig. 1(e) with the repair target Fig. 1(c), We find that the repair result is reasonable and natural, consistent with the image scene semantics. However, when comparing the details, we find that the content is different, and the details of the target image are not restored correctly.

The existing based on deep learning image restoration methods repair the damaged image according to the scene expression and the distribution of the training data witch can generate reasonable scene content in the damaged area. However, the details of the content are not the same as the repair target. In this paper, we solve the existing problems in the image restoration methods with introducing other perspective images in the multicamera system to assist and constrain the image restoration process so that the damaged images in the multiview system can be automatically and accurately restored, and the security of the system is improved. Comparing the left-view damaged graph Fig. 1(a) with the right-view graph Fig. 1(b), it can be seen that there is occlusion between the objects of different viewing angles, and the viewing angles deviate. It is not possible to directly transform the right-view image to the left-view to rematch the missing region by spatial transformation according to the two cameras pose relationship. In this paper, based on the architecture of the generative adversarial network [30], the synchronization frames of other cameras are introduced into the image restoration in the multiview system to assist and constrain the image restoration process so that the repair results are true and clear. The contribution of this article lies in the following three points:

1) The architecture of a multiview scene image restoration method based on conditional generative adversarial networks is proposed. The multiview image is combined to repair the damaged image, which solves the problem of inaccurate and noise-filled repair results (with the existing image restoration methods).
2) Due to the deviation of the field of view and the multiple viewing angles, the method introduces spatial transformation into the image restoration so that the multiview image is aligned on the scene to be repaired, thereby improving the accuracy of the repair result.
3) This paper proposes a multiview scene image fusion model that can efficiently fuse information between different perspectives. The way is grouping convolving images in different views, and then exchanging each group's feature map channels.

The whole method combines the reconstruction loss and the GAN loss and integrates spatial transform, group convolution and channel exchange processing. The damaged image achieves a natural, accurate and clear repair effect. Fig. 1(f) shows that the repair result with our method is very similar to the target image shown in Fig. 1(c). We will conduct detailed comparative experimental analysis in the experimental section.

II. Proposed Algorithm

This paper proposes an accurate image inpainting method for digital images that are damaged during acquiring, processing, compressing, transmitting and decompressing. The method references a priori information provided by other cameras to assist and constrain the inpainting process. Fig. 3 is the global architecture of our method, which is a conditional generative adversarial network that consists of a generator and a discriminator. The network is used to learn the distribution of the training images.

The generator is an autoencoding structure composed of an encoder and a decoder, which are used to learn the content and semantic features of the image scene. We tested the proposed method on a public dataset [33], which was acquired by the ETH Zurich vision group using a mobile platform equipped with two cameras. Our method combines the damaged left camera image with the intact right camera image for inpainting. The generator encodes the damaged image (Input) together with the intact image (Condition) and reconstructs the inpainted left image (Output) during the decoding process. The role of the discriminator is to verify the authenticity of the image pairs (True (Condition, Target) or False (Condition, Output)), which allows the generator to produce more realistic inpainting images. In the encoder, we separately convolve the left and right viewing images and exchange some channels (Channel Shuffle) after convolution to fully utilize the left and right views information. At the beginning of the generator, spatial transformation processing (STN) is performed on the right-view image so that the left and right view images are aligned in the damaged region to make full use of the right perspective with intact information to repair the left perspective with missing information. Since the image is locally damaged, the repaired image has the same image content as the damaged image in the undamaged area, so the encoder information (Add) is added to the decoder to reduce the loss of existing information, especially the image details. In this way, the focus of the network is concentrated on the reconstruction of the damaged areas. Experiments show that all the strategies adopted in our method are effective. Each component theory of the method is presented below.

A. Encoder-Decoder

Our generator is a simple encoder-decoder pipeline. This architecture tries to reconstruct the image after passing it to a low-dimensional bottleneck layer. Then the networks learned the image content and semantics [26]–[28]. The context-encoder [23] first combines the autoencoder with GAN for image inpainting. The autoencoder uses the L2 distance to capture the overall structure of the missing region in relation to the context. Image-to-Image [25] using the L1 distance replaces the L2 distance to reduce the blurring. Our method uses the L1 distance to recover the damaged image. The difference is that the image-to-image method only encodes the damaged image to reconstruct the inpainted image; this prior information is not sufficient to reconstruct a realistic, clear, less noisy image. Our method introduces other perspective information to assist and constrain the damaged image for inpainting. The process reconstructs the inpainting image using the left damaged image ($\tilde{x}$) and the corresponding right intact image information (y). The reconstructed loss can be expressed as the following form:

$$L_{L_1}(G) = E_{x,y}[\|x - G(y, \tilde{x})\|_1] \tag{1}$$

where x is the inpainting target of the left view image, $\tilde{x}$ is the inpainting target of the left view image, y is the intact right view image.

