Sensor Applications

# Multi-view Scene Image Inpainting Based on Conditional Generative Adversarial Networks

Michael Shell[1*], John Doe[2], and Jane Doe[2**]

[1] Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA
[2] Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, India
[*] Member, IEEE
[**] Senior Member, IEEE

Manuscript received June 7, 2017; revised June 21, 2017; accepted July 6, 2017. Date of publication July 12, 2017; date of current version July 12, 2017.

Abstract—Multi-views systems have been widely used in robots, ADAS(Advanced Driver Assistance Systems), monitor systems and so on, using multi-views, the machine can better perceive the surrounding scenes. The exposed lens and the camera are easily contaminated by the outside, resulting in abnormal images. Image inpainting technology can utilize the prior information of the image structure, texture and other information provided by the surrounding pixels of the abnormal area to recover the damaged image, which can reduce the loss of visual information, providing as much information as possible for the machine's decisions. In order to achieve the above purposes, considering the characteristics of multi-vision system, a novel image inpainting method is proposed. The basic idea is that using conditional generative adversarial networks(CGAN) to amend defect images, in which the priori condition is the synchronization frame from other cameras in different viewpoints. The generator in the CGAN is a autoencoder which has skip connected from encoder to decoder. We also integrate spatial transform networks, group convolution and channel switching technology in our network structure to better fusion the multi-views information. Experimental results show the advantage of our method.

Index Terms—Image inpainting, generative adversarial networks, convolutional neural network, deep learning.

## I. INTRODUCTION

Image inpainting means to restore the defective image according to the image texture, structure and other information. It has been broad applied in many field, such as defect images restoration [1], [2], video communication error repairing [3], [4] and photo editing [5], [6]. With the development of image and video processing technology, visual information has played a key role in the field of automation. Due to the limited information available from monocular cameras, the multi-views system is widely used. Fig 1 show a typical multi-views system—a vehicle equipped with four cameras to detection objects [7]. Some reasons easily cause abnormal images. First, the camera lens were blocked by rain, snow or mosquitoes; Second, losing some information in the process of image signal compression, transmission and decompression. When the autonomous vehicles are running and these unexpected things happened, would lead to traffic accidents. In order to automatically restore the abnormal images on driving, we propose a novel image inpainting method based on multi-views. Our method can be used on other multi-views systems.

Image inpainting has made tremendous progress in the past nearly two decades. Many methods has been proposed which can be divided into two sets. The first set of approaches relies on texture synthesis techniques, which fills in the hole by extending the textures of the surrounding area [1], [8]–[10]. What these techniques have in common is the use of patches with similar textures to synthesize the content of the hole region from coarse to fine. Drori et al. [8] and Wilczkowiak et al. [9] introduced multiple scales and orientations to find better matching patches. Barnes et al. [10] used the fast approximate nearest neighbor algorithm to search the match patches. Such methods are good at propagating high-frequency texture details. When part ot the object is missing, using these methods can perfect restore, but it's hard to use these methods to reproduce the small object when the whole object is missing. Fig 2d show the result using Barnes et al. [10] method to restore the defective image (Fig 2a). Compare the result with the target (Fig 2c), we can find part of the black coat is restore and some small pedestrians fall in the blank region is not reproduce. The second set of approaches solve this problem in a data-driven way [11], involving a cut-paste formulation using nearest neighbors from a dataset of millions of images. This approach is very effective when it finds an example image with sufficient visual similarity to the query but could fail when the query image is not well represented in the database. A serious problem is that the image restored with this method seems reasonable, but the image content is quite different from the target image. Furthermore, it is struggles to fill arbitrary holes, e.g. objects are partially missing. Additionally, the data-driven way restricts application scenarios.

With the continuous updating and development of convolutional neural network, various tasks of computer vision have been breakthrough. Image inpainting technology has also been improved. Autoencoders [12], [13] encode image to a low-dimensional "bottleneck", decode it by reconstructing the high-dimensional image from the "bottleneck". The purpose of doing this to obtain the compact

Corresponding author: M. Shell (e-mail: nospam@nospam.org).
(For the real e-mail address, see http://www.michaelshell.org/contact.html).

