A video inpainting method for unmanned vehicle based on fusion of time series optical flow information and spatial information

Rui Zhao¹, Hengyu Li¹*, Jingyi Liu¹, Huayan Pu¹, Shaorong Xie¹ and Jun Luo¹,²

Abstract

In this article, the problem of video inpainting combines multiview spatial information and interframe information between video sequences. A vision system is an important way for autonomous vehicles to obtain information about the external environment. Loss or distortion of visual images caused by camera damage or pollution seriously makes an impact on the vision system ability to correctly perceive and understand the external environment. In this article, we solve the problem of image restoration by combining the optical flow information between frames in the video with the spatial information from multiple perspectives. To solve the problems of noise in the single-frame images of video frames, we propose a complete two-stage video repair method. We combine the spatial information of images from different perspectives and the optical flow information of the video sequence to assist and constrain the repair of damaged images in the video. This method combines the interframe information of the front and rear image frames with the multiview image information in the video and performs video repair based on optical flow and a conditional generation adversarial network. This method regards video inpainting as a pixel propagation problem, uses the interframe information in the video for video inpainting, and introduces multiview information to assist the repair based on a conditional generative adversarial network. This method was trained and tested in Zurich using a data set recorded by a pair of cameras mounted on a mobile platform.

Keywords

Optical flow, deep learning, generative adversarial networks, video inpainting

Date received: 7 June 2021; accepted: 27 September 2021

Topic: Vision Systems
Topic Editor: Antonio Fernandez-Caballero
Associate Editor: Shenyong Chen

Introduction

With the development of deep learning and computer vision technology, multiview vision system has become significant in the field of automation.¹,² Due to the abundant image information provided by the multiview system, the multiview image acquisition system is widely used in navigation,³ panorama,⁴ the occlusion process and vehicle classification,⁵ target detection,⁶–⁸ and tracking⁹,¹⁰,²⁶ As shown in Figure 1, multiview vision target recognition and

¹ School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China
² College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing, China

Corresponding author:
Hengyu Li, School of Mechatronic Engineering and Automation, Shanghai University, 99 Shangda Road, Shanghai 200444, China.
Email: lhengyu@shu.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
positioning is used for assisted driving systems, and multi-view vision systems are also used for obtaining surround views around the car body. At present, the basis of autonomous driving is that the vehicle autonomously completes environmental perception, positioning, and mapping. In order to ensure the robustness and reliability of the system, most tasks require high sensor redundancy. Therefore, automatic driving systems generally use a variety of onboard sensors. Hardware modules can be roughly divided into five categories: exteroceptive sensors, proprioceptive sensors that monitor the state of the vehicle itself, communication units, actuators, and computing units. For positioning and mapping, the currently widely used methods are GPS-IMU fusion, SLAM, positioning based on prior maps, and so on. Perceiving the surrounding environment and extracting information for safe navigation is one of the cores of autonomous driving. And with the development of computer vision research in recent years, cameras including three-dimensional vision have gradually become the most commonly used sensors in perception. The main algorithms are image-based target detection, semantic segmentation, three-dimensional target detection, road and lane line detection, target tracking, and so on. In a word, in terms of environment perception and positioning and mapping, it is mainly through the fusion of information collected by various sensors such as cameras, lasers, and radars to achieve autonomous navigation and control of vehicles. Therefore, the interference or loss of video image signals throughout the method of signal acquisition, compression, translation, and decompression can have an effect on the system’s ability of perceiving the encompassing atmosphere. The aim of video inpainting is to fill the missing areas with content that is consistent in space and time for a given video sequence. This technology has been widely used in the field of autonomous driving.