Similar to the image-to-image method, we add skip connection from the encoder to the decoder in each layer. This strategy increases the information flow from the encoder to the

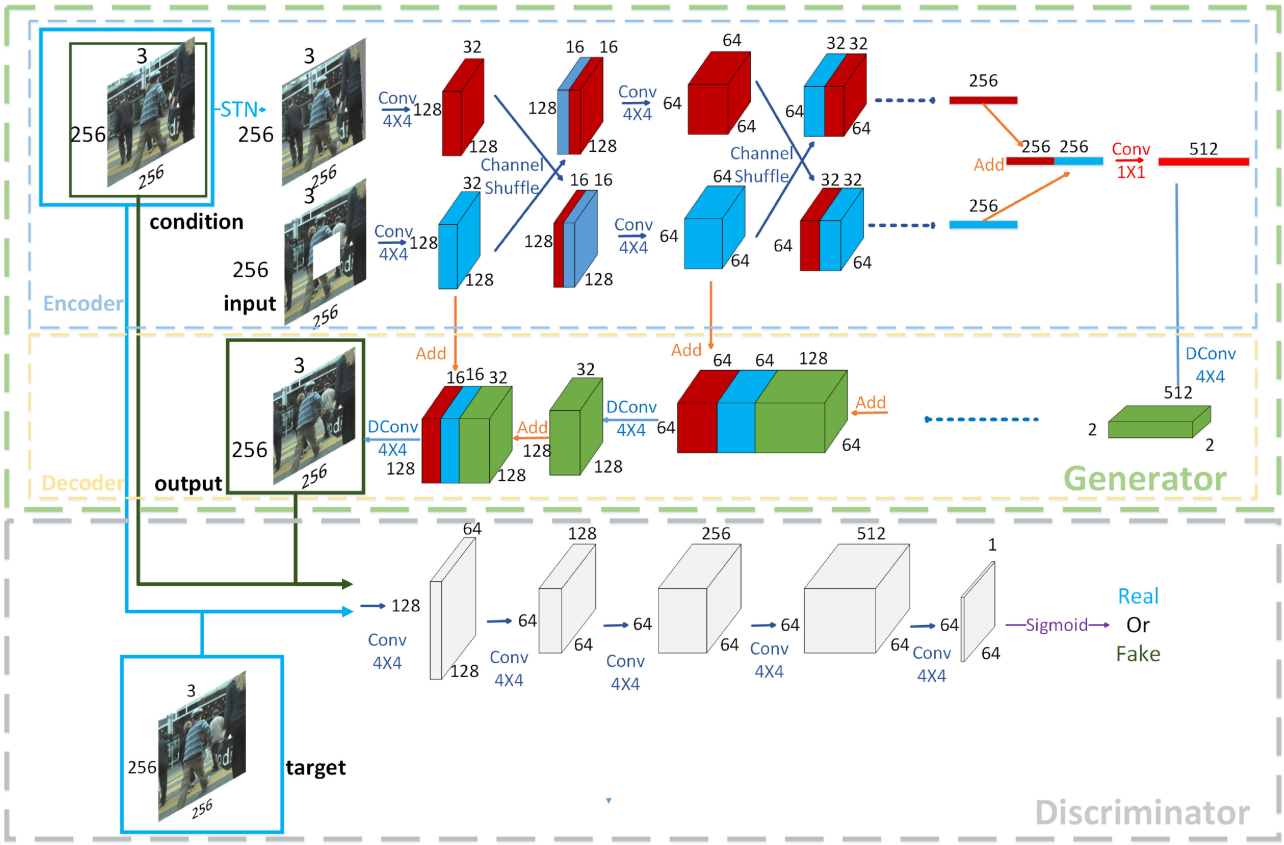


Fig. 3. The network architecture of our method. In the figure, red, blue, green, and white blocks all represent convolution feature maps. After each group convolution in Encoder, the feature maps from different groups (red, green) are exchanged via channel shuffle. In the decoder, the deconvolution feature maps (green) at each stage of the Decoder is then concated with the corresponding features maps of the Encoder (red, green), which is the input of the next phase deconvolution. The white block in the figure represents the convolution feature maps of the discriminator.

decoder and decreases the difficulty of reconstruction so that the generator can focus on the recovery of the abnormal areas. Spatial transform, group convolution, and channel shuffle are also applied to the generator in our method to fuse information from multiple perspectives.

B. Spatial Transform

To reduce the viewing angle deviation between multiple views, spatial transform networks (STN) were introduced into our architecture. A spatial transform network is a learning module proposed by Jaderberg *et al.* to enhance the robustness of neural networks [34], which can be inserted into convolutional architectures, giving neural networks the ability to actively spatially transform feature maps. By introducing spatial transformation processing, the network can select important areas in the image and transform the area into a posture that is conducive to the task. Jaderberg *et al.* used STN to increase the rotation invariance of the network for object detection. However, in our method, the spatial transform processing is only performed on the right-view image, so that the left and right camera images are aligned in the damaged area eliminating the field of view offset between the multiple view images and utilizing other perspective information to inpaint the damaged images more completely. Fig. 4 shows the

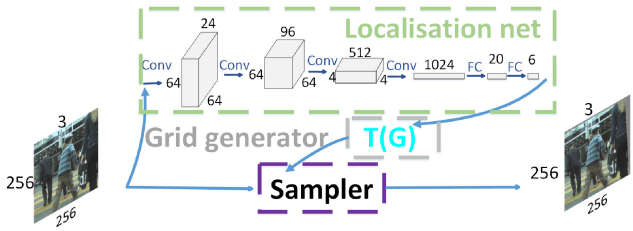


Fig. 4. The spatial transform networks used in our method.

structure of our STN module. The right-view image is inputted into the model and through localization net (four convolutional layers and two fully connected layers), we obtain the six affine transformation parameters (θ). With these parameters, we can perform affine transformation by the following formula:

$$\begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{bmatrix} x^s \\ y^s \\ 1 \end{bmatrix} = \begin{bmatrix} x^t \\ y^t \end{bmatrix} \tag{2}$$

where $\{x^s, y^s\}$ represents the pixel coordinates of the original image, and $\{x^t, y^t\}$ represents the pixel coordinates of the image after the affine transformation. To obtain the mapping of the