Associate Editor: Alan Smithee.
Digital Object Identifier 10.1109/LSENS.2017.0000000

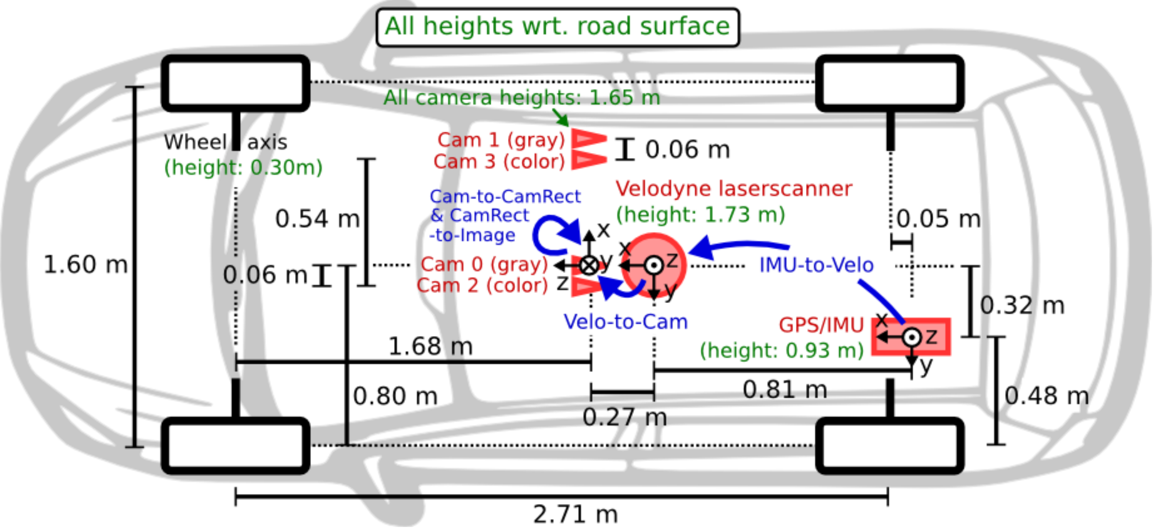
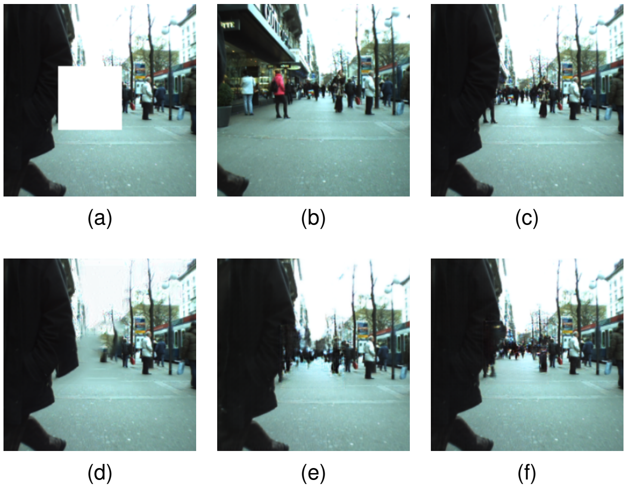


Fig. 1. A vehicle equipped with four cameras(Cam0~Cam3).


(a) (b) (c)

(d) (e) (f)

Fig. 2. Qualitative illustration of the different image inpainting methods. (a) Given an image with a missing region captured by left camera. (b) Given the same scene image captured by right camera. (c) The target of the left image inpainting. (d) PatchMatch method result. (e) Image-to-Image method result. (f) Our method result.

feature representation of the scene. Denoising autoencoders [14] reconstruct the image from corrupted status to learn more robust features. Using denoising autoencoders to inpaint defective image can get blurred filling in the blank area. Generative adversarial nets [15] (GAN) can learn the distribution of real data, using GAN can generate images that correspond to train data [16]. Pathak et al. [17] combined autoencoders and GAN for image inpainting. They used autoencoders as the generator in GAN architecture, combining autoencoders reconstruct loss and GAN adversarial loss to do image inpainting get sharpness results. Li et al. [18] used the same idea to do face completion. Mirza et al. [19] introduced a condition into GAN to control the processes, which can generate a special image according to the condition. Isola et al. [20] further developed the idea of Pathak. Generator adopt the autoencoders with skip connection from encoder to decoder, like the UNet [21] structure. Discriminator added input image as condition, learned to classify between fake (input image,inpainting result) and real (input image, target image) tuples. Another different in discriminator is that they use a convolutional "PatchGAN" classifier, which only penalizes structure at the scale of image patches. The PatchGAN architecture was firstly applied in [22] to capture local style statistics. Fig 2e show this method produce a plausible hypothesis for the missing part(s), but the details are not same with the target image. Only used generator learned the real images distribution to conjure up the scene "out of thin air" is hard to produce image same as the target image.