Due to the motion of video image capture equipment and the complex movement of objects in the video, inpainting real-world HD video sequences is still challenging. Most existing video inpainting algorithms can be classified into two categories. The first follows the normal image inpainting pipeline that describes drawback as a patch-based improvement problem, which inpaints missing regions by sampling spacial or spatial-temporal patches of the well-known regions. Although some good results have been achieved, these methods have two disadvantages. First, these methods fill the missing regions by forward a sleek and homogeneous motion field within the missing region. Therefore, they cannot complete the repair of videos with complex motions. Second, these methods are mainly based on the idea of mathematical optimization. The optimization-based method has high computational time complexity and cannot meet the requirements of real-time processing. Recently, the method of video image restoration supported by deep learning uses associate degree autoencoder to find out the content and linguistics expression of the scene, combined with the generative adversarial network (GAN) to perform video image restoration. The network trains the reconstruction loss of the autoencoder and the adversarial loss of the GAN and reconstructs a good image from the damaged image.

Autoencoder is an unsupervised neural network model. It can learn the hidden features of the input data, which is called encoding. At the same time, the learned new features can be used to reconstruct the original input data, which is called decoding. Autoencoder performs feature extraction on the image to obtain a feature vector and reconstructs the high-dimensional image from the low-dimensional vector, which tries to obtain high-level semantic features of the scene. Pathak et al. pioneeredly proposed an approach that combines the reconstruction loss of the autoencoder and generate the data learned by the adversarial loss to solve the problem of video image restoration. As shown in Figure 2(a), an encoder is employed because of the generator (G) of the GAN. The broken image (ⅹ) is that the
input, and the complete image is reconstructed through an associate in nursing cryptography and decipherment pipeline. Li et al.\textsuperscript{17} used an equivalent plan to finish the missing region of a picture. However, due to the GAN structure utilized in the method, the generated pictures are random with the learned distribution from the real data. So it’s tough to regulate the method of generating the photographs that are different from the target.

Mirza and Osindero\textsuperscript{18} projected conditional generative adversarial networks (CGANs) that constrain the generation method by adding conditional variables to the generator and discriminator. Therefore, the generator can generate pictures that meet the special conditions. Isola et al.\textsuperscript{14} used the thought of CGAN, improved the context-encoder technique, and planned the image-to-image technique. As shown in Figure 2(b), the broken image (\(\hat{x}\)) is introduced as a condition into the discrimination method so that the discriminator will distinguish between a real (damaged image (\(\hat{x}\)), original image (\(x\))) or pretend (damaged image (\(\hat{x}\)), the generating image (\(G(\hat{x})\))) image combine, which achieves a selected affected repair result for the broken image. Since the broken image is the same as the image content of the inpainting target within the nondamaged space. Image-to-image adds skip layers from the encoder to decoder within the autoencoder structure just like UNet,\textsuperscript{19} sharing the hierarchic feature of the image within the nondamaged space so the inpainting task is targeted within the broken space, reducing the complexity of the reconstruction method. Another improvement is that the image-to-image technique uses the PatchGAN classifier that penalizes structures at the size of the image patches. The PatchGAN design was initial utilized in Li and Wand\textsuperscript{20} to capture native vogue statistics. Figure 3(e) is the inpainting result of the image-to-image method. By comparing the inpainting result (Figure 3(e)) with the inpainting target (Figure 3(c)), we find that the inpainting result is reasonable and natural, which is consistent with the image scene semantics. However, we find that the inpainting result is affordable and natural, which is consistent with the image scene linguistics. However, we discover that the detail of the inpainting result is not inpainted properly once examination the main points.

Although convolutional neural networks (CNNs) and GAN have significant effects in image restoration,\textsuperscript{21} video restoration through deep learning methods is still a challenge. Even if the above methods are applied to the video inpainting process, it is difficult to merge the interframe information of the video sequence with the spatial information of the single-frame image. Using deep learning-based image inpainting methods to solve a series of problems in the video field is still challenging. On the one hand, an immediate application of a picture inpainting algorithmic rule to every frame of a video sequence separately cannot make full use of the information on time series. On the other hand, due to the large number of single-frame images of a video sequence, it is difficult to ensure consistency in timing when the entire video sequence is input into the CNN at one time. Furthermore, directly repairing the entire video sequence requires a large amount of graphics storage to store the subsequent processing operations of video image frames and video image processing result, which is computationally impractical.