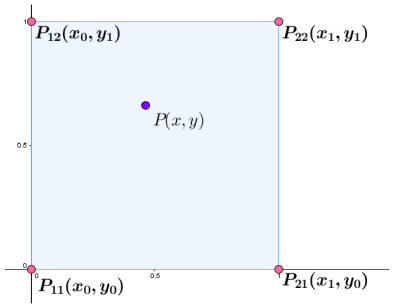


Fig. 5. Bilinear interpolation schematic.

target pixel at the original pixel point, we transform the affine transformation formula:

$$\begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix}^{-1} \begin{bmatrix} x^t \\ y^t \\ 1 \end{bmatrix} = \begin{bmatrix} x^s \\ y^s \end{bmatrix}. \tag{3}$$

Thus, the construction of the transformation target map is transformed into the original image pixel subscript swap problem:

$$PixelMatrix^t = PixelMatrix^s[x^s, y^s]. \tag{4}$$

However, the floating point format $[x^s, y^s]$ does not correspond to the integer value of the pixel coordinates of the original image. We must use the local approximation principle of the image data for interpolation. Bilinear interpolation is a method of combining quality and speed. Fig. 5 shows that a floating point coordinate is located in the middle of four pixel coordinates. Using bilinear interpolation can we get its pixel value:

$$\begin{aligned} Pixel(x, y) = {} & \frac{x_1 - x}{x_1 - x_0} \cdot \frac{y_1 - y}{y_1 - y_0} \cdot Pixel(x_0, y_0) \\ & + \frac{x - x_0}{x_1 - x_0} \cdot \frac{y_1 - y}{y_1 - y_0} \cdot Pixel(x_1, y_0) \\ & + \frac{x_1 - x}{x_1 - x_0} \cdot \frac{y - y_0}{y_1 - y_0} \cdot Pixel(x_0, y_1) \\ & + \frac{x - x_0}{x_1 - x_0} \cdot \frac{y - y_0}{y_1 - y_0} \cdot Pixel(x_1, y_1). \end{aligned} \tag{5}$$

This is the T(G) performed in the spatial transformation model that generates the coordinate map from the original map to the transformed map. The sampling work is performed by the "sampler". Fig. 6 is a presentation of the role of the spatial transformation network in our method, where Fig. 6(a1) and Fig. 6(a2) are the damaged left perspective images, Fig. 6(b1) and Fig. 6(b2) are the intact right perspective images, Fig. 6(c1) and Fig. 6(c2) are the spatial transformation results, and the left column is the result of processing in the middle area damaged images. The right column is the result of the processing in the random area damaged images. As seen in Fig. 6(c1) and Fig. 6(c2), the spatial transformation network enables the right view image to align the scene content on the missing area of the left view image, and the right view image can better provide the information required by the left view damaged image.

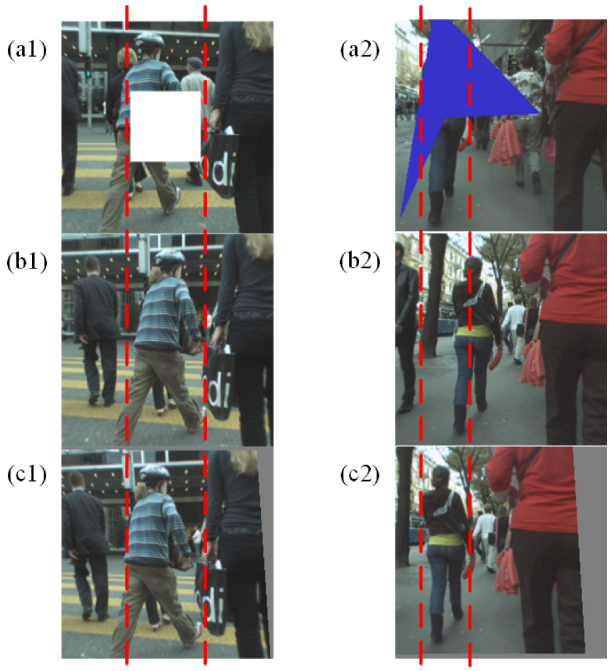


Fig. 6. The effect of using spatial transformation network.

C. Group Convolution and Channel Shuffle

To fully exchange and fuse the information between different perspective images, our method convolves separately (Group Convolution) on different perspective images and then splits and exchanges the feature maps according to the number of images and the feature map channels. Group convolution is used as an optimization design method in MobileNets [35], Xception [36] and ResNeXt [37]. The main idea is first to carry out group convolution on the feature maps and then use point-by-point convolution to fuse information of the grouped channels; in this way, the number of parameters and calculations of the network can be reduced. Because point-by-point convolution also requires many parameters and calculations, ShuffleNet [38] proposed the channel rearrangement idea to group the point-by-point convolution, further reducing the number of network parameters. The above methods all use group processing to reduce the number of network parameters and calculations in processing a single image feature. Different from the above methods, in our method, the group convolution operation is performed on the left and right view images in the encoder, then half of the feature map channels are exchanged with each other. The purpose is to fully integrate the information between the multiple views. At the same time, the number of parameters and calculations is reduced. Unlike shuffleNet [38], the information fusion is performed by point-by-point convolution after each group convolution; we only use a point-by-point convolution after the bottleneck layer to fuse the information of the two group feature maps. Through group convolution and channel shuffle

processing, information between the different perspectives can be efficiently merged.