In this paper, we proposed a novel image inpainting method, which can be used in multi-views system. Our methods is fusion other viewpoint images to restore defective image. The basic structure of our method is condition generation networks, in which the priori condition is the synchronization frame of other cameras from different viewpoints. In order to make the images from other viewpoints better guide the anomalous image to be repaired, the spatial transform networks [23] are introduced to carry out affine transform for other viewpoints images to achieve the purpose of multi-view scene alignment in the abnormal area. In order to better utilize the complete information from other viewpoints into the defect image, group convolution [24] and channel shuffle [25] are used to process these images and fuse information. Group convolution also serves the purpose of reducing the amount of parameters and computation. The whole method combines reconstruction loss and confrontation adversarial loss, integrated spatial transform processing, group convolution and channel shuffle technologies to achieve a high quality inpainting result. Fig 2f shows our method result is the most same with target.

## II. PROPOSED ALGORITHM

In the paper, we propose a novel image inpainting method which can be used in multi-view system as an emergency remedy when the cameras happened some unexpected things. We fuse multi-view images to restore the abnormal image. Our approach is based on convolutional neural networks, specifically based on condition generation networks. Fig 3 shows the overall frameworks of our method. It consists of a generator and a discriminator where the

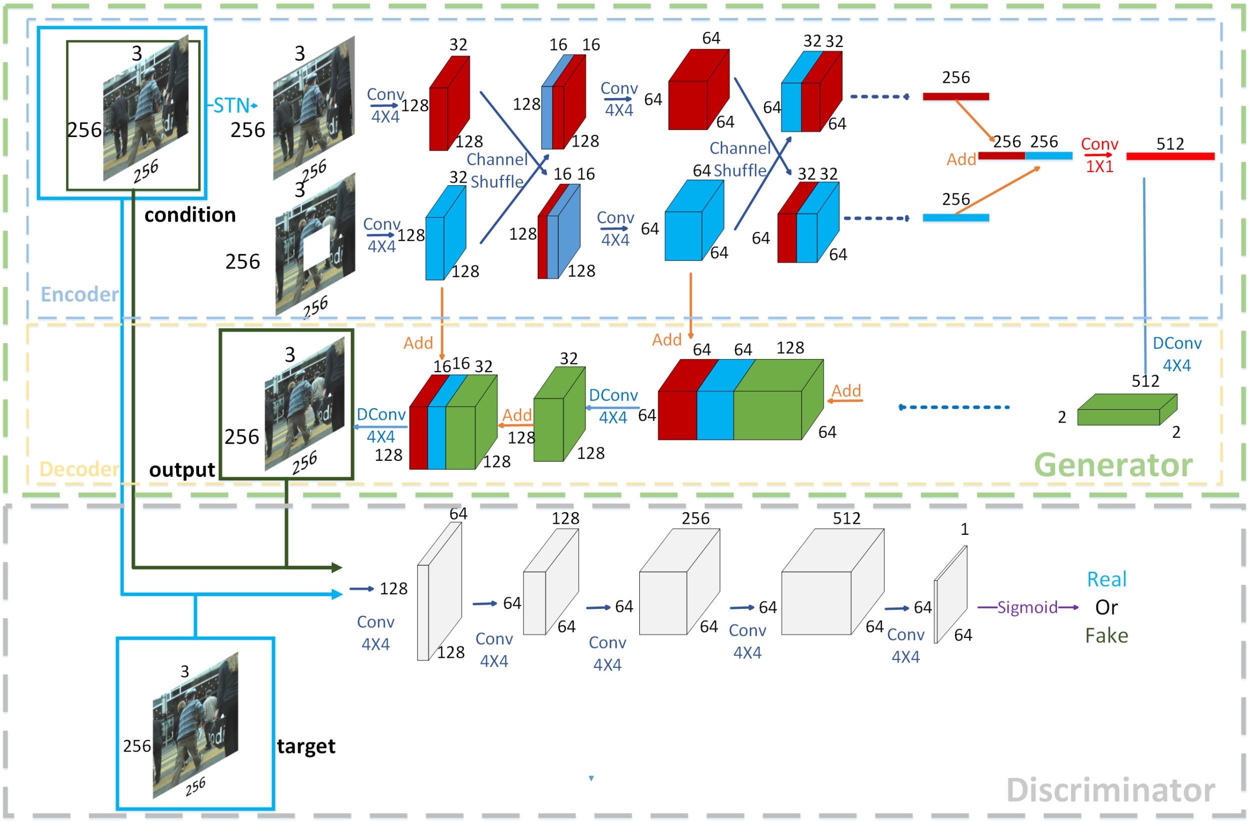


Fig. 3. The network architecture of our method.

generator is a autoencoder consisting of an encoder and a decoder. We test the proposed method in the public dataset [26], what is acquired by the ETH Zurich vision group using using a mobile platform equipped with two cameras. We damaged the left camera image and tried to restore it int help of the right camera image. The generator encode the left camera damaged image and the right camera image, then decode them to reconstruct the sound left image in order to fool the discriminator. The discriminator learn to classify between fake right image, synthesized left image and real right image, left image tuples. The encoder adopt group convolution and channel shuffle to full exchanging and fusing information between two camera. There also has skip connected from encoder to decoder to give the generator a means to circumvent the bottleneck for information. At the begin of the generator, the spatial transform networks is carried out on the right image to achieve the purpose of context alignment in the abnormal area. Experiments will show that all the strategies adopted in our method are effective.