Xu et al.\textsuperscript{22} proposed a video inpainting method guided by optical flow. In this method, the interframe information in the video sequence is expressed in the form of optical flow, and the damaged optical flow is repaired instead of restoring original single-frame images in the video sequence. Compared with the original image method, using this method to complete the restoration of missing optical flow is easier than directly filling the actual pixels of an image. Making use of optical flow to propagate the pixels can maintain natural consistency in timing, which makes better use of the video sequence. Since the background and most object motions in the video are traceable, the pixel motion of the image is easy to obtain. This method can reduce the difficulty of video restoration by synthesizing a cross-frame optical field. Most of the pixels in the missing area can be made up by spreading from the visible area. This method can make full use of the interframe information of the images in the video sequence, and it is far easier to repair the single-frame image in the video sequence through the refinement of the optical flow field. However, for binocular systems, it is difficult to fuse multiview

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure2}
\caption{Schematic diagram of different repair methods. (a) Context-encoder method diagram, (b) image-to-image method diagram, and (c) conditional generation adversarial networks method diagram.}
\end{figure}
information. In addition, it is difficult to repair objects that are completely damaged in the video sequence.

We propose a video inpainting method that combines optical flow guidance and multiview scene. This method combines the consistency of the video sequence in the timing and the image information in the space of the multiview system to perform video inpainting. The solution in this article is divided into two stages. The purpose of the first stage is to fuse the image information of the perspective that need to be inpainted and auxiliary perspective in the current multiview system and obtaining the preliminary restoration result of the image frames in the video sequence, which reduces the additional cost of the subsequent optical flow guidance inpainting process and makes full use of the multiview image information. The second stage utilizes the initial inpainting results from the first part to extract the optical flow of the video sequence and completes the refinement and inpainting process of the optical flow field from coarse to fine through the deep convolutional network. Finally, we can take advantage of the optical flow to propagate the pixels of visible areas. The innovations of the method in this article are as follows:

1. A video inpainting method based on optical flow guidance and a multiview scene is proposed, which combines the timing information of video sequences and the spatial information of a multiview scene for video inpainting and explores new methods in the field of video inpainting.
2. Due to the angle deviation of the multiview system, the spatial transformation network (STN) is introduced to complete the spatial alignment of the multiview images and better integrate the spatial information of the multiview image.
3. A deep optical flow repair network is introduced, which can maintain time consistency for processing missing areas of arbitrary shapes and complex movements.

The whole method combines the temporal information of the video sequence and the spatial information of the multiview scene. Damaged image frames in a single-view video sequence can get accurate and clear results by this method that fuses spatial transformation, group convolution and features channel information exchange.

**Proposed algorithm**

This article proposes an efficient video inpainting method to deal with video signal abnormalities and damage resulting from the process of signal acquisition, compression, transmission, and decompression. This method utilizes the
prior information provided by the multiview scene combined with the interframe information of the images in the video sequence to complete the restoration of the video sequence. Figure 4 describes the overall architecture of our video inpainting method. The approach consists of three steps. The first step aims to complete the inpainting of the damaged image frame from another view with the CGAN which introduces an STN, thus obtaining the preliminary repair result. The second step aims to complete a coarse-to-fine flow inpainting which is extracted from the preliminary repair result. Finally, the third step is to complete the pixel propagation guided by the optical flow to achieve the final repair result.

The CGAN of the primary step may be a structure composed of a generator and a discriminator, which is employed to be told the linguistics info and have distribution of image frames during a single-view video sequence. The data set we used consists of eight consecutive time series from two perspectives. The first step which integrates multiview scenes is to combine the damaged video image frames taken by the left camera and the intact right camera video image frames for preliminary restoration. The damaged left-view image frame and the right-view image frame which is aligned through the STNs are input in the generator. In the decoding part, the generator reconstructs the restored left-view image frame. The options of the discriminator are to verify whether or not the image frame input to the discriminator is the generated distribution or the real distribution, resulting in generator generates a more realistic repaired image frame. In the encoder, we respectively perform feature extraction on the left-view damaged image frame and the spatially transformed right-view image frame through convolution. After convolution, we perform half-channel analysis on the feature maps extracted from the two perspective image frames, integrating the information of the two perspective image frames adequately. Since the images in the video sequence are partially damaged, a partial feature map of the encoder is additional to the decoder to scale back the loss of data within the uninjured space. In this way, the network focuses on the damaged area during the training process.