D. Conditional Generative Adversarial Networks

The context-encoder method solves the problem of inpainting images with large damaged areas, combining an autoencoder with GAN to perform image inpainting. As shown in Fig. 2(a), it encodes and decodes damaged image ($\tilde{x}$) to reconstruct the inpainting result image (G($\tilde{x}$)). The discriminator makes a true or false discrimination between the repair target (x) and the repaired result (G($\tilde{x}$)). However, only utilizing the data content, semantic features learned by the autoencoder and the data distribution learned by GAN to inpaint the image, the inpainting process lacks the assistance and constraints of other prior information, the generated inpainting image has the same distribution as the target image, and the content is not completely consistent. Image-to-Image solves this problem by introducing a conditional discriminant model. As shown in Fig. 2(b), it introduces the damaged image ($\tilde{x}$) as a condition to the discriminator. The discriminator classifies a pair of images that is composed of the damaged image ($\tilde{x}$) integrated separately with the target inpainting image (x) and the inpainting result image ($G(\tilde{x})$). However, due to the limited a priori information provided by the damaged image, the image-to-image method does not substantially improve the accuracy of the inpainting results. Our method assists and constrains the inpainting process of the damaged images ($\tilde{x}$) by integrating images (y) from other perspectives. Fig. 2(c) is a schematic representation of our method. We combine the damaged left-view image ($\tilde{x}$) with the corresponding right-view image (y) to reconstruct the intact left-view image ($G(y, \tilde{x})$). The discriminator discriminates between true {right view image (y), repair target image (x)} or false {right view image (y), repair result image (G($y, \tilde{x}$))}.

In our method GAN loss can be expressed as

$$L_{CGAN}(G, D) = E_{y,x}[\log D(y, x)] + E_{y,\tilde{x}}[\log(1 - D(y, G(y, \tilde{x})))] \quad (6)$$

where G tries to minimize this objective against an adversarial D that tries to maximize it, i.e.:

$$G_* = \arg \min_G \max_D L_{CGAN}(G, D). \quad (7)$$

E. Joint Loss Function

Previous work [23], [25], [39] proved that only the reconstruction loss can obtain a smooth-blurred image, and combining the reconstruction loss with the adversarial loss can improve the sharpness of the generated image. Our method utilizes multiple perspective images and combines the reconstruction loss with adversarial loss to inpaint images to obtain high-quality results.

The final objective of our method is

$$G_* = \arg \min_G \max_D \lambda_{L_{CGAN}} L_{CGAN}(G, D) + \lambda_{L_1} L_{L_1}(G) \quad (8)$$

where $\lambda_{L_{CGAN}}$ and λ_{L_1} are the weight of adversarial loss and the reconstructed loss, respectively.

III. Experiments

A. Database Description

To evaluate the image inpainting effects, we conducted experiments on a public dataset collected by a pair of AVT Marlins F033 C mounted on a car, with a resolution of 640 × 480 (bayered), and a framerate of 13–14 FPS [33]. The dataset consists of 8 consecutive time series, a total of 5,263 pairs of synchronized time frame images. The experiment randomly takes 4,208 pairs to train, and 1,055 images to perform the test. The left and right images are all resized to 256 × 256 before being input to the network. We set a blank area in the middle of the left camera images with 76 pixels (RGB pixel values are set to 255). Using these left camera images and the right intact camera images, we train and test the proposed algorithm. We also set random areas and shapes and fill random colors in the left-view images to verify the versatility and robustness of the algorithm.