## A. Encoder-decoder

The generator is a simple encoder-decoder pipeline. This architecture try to reconstruct image after passing it to a low-dimensional bottleneck layer. By doing this, the networks learned the image content and semantically [12]–[14]. Pathak et al. [17] first integrate this architecture in their Context-Encoders method to do image inpainting. They use L2 distance to capture the overall structure of the missing region in relation to the context. Isola et al. [20] using L1 distance replace L2 distance to reduce the blurring in their Image-to-Image method. Our method also adopts L1 distance to reconstruct the original left image, the difference is that we joint the abnormal left image($\tilde{x}$) and corresponding right image($y$) to achieve the goal:

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

Like Image-to-Image method, we add skip connected from encoder to decoder in each layer. This strategy increase the information flow from encoder to decoder and decrease the difficulty of reconstructing, so that the generator can focuses on the recovery of the abnormal areas. We also introduce spatial transform, group convolution and channel shuffle in generator to better fuse the left and right image information.

## B. Spatial Transform

Spatial transformer networks(STN) [23] is a learnable module, it can be inserted into convolutional architectures, giving neural networks the ability to actively spatially transform feature maps. Using STN, neural networks can select the most important area of the the image and transform it into the optimal posture which are benefit for the task. We use STN at the begin of the encoder to carry out affine transformation on the right image, in order to achieve the scene aligning between the two image. Fig 4 show the STN models applied in our method. The model input the right image, through the localisation net(four convolutional layers and two fully connected layer) to obtain the six affine transformation coeficients($\theta$). Having these coeficients we can do affine transform using the follow formula:

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

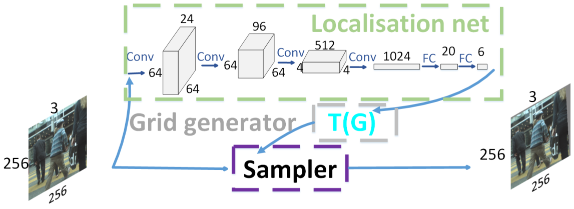


Fig. 4. The spatial transform networks used in our method.