### Encoder–decoder

The generator of the conditional generation adversarial network within the initiative is a straightforward encoder–decoder design. This architecture reconstructs the image after encoding the image into high-dimensional features. The image content and semantic information is learned through the generator. The context encoder first combines the autoencoder with GAN for image generation. The image-to-image method uses $L_1$ distance to constrain the gap between the generated image and the real image. However, the image-to-image method only encodes and reconstructs the damaged image to get the repaired image. Because of the lack of prior information, this method cannot get a true and clear repaired image. The conditional generation adversarial network in this article introduces image frames ($y$) from the intact right-view video sequence to assist and constrain the repair of damaged image frames ($\tilde{x}$) in the left-view video sequence. The reconstruction loss can be expressed as the following form

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

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

### Spatial transform network

Since the multicamera system has a certain angle in space, spatial alignment is achieved in the damaged area between the two perspectives. Before inputting the right-view intact image frame into the generator, it undergoes spatial transformation through the STN. A spatial transform network is a module planned by Jaderberg et al. to boost the hardness of neural networks. By introducing an STN, the network will mechanically align the pictures from totally...
different views in space, so that the network can better integrate spatial information from different perspectives. In our work, the STN performs spatial transformation preprocessing on the proper perspective image to eliminate the sector of read deviation between the left and right views. Figure 5 describes the structure of the spatial transform network.

The right-view image passes through the STN, and the network outputs six affine transformation parameters ($q_{11}$, $q_{12}$, $q_{13}$, $q_{21}$, $q_{22}$, $q_{23}$). Through these six parameters, we can use the following formula to perform affine transformation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

(2)

where $(x', y')$ represents the picture element coordinates of the initial image, and $(x'', y'')$ represents the picture element coordinates of the image once the transformation. In order to get the point relationship between the target pixel and therefore the original pixel, we would like to reciprocally remodel the transformation formula

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix}$$

(3)

However, the pixel coordinates after the inverse transformation may be floating point data. We should use the neighborhood knowledge of the initial image knowledge for interpolation. Bilinear interpolation is employed here. Figure 6 shows that a floating purpose coordinate is found within the middle of four constituent coordinates.

**Conditional generative adversarial networks**

Context encoder combines autoencoder and generative adversarial loss for image restoration. As shown in Figure 1(a), it rebuilds and repairs broken images ($\tilde{x}$) to realize the inpainting result image ($G(\tilde{x})$) through the encoding and decoding operations of the generator. The discriminator makes a real or false determination between the repair target ($x$) and also the repaired result ($G(\tilde{x})$). However, it is difficult to constrain and control the image generation process using this method. Additionally, it is difficult to control the image generation.

By introducing conditional input to constrain the image generation process, it takes the damaged image ($\tilde{x}$) as another input of the discriminator, which is also called the first stage of our algorithm.

Image-to-image solves these problems by introducing conditional input to constrain the image generation process. As shown in Figure 1(b), it takes the broken image ($\tilde{x}$) as another input of the discriminator. However, due to
insufficient prior information provided by damaged images, this methodology cannot considerably improve the accuracy of the repair results. The method within the first part of this article uses a multicamera system to introduce video image frames (v) from different views to help and constrain the image restoration method. Figure 2(c) shows a schematic diagram of the tactic within the initiative of this article. We mix the broken left-view image (x̂) and also the corresponding right-view image (v) within the generator to reconstruct a whole left-view image (G(y, x̂)).