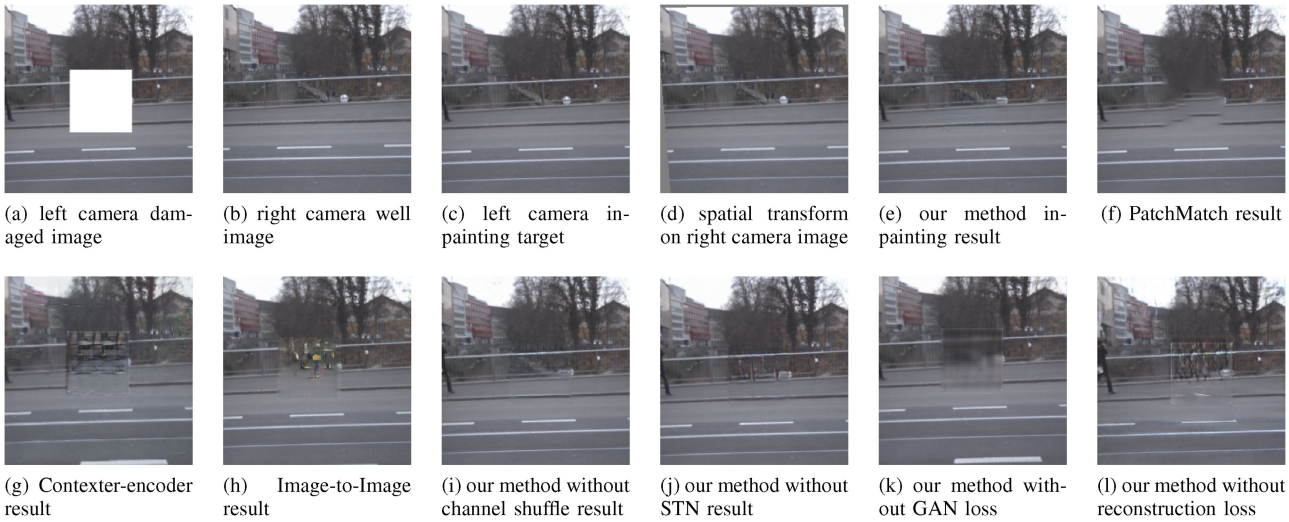
B. Experimental Setup

In the encoder, each convolution adopts a "Convolution→Batch Normalization→Relu" structure. The convolution kernel size is 4 × 4, with 1 padding and 2 stride. After one convolution, the height and width of the feature maps are cut in half, and the channels turn double. However, when the number of channels reaches 512, they are no longer increased. At the bottleneck layer, the feature map dimensionality is 1 × 1 × 256, the are no longer increased. We add a pointwise convolution after the bottleneck layer. In the decoder, we adopt the "Deconvolution→Batch Normalization→Leak ReLU" structure as the reverse of the convolution to reconstruct the left original camera images. The discriminator and localization network structures are shown in Fig. 3 and Fig. 4, respectively.

We build our models in tensorflow. We set the reconstructed loss weight λ_{L_1} to 100 and adversarial loss $\lambda_{L_{CGAN}}$ to 1 using the backpropagation algorithm with Adam [40] optimization to tune the parameters.

C. Experimental Results

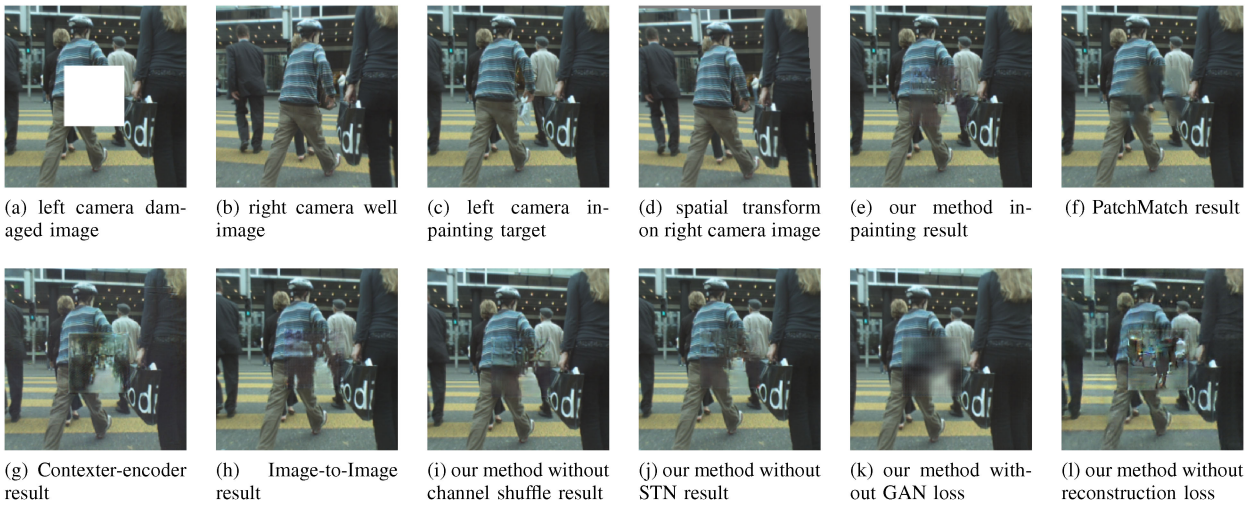
To evaluate our proposed approach visually and quantitatively, we take some previous methods including PatchMatch [21], Context-Encoder [23] and Image-to-Image [25] as the comparable objects. To verify the effectiveness in image inpainting, we carry out ablation experiments and analyze the influences of each factor on the inpainting results. Firstly, we qualitatively analyze the inpainting effects of the different methods. Fig. 7, Fig. 8 and Fig. 9 show some experimental results. As shown in Fig. 7(f), Fig. 8(f) and Fig. 9(f), PatchMatch fills the damaged area according to the surrounding pixels. When the target object to be inpainted is entirely in the damaged area and is different from the surrounding pixel features, the object information is not recovered. For example, the elongated pipe in Fig. 7(f), the hand in Fig. 8(f) and the plastic bag in Fig. 9(f) have not been inpainted by the PatchMatch method. The resulting maps inpainted by the Context-Encoder and Image-to-Image methods have obvious artifacts and look unnatural. Because these



(a) left camera damaged image (b) right camera well image (c) left camera inpainting target (d) spatial transform on right camera image (e) our method inpainting result (f) PatchMatch result

(g) Contexter-encoder result (h) Image-to-Image result (i) our method without channel shuffle result (j) our method without STN result (k) our method without GAN loss (l) our method without reconstruction loss