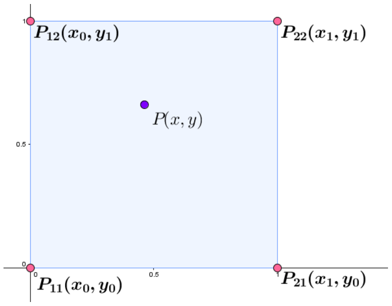


Fig. 5. Bilinear interpolation schematic.

Transformed the above formula can we get the mapping of the target image pixel coordinates at the original image pixel coordinates:

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

In this way, the transformation target image becomes pick up pixels from original image:

$$PixelMatrix^{Target} = PixelMatrix^{Source}[x^{Source}, y^{Source}] \tag{4}$$

However, the $x^{Source}$ and $y^{Source}$ may be floating numbers, which does not correspond to the arbitrary integral coordinate values of the original image. We need to take advantage of the local approximation principle of the image data, take the adjacent pixels to do the average generation, namely, interpolation. Fig show a pixel coordinate locates in the middle of four pixels. We use bilinear interpolation to get the 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 what the T(G) doing. The "sampler" to finish the transform from input image to target.

## C. Group Convolution and Channel Shuffle

Group convolution is the network optimization design method proposed in MobileNets [24] . The idea is that implement group convolution first and then using pointwise convolution to merge these group feature map. This can reduce the amount of network parameters and computation. This idea also used in Xception [27] and ResNeXt [28]. Due to the costly dense pointwise convolutions, shuffleNet [25] extends the idea of group convolution, using pointwise group convolution and channel shuffle operation to further reduce the number of parameters and calculation. We introduce group convolution and channel shuffle in our method to increase the exchange of information between multiple views. Contrastive experiments found that these measures are effective. In order to further reduce the amount of parameters and computation, we only perform channel shuffle after group convolution, and only carry out a pointwise convolution after bottleneck layer to fusion these feature map.

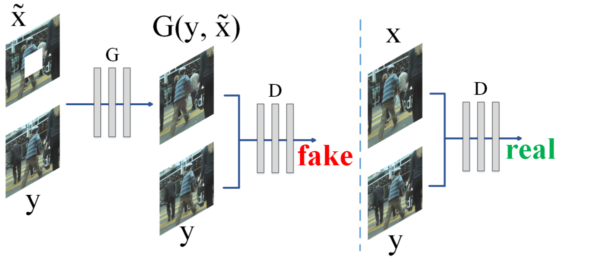


Fig. 6. Multi-view scene image inpainting based on conditional generative adversarial networks method diagram.

## D. Conditional Generative Adversarial Networks

Generative Adversarial Networks [15] is composed of a generator(G) and a discriminator(D). Discriminators can authenticate real data and generate data, and the generator's goal is to generate more realistic data to fool the discriminator. This is a two-player game, the discriminator improve the ability to distinguish between true and false by gradually and generator learns to produce more realistic sample to deceive the discriminator. Through training iteration, the generator can produce "real data", that is, the generator learned the distribution of the real data. However, the modes of the generator produces is stochastic and uncontrollable. Mehdi et al. [19] solve this problem by feeding a condition on both the generator and discriminator, proposed conditional generative adversarial networks(CGAN). Our image inpainting method based on CGAN architecture, which the condition is the other viewpoint image. Fig 6 show the diagram of our method. In generator, we use both left defective image($\tilde{x}$) and the corresponding right image(y) to reconstruct the original intact left image($G(y, \tilde{x})$). The discriminator distinguishes between fake $\{y, G(y, \tilde{x})\}$ and real $\{y, x\}$ tuples.

The objective of the CGAN can be expressed as

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

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

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

## E. Joint Loss Function

A lot of previous researches have demonstrated that only use reconstructed loss(L1 loss or L2 loss) to generation image can produce blurry results [17], [20], [29], adding adversarial loss results in much sharper predictions. Our method combine multiple viewpoint images, jointing reconstructed loss and adversarial loss to inpaint image can produce high quality results. The final objective is