In our method, GAN loss can be described as

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

where G tries to minimize this objective against an adversarial D that tries to maximize it

\[
G^* = \arg\min_G \max_D L_{\text{CGAN}}(G, D)
\]  

Previous work \(^4,14,24\) tested that the mixture of reconstruction loss and adversarial loss will improve the smoothness of the generated image. This part utilizes image frames of multiple scenes, combined with reconstruction loss and counter loss to jointly train the network for multitask learning.

The final optimization goal is

\[
G^* = \arg\min_G \max_D \lambda L_{\text{CGAN}}(G, D)
\]

where \(\lambda_{L_{\text{CGAN}}}\) and \(\lambda_L\) are the burden of adversarial loss and the reconstructed loss severally.

**Deep convolution optical flow repair subnetwork (DFC-S)**

The second stage is to make full use of the interframe information about the image frames in the single-view video sequence to repair the damaged image frames. In order to avoid the degree of complicated calculation caused by directly filling the image pixels, the method proposed in the second step is to refine and repair the optical flow of the video sequence composed of the initial repaired image in the first step, and further spread through the optical flow to guide the video images between sequences are repaired. Figure 2 describes the main architecture of single-view video repair guided by optical flow. It mainly includes two parts. The first part completes the repair of the missing optical flow. Additionally, the second part completes the repair of the images in the video sequence under the guidance of the optical flow field.

In the first part, an optical flow refinement network (DFC-Net) from coarse to fine is mainly introduced. DFC-Net consists of three similar subnetworks (DFC-S). The first subnetwork estimates the optical flow at a relatively rough scale, and inputs the optical flow output by the first subnetwork into the second and third subnetworks for further refinement. In the second part, the propagation is guided by the optical flow from different frames, and the pixels in the damaged area are filled by the pixels in the known area. The theory of the second step method is as follows.

The inputs of the first subnetwork are the spliced tensor in the channel dimension of the optical flow map from consecutive frames of the video sequence and the binary mask image associated with each frame image. Each binary mask image represents an optical flow in the real area of the figure. The output of the network is the optical flow of the intermediate frame of the input optical flow.

Specifically, supposing \(f^0_{i-(i+1)}\) represents the initial flow between \(i\)th and \((i+1)\)th frames and \(M_{i-(i+1)}\) denotes the corresponding indicating mask. First, we use FlowNet2.0 to extract the optical flow, and initialize all the holes in the optical flow \(f^0_{i-(i+1)}\) through the interpolation method. To repair \(f^0_{i-(i+1)}\), the input of the first subnetwork are \(\{f^0_{i-(i-k+1)}, \ldots, f^0_{i-(i+1)}, \ldots, f^0_{i+(k-1)}\}\) and \(\{M_{i-(i-k)}, \ldots, M_{i+1}, \ldots, M_{i+(k)}\}\) are spliced in the channel dimension, where \(k\) denotes the length of the consecutive frames. In our work, we use \(k = 5\).

**Optical flow refinement repair network (DFC-Net)**

Figure 4 shows the overall structure of DFC-Net, which is stacked through three subnetworks (DFC-S). Generally, the smaller the actual part of the optical flow, the easier it is to repair the lost optical flow. Therefore, in the first subnetwork, the shape of the optical flow graph is reduced to obtain a good initial repair effect. Then, in the second and third subnetworks, the size of each input optical flow diagram is gradually enlarged to gradually complete the restoration from coarse to fine. Compared with the scale of the original image, the input sizes of the three subnetworks are 1/2, 2/3, and 1 of the original image, respectively.