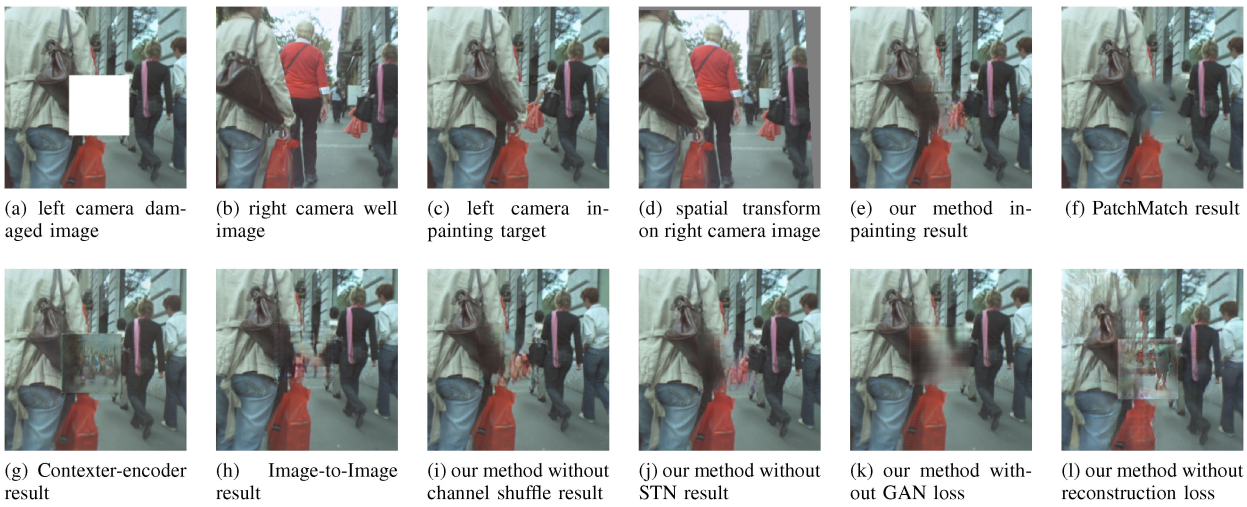
Fig. 7. Compare different image inpainting methods.



(a) left camera damaged image (b) right camera well image (c) left camera inpainting target (d) spatial transform on right camera image (e) our method inpainting result (f) PatchMatch result

(g) Contexter-encoder result (h) Image-to-Image result (i) our method without channel shuffle result (j) our method without STN result (k) our method without GAN loss (l) our method without reconstruction loss

Fig. 8. Compare different image inpainting methods.



(a) left camera damaged image (b) right camera well image (c) left camera inpainting target (d) spatial transform on right camera image (e) our method inpainting result (f) PatchMatch result

(g) Contexter-encoder result (h) Image-to-Image result (i) our method without channel shuffle result (j) our method without STN result (k) our method without GAN loss (l) our method without reconstruction loss

Fig. 9. Compare different image inpainting methods.

two inpainting methods only rely on the semantic expression learned from the autoencoder and the data distribution learned by the generative adversarial networks, the inpainting result is generated "from the air". The Image-to-Image method provides better results than the context-encoder method. Because the context-encoder has no skip connected, the generator lacks some detailed features. It is difficult to reconstruct the entire image and details only using the deep semantic features. However, by introducing informations from the other perspectives, adding more guidances and constraints to the image generation process, our method produce the more accurate and natural inpainted results (Fig. 7(e), Fig. 8(e), Fig. 9(e)). When the reconstruction loss is removed (set the reconstructed loss weight λ_{L_1} to 0), the generation process is more random and chaotic (Fig. 7(l), Fig. 8(l), Fig. 9l). Instead, if the adversarial loss is removed(set the adversarial loss weight λ_{LCGAN} to 0), the inpainting result shows a smoothing effect (Fig. 7(k), Fig. 8(k), Fig. 9(l)). In the experiment of removing the group convolution, we stacked the left and right viewing image channels together for convolution, the number of channels in one convolution layer is equal to the sum of the number of channels in which the left and right images are separately convolved. Observed from the images of the inpainted results after removing the group convolution (Fig. 7(i), Fig. 8(i), Fig. 9(i)), we find that the inpainted results are slightly worse. For example, compared with Fig. 7(e), the stone inpainted under the guardrail is dim in Fig. 7(i). Observed from the images of the spatial transformation results (Fig. 7(d), Fig. 8(d) and Fig. 9(d)), the right perspective has some left offset after spatial transformation, and each image is transformed slightly differently. The transform coefficients are determined by the spatial transformation network module according to the area that must be inpainted by the left and right images. So that the left and right perspectives can be overlapped in the damaged area, the information of the right perspective can assist inpainting the image of the left perspective.

To quantificationally analyze the proposed method, we compared our method with previous methods in term of L1 Loss, L2 Loss, and Peak Signal-to-Noise Ratio (PSNR).

Where, the average L1 Loss is the absolute value of the average pixel value difference between the inpainted image and the target image:

$$L1\,Loss = E_{\tilde{x},x}[\|\tilde{x} - x\|_1]. \tag{9}$$

L2 Loss is the square of the difference between the average pixel value of the inpainted image and the target image:

$$L2\,Loss = E_{\tilde{x},x}[\|\tilde{x} - x\|^2]. \tag{10}$$

Peak Signal to Noise Ratio (PSNR) is a standard used to measure the image distortion or noise levels. It is the logarithm of the mean square error between the original image and the image being processed relative to the square of the signal maximum $(2^n - 1)^2$ (n is the number of bits per sample), and its unit is dB. It is defined as follows:

$$PSNR = 10 \cdot \log_{10}\left\{\frac{(2^n - 1)^2}{MSE}\right\}. \tag{11}$$

TABLE I
QUANTITATIVE RESULTS FOR DIFFERENT IMAGE INPAINTING METHODS

Method	Mean L1 Loss	Mean L2 Loss	PSNR
PatchMatch [21]	1.389%	5.305%	26.218 db
Contex-Encoder [23]	3.124%	5.644%	25.221 db
Image-to-Image [25]	1.451%	4.608%	26.980 db
Ours (remove group convolution and channel shuffle)	1.008%	3.171%	30.421 db
Ours (remove STN)	1.000%	3.052%	30.725 db
Ours (remove GAN loss)	1.301%	3.457%	29.521 db
Ours (remove L1 loss)	4.009%	7.283%	23.214 db
Ours	**0.985%**	**3.018%**	**30.745 db**

The larger the PSNR value between the two images, the more natural the inpainted image is. Among them, MSE is the mean square error between the original image and the inpainted image; n is the number of bits per sample value, $(2^n - 1)^2$ indicating the maximum value of the image color, and the maximum value of the 8-bit graph point is 255.