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

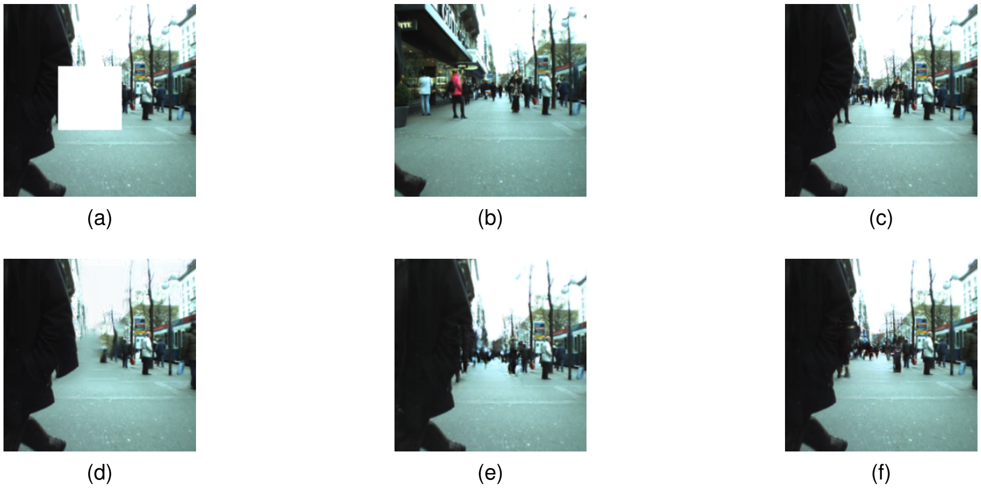


Fig. 7. Qualitative illustration of the different image inpainting methods. (a) Given an image with a missing region captured by left camera. (b) Given the same scene image captured by right camera. (c) The target of the left image inpainting. (d) PatchMatch method result. (e) Image-to-Image method result. (f) Our method result.

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

# III. EXPERIMENTS

## A. Database Description

In order to evaluate the image inpainting effect of our method, we conducted experiments on a public dataset collected by the a pair of AVT Marlins F033C mounted on a chariot respectively a car, with a resolution of 640 x 480 (bayered), and a framerate of 13–14 FPS [26]. The dataset consists of 8 consecutive time series, a total of 5263 pairs of synchronized time frame images. Experiment randomly take 4208 pairs to train, 1055 images to do test. The left and right images are all resized to 256 x 256 before input to the network. We set a blank area in the middle of the left camera images with 76 pixels (the RGB pixel values are set to 255). We train and and test our the proposed method using this defective left camera images and the corresponding right camera images. In the end, we also carry out experiments with a stochastic shape and color block areas in the left camera images.

## B. Experimental Setup

In the encoder each convolution is adopt "Convolution —> Batch Normalization —> Relu" structure. The convolution kernal size is $4\times4$, with 1 padding and 2 stride. After one convolution, the height and width of the feature maps cut half and channels increase twice. however, when the numbers of channels reaches to 512, keep it and don't increase any more. At the bottleneck layer, the feature map dimensionality is $1\times1\times1$. We add a pointwise convolution in the later. In the decoder, we adopt "Deconvolution —> Batch Normalization —> Leak Relu" structure as the reverse of convolution to reconstruct the original left camera image. The discriminator and the localisation networks structure are showed in Fig 3 and Fig 4 respectively.

We build our models in tensorflow. We set the reconstructed loss weight $\lambda_{L_1}$ to 100 and adversarial loss $\lambda_{L_{CGAN}}$ to 1, using the back-propagation algorithm with the Adam [30] optimization to tune the parameters.

## C. Experimental Results

# IV. CONCLUSION

The conclusion goes here.

# ACKNOWLEDGMENT

This work was supported by the IEEE. The authors would like to thank ... Note that the Acknowledgment section of *IEEE Sensors Letters* is rendered in scriptsize.

# REFERENCES

[1] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. Seventh IEEE Int. Conf. Computer Vision*, vol. 2, 1999, pp. 1033–1038 vol.2.

[2] Z. Lu, H. Huang, L. Li, and D. Cheng, "A novel exemplar-based image completion scheme with adaptive TV-constraint," in *Proc. Fourth Int. Conf. Genetic and Evolutionary Computing*, Dec. 2010, pp. 94–97.

[3] 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 Processing*, vol. 1, 2002, pp. I–309–I–312 vol.1.