The optical flow obtained by the first subnetwork is relatively rough repaired, and the second subnetwork focuses on the repair of the details of the optical flow. To better refine the optical flow, the second subnetwork will forward the optical flow, and it is combined with backward optical flow to complete the optical flow refinement and repair together. Supposing \(f^1\) is the coarse optical flow generated by the first subnetwork. For each continuous image frame in the video sequence (\(i\)th frame and \((i+1)\)th frame), the second subnetwork performs bidirectional optical flow estimation and uses \(\{f^1_{i-(i-k+1)}, \ldots, f^1_{i-(i+1)}, \ldots, f^1_{i+(k+1)}\}\) and \(\{f^1_{i-(i-k)}, \ldots, f^1_{i-(i+1)}, \ldots, f^1_{i+(k)}\}\) as input to generate fine optical flow.
Finally, the output of the second subnetwork is amplified in scale and input to the third subnetwork, which strictly follows the same procedure as the second subnetwork to obtain the final results. The visualization is provided in Figure 7, which shows that the process of the flow field is gradually improved through the coarse-to-fine refinement.

In the training process, the missing regions are randomly generated for each video sequence to train the three subnetworks. The goal of optimization is to minimize the distance between the predicted optical flow and the true optical flow. Specifically, the loss of each subnetwork is defined as

\[
L_i = \frac{\| M \odot (f^i - \hat{f}) \|_1}{\| M \|_1}
\]

where \( \hat{f} \) is the ground-truth flow and \( \odot \) is element-wise multiplication.

**Experiments**

The GTX 2080Ti GPU was used for training and evaluation. GTX 2080Ti has 4352 cuda cores with a base frequency 1350 MHz and a boost frequency 1545 MHz. The PC has Intel i7-7800X CPU with 3.5 GHz and 32 GB of RAM. Ubuntu 16.04 for OS and PyTorch for deep learning framework were used. In order to evaluate our proposed method intuitively and quantitatively, we compare the previous methods PatchMatch, context encoder, and image-to-image. To verify the effectiveness of video image frame restoration, an ablation experiment was carried out. The influence of various factors on the repair effect was analyzed. Figure 8 shows the repair effects of different repair methods. As shown in Figure 8(f), PatchMatch fills the space to be repaired with pixels around the broken area. When the target to be repaired is totally settled within the broken space and has completely different characteristics from the encompassing pixels, the repair cannot be completed. For example, the elongated tube in Figure 8(f) cannot be repaired by the repair technique. There are artifacts on the results repaired by the context encoder and pix2pix.
These two repair ways solely deem the linguistics data learned by the autoencoder from the scene and the information distribution generated by the GAN. The generation process of the repair results is difficult to control. Pix2pix’s method provides better results. Because jump connections are added to the pix2pix generator, there are more detailed features in the generated image. It is tough to reconstruct complete pictures and details solely by hoping on linguistics options. By introducing info from alternative views, there square measure a lot of constraints and steerage within the image generation method. Figure 8(e) shows the repair effect of this module. When the reconstruction loss is removed, the generation method of the multiview module is additional random and chaotic (Figure 8(l)). Additionally, if the adversarial loss is removed, the inpainting result shows a smoothing effect (Figure 8(e)). Finally, if the optical flow is removed, the inpainting result would be better than the results for other methods.

In order to quantitatively analyze the methods proposed in this article, we compared the previous methods from the three aspects of $L_1$ distance, $L_2$ distance, and peak signal-to-noise ratio (PSNR).

Where, the typical $L_1$ Loss is that the definite quantity of the typical constituent value distinction between the inpainted image and also the target image

$$L_{1\text{Loss}} = E_{x, \tilde{x}} [||\tilde{x} - x||_1] \quad (8)$$

$L_2$ Loss is the square of the distinction between the typical pixel worth of the inpainted image and also the target image

$$L_{2\text{Loss}} = E_{x, \tilde{x}} [||\tilde{x} - x||^2] \quad (9)$$

![Figure 8](image-url)  
**Figure 8.** Compare different image inpainting methods. (a) Left camera damaged image, (b) right camera intact image, (c) left camera inpainting target, (d) spatial transform image, (e) our method inpainting result, (f) PatchMatch result, (g) context-encoder result, (h) image-to-image result, (i) our method without flow module, (j) our method without STN, (k) our method without GAN loss, and (l) our method without $L_1$ loss.