Table I reports the quantitative results, and our method achieves the highest numerical performance (minimum mean L1 Loss, minimum mean L2 Loss, highest PSNR). After adding the skip-connected, conditional discriminant, and changing the reconstructed loss from L2 to L1 in the Context-Encoder method, the Image-to-Image method Mean L1 Loss decreases 1.673%, Mean L2 Loss decreases 1.036%, and PSNR increases by 3.765 dB. Compared with the Image-to-Image method, we integrate the other view informations into the generator and discriminator, mean L1 Loss decreases 0.466%, mean L2 Loss decreases 1.590% and PSNR increases by 3.765 dB. If we remove the adversarial loss in the loss function (Eq. (8)), only the reconstruction loss is included, mean L1 Loss increases by 0.316%, mean L2 Loss increases by 0.439%, PSNR decreases 1.224 dB; If we remove the reconstruction loss in the loss function (Eq. (8)), only the adversarial loss is used; mean L1 Loss increases by 3.024%, mean L2 Loss increases by 4.265%, and PSNR decreases 7.531 dB. This shows that the reconstruction loss of the auto-encoder plays a decisive role in the reconstruction of the image missing area. After adding the adversarial loss, the reconstructed image eliminates the smoothing blur effect and makes the reconstructed image sharper. If we do not use group convolution in our method, the mean L1 Loss increases by 0.023%, mean L2 Loss increases by 0.153%, and PSNR decreases 0.324 dB. In addition, adopting group convolution, the numbers of calculations and parameters are reduced. If we do not perform spatial transformation on other perspectives, the mean L1 Loss increases by 0.015%, mean L2 Loss increases by 0.034%, and PSNR decreases 0.020 db. Experiments demonstrate that spatial transformation processing aligns missing content between multiple views, this has a certain effect on the improvement of the inpainting effect.

In addition, we also performed inpainting experiments on damaged areas with random areas, random shapes and random colors, shown in Fig. 10. The results show the proposed algorithm is versatile and robust to general image inpainting.

Fig. 10. Experimental demonstration of damage area restoration withrandom region, random shape and random color. (a) left camera damaged image (b) right camera well image (c) left camera inpainting target (d) spatial transform on right camera image (e) our method inpainting result.

The proposed algorithm is tested on the GTX1080 graphics card. When the resolution of the images is set to 256×256, the inpainting frame rate is 22 FPS, which can basically achieve the real-time inpainting in the multiview system.

IV. Conclusion

In this paper, we proposed an image anomaly inpainting method for multiview scenes. We used the conditional generative adversarial networks as the infrastructure and took the images of other perspectives as conditions. We also performed spatial transformation processing on the other perspective images to eliminate the field of view deviation. We also established a model for efficient fusion of multiview informations. Group convolution is performed for different perspective images and channel shuffle is then conducted on these convolution feature maps. With this method, different perspectives of informations can be fully fused using a small number of parameters and calculations. Finally, the ablation experiments analyze the inpainting effectiveness of the proposed method qualitatively and quantitatively and verify the reliability of the strategy adopted in the method.

The proposed method has the weakness that it can only be applied to a multiview system where there is a coincidence between different viewing angles. In addition, the image inpainting speed is a bottleneck that can constrain the application for self-driving cars based on low-cost platforms. Our aim is to address these issues in future work.

Acknowledgment

The authors would like to thank Prof. Y. Sun (University of Toronto) for guidance during revision.

References

[1] Y. Li, Z. Hu, Z. Li, M. A. Sotelo, and Y. Ma, "Multiscale site matching for vision-only self-localization of intelligent vehicles," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 3, pp. 170–183, Fall 2018.

[2] C. Guindel, D. Martin, and J. M. Armingol, "Traffic scene awareness for intelligent vehicles using convnets and stereo vision," *Robot. Auton. Syst.*, vol. 112, pp. 109–122, 2019.

[3] C. Patruno, M. Nitti, A. Petitti, E. Stella, and T. Dorazio, "A vision-based approach for unmanned aerial vehicle landing," *J. Intell. Robot. Syst.*, vol. 95, no. 2, pp. 645–664, 2019.

[4] R. Ma, T. Maugey, and P. Frossard, "Optimized data representation for interactive multiview navigation," *IEEE Trans. Multimedia*, vol. 20, no. 7, pp. 1595–1609, Jul. 2018.

[5] B.-S. Kim, K.-A. Choi, W.-J. Park, S.-W. Kim, and S.-J. Ko, "Content-preserving video stitching method for multi-camera systems," *IEEE Trans. Consum. Electron.*, vol. 63, no. 2, pp. 109–116, May 2017.

[6] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Vision-based occlusion handling and vehicle classification for traffic surveillance systems," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 2, pp. 80–92, Summer 2018.

[7] H. Liu, S. Lee and J. S. Chahl, "A multispectral 3D vision system for invertebrate detection on crops," *IEEE Sensors J.*, vol. 17, no. 22, pp. 7502–7515, Nov. 2017.