[4] S. D. Rane, G. Sapiro, and M. Bertalmio, "Structure and texture filling-in of missing image blocks in wireless transmission and compression applications," *IEEE Transactions on Image Processing*, vol. 12, no. 3, pp. 296–303, Mar. 2003.

[5] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[6] T. K. Shih and R.-C. Chang, "Digital inpainting - survey and multilayer image inpainting algorithms," in *Proc. Third Int. Conf. Information Technology and Applications (ICITA'05)*, vol. 1, Jul. 2005, pp. 15–24 vol.1.

[7] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2012, pp. 3354–3361.

[8] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," in *ACM Transactions on graphics (TOG)*, vol. 22, no. 3. ACM, 2003, pp. 303–312.

[9] M. Wilczkowiak, G. J. Brostow, B. Tordoff, and R. Cipolla, "Hole filling through photomontage," in *BMVC 2005-Proceedings of the British Machine Vision Conference 2005*, 2005.

[10] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics-TOG*, vol. 28, no. 3, p. 24, 2009.

[11] J. Hays and A. A. Efros, "Scene completion using millions of photographs," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3. ACM, 2007, p. 4.

[12] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

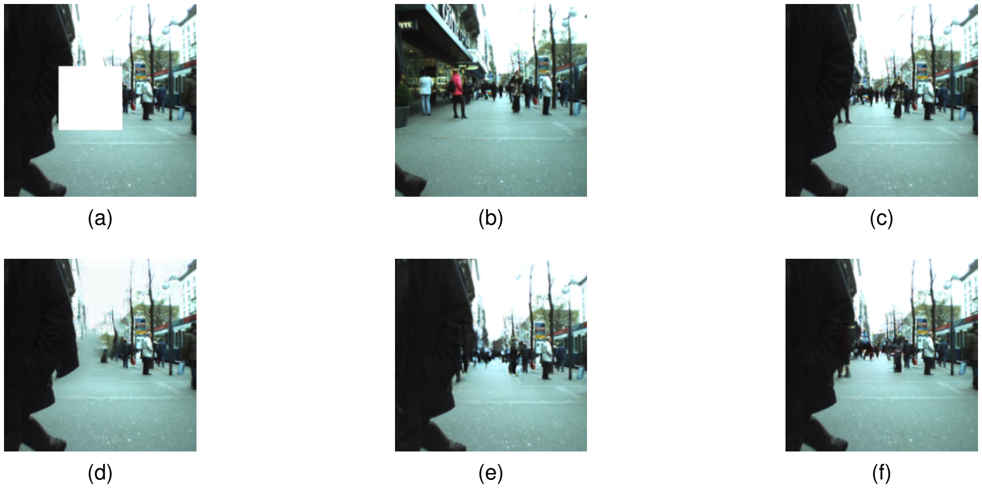


Fig. 8. Qualitative illustration of the different image inpainting methods. (a) Given an image with a missing region captured by left camera. (b) Given the same scene image captured by right camera. (c) The target of the left image inpainting. (d) PatchMatch method result. (e) Image-to-Image method result. (f) Our method result.

[13] Y. Bengio, "Learning deep architectures for ai. foundations and trends r in machine learning, 2 (1): 1–127, 2009," *Cited on*, p. 39.

[14] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[16] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *ICLR*, 2016.

[17] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.

[18] Y. Li, S. Liu, J. Yang, and M.-H. Yang, "Generative face completion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 3, 2017, p. 6.

[19] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *Computer Science*, pp. 2672–2680, 2014.

[20] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[22] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 702–716.

[23] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[24] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv1704.04861*, 2017.

[25] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet an extremely efficient convolutional neural network for mobile devices," *arXiv preprint arXiv1707.01083*, 2017.

[26] A. Ess, B. Leibe, K. Schindler, and L. V. Gool, "A mobile vision system for robust multi-person tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2008, pp. 1–8.

[27] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1800–1807.

[28] S. Xie, R. Girshick, P. DollÃąr, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 5987–5995.

[29] A. Boesen Lindbo Larsen, S. Kaae Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *arXiv preprint arXiv:1512.09300*, 2015.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

## REFERENCES

[1] H. Kopka and P. W. Daly, *Guide to LaTeX*, 4th ed. Boston, MA: Addison-Wesley, 2004.