| Method                  | Mean $L_1$ loss (%) | Mean $L_2$ loss (%) | PSNR (db) |
|-------------------------|---------------------|---------------------|------------|
| PatchMatch              | 1.389               | 5.305               | 26.218     |
| Context-encoder         | 3.124               | 5.644               | 25.221     |
| Image-to-image          | 1.451               | 4.608               | 26.980     |
| Ours (remove flow propagation) | 0.985             | 3.018               | 30.745     |
| Ours (remove STN)       | 0.998               | 3.047               | 30.728     |
| Ours (remove GAN loss)  | 1.299               | 3.455               | 29.533     |
| Ours (remove $L_1$ loss) | 4.010              | 7.298               | 23.224     |
| Ours                    | 0.973               | 3.014               | 30.752     |

Table 1. Quantitative result for various inpainting strategies.

Note. $L_1$ distance and $L_2$ distance are the pixel-by-pixel difference between the restoration result and the target image, respectively, which measure the difference between the restoration result and the target image. The larger the PSNR value between the two images, the more natural the inpainted image is.

Peak Signal to Noise Ratio (PSNR) is a standard used to measure the image distortion or noise levels. The unit is decibel, which is outlined as follows

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

shows the quantitative results of the experiment. Our method shows good repair effects on $L_1$ loss, $L_2$ loss, and PSNR. Table 1 shows the quantitative results of the experiment. Our method shows good repair effects on $L_1$ loss, $L_2$ loss, and PSNR. Experiments demonstrate that the method which combines the multiview scene and the information
between frames in the video sequence is of significance on
the improvement of the inpainting result.

In addition, we tend to additionally take a look at on
broken areas with random areas, random traits, and random
colors. As shown in Figure 9, the results show that the
formula has sensible strength.

**Conclusion**

This article proposes a video repair method that com-
bines multiview information under the guidance of opti-
cal flow. We combine the network model based on the
conditional generation adversarial network and the opti-
cal flow information in the video sequence. Under the
multiview scene model, we perform spatial transforma-
tion process on pictures from different views to elimi-
nate viewing angle deviations. Through grouping
convolution and channel exchange with feature fusion
of multiview images combined with optical flow, the
method makes full use of timing information in video
sequences to repair the video. Finally, a qualitative and
quantitative chemical analysis of the repair impact of the
strategy was dispensed through ablation experiments,
which verified the dependableness of the strategy
adopted by the strategy. The method can be applied to
scenes in which images are damaged due to the actual
working environment and the influence of its own image
acquisition equipment during the image acquisition pro-
cess of the binocular robot and the automatic driving
multiview system. Using this method to repair a video
image from a certain perspective is convenient for the
subsequent video image processing process and com-
pletes the subsequent autonomous environment percep-
tion tasks of the robot system and the autonomous
driving system.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with
respect to the research, authorship, and/or publication of this
article.

![Figure 9. Damaged area image repair results with random areas, random shapes, and random colors. (a) Left camera damaged image, (b) right camera intact image, (c) left camera inpainting target, (d) result without multiview scene, and (e) our method inpainting result.](image-url)
**Funding**
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work described in this article was supported by the National Natural Science Foundation of China (grant numbers 61625304, 62073209, 61703181, and 61991415) and the Shandong Provincial Natural Science Foundation of China (grant number ZR2017BF021).