[8] C. Wang, Y. Fang, H. Zhao, C. Guo, S. Mita, and H. Zha, "Probabilistic inference for occluded and multiview on-road vehicle detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 215–229, Jan. 2016.
[9] L. Liu, H. Li, Y. Dai, and Q. Pan, "Robust and efficient relative pose with a multi-camera system for autonomous driving in highly dynamic environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2432–2444, Aug. 2018.
[10] X. Ji, G. Zhang, X. Chen, and Q. Guo, "Multi-perspective tracking for intelligent vehicle," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 518–529, Feb. 2018.
[11] B. Bozorgtabar and R. Goecke, "Msmct: Multi-state multi-camera tracker," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 12, pp. 3361–3376, Dec. 2018.
[12] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. 7th IEEE Int. Conf. Comput. Vision*, 1999, vol. 2, pp. 1033–1038.
[13] Z. Lu, H. Huang, L. Li, and D. Cheng, "A novel exemplar-based image completion scheme with adaptive TV-constraint," in *Proc. 4th Int. Conf. Genetic Evol. Comput.*, Dec. 2010, pp. 94–97.
[14] D. Ding, S. Ram, and J. J. Rodríguez, "Image inpainting using nonlocal texture matching and nonlinear filtering," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1705–1719, Apr. 2019.
[15] S. D. Rane, J. Remus, and G. Sapiro, "Wavelet-domain reconstruction of lost blocks in wireless image transmission and packet-switched networks," in *Proc. Int. Conf. Image Process.*, 2002, vol. 1, pp. I–309–I–312.
[16] S. D. Rane, G. Sapiro, and M. Bertalmio, "Structure and texture filling-in of missing image blocks in wireless transmission and compression applications," *IEEE Trans. Image Process.*, vol. 12, no. 3, pp. 296–303, Mar. 2003.
[17] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. 27th Annu. Conf. Comput. Graph. Interact. Techn.*, 2000, pp. 417–424.
[18] T. K. Shih and R.-C. Chang, "Digital inpainting - survey and multilayer image inpainting algorithms," in *Proc. 3rd Int. Conf. Inf. Technol. Appl.*, Jul. 2005, vol. 1, pp. 15–24.
[19] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 303–312, 2003.
[20] M. Wilczkowiak, G. J. Brostow, B. Tordoff, and R. Cipolla, "Hole filling through photomontage," in *Proc. Brit. Mach. Vis. Conf.*, Oxford, U.K., 2005, pp. 492–501.
[21] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, 2009, Art. no. 24.
[22] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Trans. Graph.*, vol. 26, no. 3, 2007, Art. no. 4.
[23] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 2536–2544.
[24] Y. Li, S. Liu, J. Yang, and M.-H. Yang, "Generative face completion," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, vol. 1, pp. 5892–5900.
[25] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1125–1134.
[26] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[27] Y. Bengio, "Learning deep architectures for AI," *Foundations Trends Mach. Learning*," vol. 2, no. 1, pp. 1–127, 2019.
[28] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.
[29] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Advances Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[30] M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv:1411.1784, 2014.
[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assisted Intervent.*, 2015, pp. 234–241.
[32] C. Li and M. Wand, "Precomputed real-time texture synthesis with Markovian generative adversarial networks," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 702–716.
[33] A. Ess, B. Leibe, K. Schindler, and L. V. Gool, "A mobile vision system for robust multi-person tracking," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2008, pp. 1–8.
[34] M. Jaderberg *et al.*, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
[35] A. G. Howard *et al.*, "Mobilenets efficient convolutional neural networks for mobile vision applications," 2017, *arXiv1704.04861*.
[36] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jul. 2017, pp. 1800–1807.
[37] S. Xie, R. Girshick, P. Dollr, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jul. 2017, pp. 5987–5995.
[38] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet an extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2018, pp. 6848–6856.
[39] A. B. L. Larsen, S. K. Sonderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 2341–2349.
[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–41.

Zefeng Yuan received the B.S. degree in mechanical and electronic engineering from Taiyuan Science and Technology University, Taiyuan, China, in 2015, and the M.S. degrees in mechanical and electronic engineering from Shanghai University, Shanghai, China, in 2018. He is currently an Automatic Driving Algorithm Engineer with Hangzhou Leapmotor Company.

Hengyu Li received the B.S. degree in mechanical engineering and automation from Henan Polytechnic University, Jiaozuo, China, in 2006, and the M.S. and Ph.D. degrees in mechanical and electronic engineering from Shanghai University, Shanghai, China, in 2009 and 2012, respectively. He is currently an Associate Professor with the School of Mechatronic Engineering and Automation, Shanghai University. His research interests include mechatronics and robot bionic vision system and autonomous cooperative control for multiple robots.

Jingyi Liu received the B.S. degree in mechanical engineering and automation and the M.S. degree in mechanical and electronic engineering in 2014 and 2017, respectively, from Shanghai University, Shanghai, China, where he is currently working toward the Ph.D. degree in mechanical engineering. His research interests include computer vision, machine learning, and vision for mobile robotics.

Jun Luo received the B.S. and M.S. degrees in mechanical engineering and automation from Henan Polytechnic University, Jiaozuo, China, in 1994 and 1997, respectively, and the Ph.D. degrees in mechanical and electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2000. He is currently a Professor with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China. His research interests include unmanned craft, mechatronics, artificial intelligence, and autonomous cooperative control for multiple robots.