**ORCID iD**
Hengyu Li https://orcid.org/0000-0002-2243-5908

**References**
1. Li Y, Hu Z, Li Z, et al. Multiscale site matching for vision-only self-localization of intelligent vehicles. IEEE Intell Transp Syst Mag 2018; 10(3): 170–183.
2. Guindel C, Martin D, and Armingol JM. Traffic scene awareness for intelligent vehicles using ConvNets and stereo vision. Rob Auton Syst 2019; 112(3): 109–122.
3. Ma R, Maugéy T, and Frossard P. Optimized data representation for interactive multiview navigation. IEEE Trans Multimedia 2018; 20(7): 1595–1609.
4. Pathak D, Krahenbuhl P, Donahue J, et al. Context encoders: feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2016.
5. Saha R, Debi T, and Arefin MS. Developing a framework for vehicle detection, tracking and classification in traffic video surveillance. 2020 3rd International Conference on Intelligent Computing and Optimization, 2020.
6. Chang J, Wang L, Meng G, et al. Vision-based occlusion handling and vehicle classification for traffic surveillance systems. IEEE Intell Transp Syst Mag 2018; 10(2): 80–92.
7. Liu H, Lee S, and Chahl JS. A multispectral 3-D vision system for invertebrate detection on crops. IEEE Sensors J 2017; 17(22): 7502–7515.
8. Wang C, Fang Y, Zhao H, et al. Probabilistic Inference for occluded and multiview on-road vehicle detection. IEEE Trans Intell Transp Syst 2016; 17(1): 215–229.
9. Liu L, Li H, Dai Y, et al. Content-preserving video stitching method for multi-camera systems. IEEE Trans Intell Transp Syst 2018; 19(8): 2432–2444.
10. Ji X, Zhang G, Chen X, et al. Multi-perspective tracking for intelligent vehicle. IEEE Trans Intell Transp Syst 2018; 19(2): 518–529.
11. Bertalmio M, Bertozzi AL, and Sapiro G. Navier-stokes, fluid dynamics, and image and video inpainting. IEEE Comput Soc 2001; 1(1): 355–362.
12. Huang JB, Kang SB, Ahuja N, et al. Temporally coherent completion of dynamic video. ACM TOG 2016; 35(6): 196.
13. Patwardhan KA, Sapiro G, and Bertalmio M. Video inpainting under constrained camera motion. IEEE Trans Image Process 2007; 16(2): 545–553.
14. Isola P, Zhu J, Zhou T, et al. Image-to-image translation with conditional adversarial networks. Proc IEEE Conf Comput Vis Pattern Recognit, CVPR 2017; 16(2): 5967–5976.
15. Hinton GE and Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science 2006; 313(5786): 504–507.
16. Bengio Y. Learning deep architectures for AI. Found Trends Mach Learn 2009; 2(1): 1–127.
17. Li Y, Liu S, Yang J, et al. Generative face completion. Proc IEEE Conf Comput Vis Pattern Recognit CVPR 2017; 60(1): 5892–5900.
18. Mirza M and Osindero S. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.
19. Ronneberger O, Fischer P, and Brox T. U-Net: convolutional networks for biomedical image segmentation. Int Conf MICCAI 2015; 2(1): 234–241.
20. Li C and Wand M. Precomputed real-time texture synthesis with Markovian generative adversarial networks. Lect Notes Comput Sci 2016; 16(2): 702–716.
21. Krizhevsky A, Sutskever I, and Hinton GE. ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 2012; 16(2): 1097–1105.
22. Xu R, Li X, Zhou B, et al. Deep flow-guided video inpainting. Proc IEEE Comput Soc Conf Comput Vision Pattern Recognit 2019; 2019(1): 3718–3727.
23. Jaderberg M, Simonyan K, Zisserman A, et al. Spatial transformer networks. IEEE Trans Intell Transp Syst 2015; 20(3): 201–229.
24. Larsen AB, Sønderby SK, Larochelle H, et al. Autoencoding beyond pixels using a learned similarity metric. IEEE Trans Intell Transp Syst 2015; 18(2): 215–229.
25. Kim B, Choi K, Park W, et al. Content-preserving video stitching method for multi-camera systems. IEEE Trans Concur Electron 2017; 63(2): 109–116.
26. Bozorgtabar B and Goedecke R. MSMCT: multi-state multi-camera tracker. IEEE Trans Circuits Syst Video Technol 2018; 28(12): 3361–3376.
27. Vincent P, Larochelle H, Bengio Y, et al. Extracting and composing robust features with denoising autoencoders. Int Conf Mach Learn Proc 2008; 2(1): 1–127.