Fast event-inpainting based on lightweight generative adversarial nets*

LIU Sheng (刘盛)**, **CHENG Haohao** (程豪豪), **HUANG Shengyue** (黄圣跃), **JIN Kun** (金坤), **and YE Huanran** (叶焕然)

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

(Received 27 December 2020; Revised 16 January 2021)

Event-based cameras generate sparse event streams and capture high-speed motion information, however, as the time resolution increases, the spatial resolution will decrease sharply. Although the generative adversarial network has achieved remarkable results in traditional image restoration, directly using it for event inpainting will obscure the fast response characteristics of the event camera, and the sparsity of the event stream is not fully utilized. To tackle the challenges, an event-inpainting network is proposed. The number and structure of the network are redesigned to adapt to the sparsity of events, and the dimensionality of the convolution is increased to retain more spatiotemporal information. To ensure the time consistency of the inpainting image, an event sequence discriminator is added. The tests on the DHP19 and MVSEC datasets were performed. Compared with the state-of-the-art traditional image inpainting method, the method in this paper reduces the number of parameters by 93.5% and increases the inference speed by 6 times without reducing the quality of the restored image too much. In addition, the human pose estimation experiment also revealed that this model can fill in human motion information in high frame rate scenes.

Document code: A **Article ID:** 1673-1905(2021)08-0507-6

DOI https://doi.org/10.1007/s11801-021-0201-8

The event camera is a neuromorphic optical sensor, which can capture the pixel-by-pixel brightness change asynchronously, that is, the event. Because of their high time resolution, low power consumption and high dynamic range, they have attracted more and more attention in computer vision field[1]. In addition, the event cameras filter out redundant information, because their output essentially only reflects the temporal dynamics of the recorded scene, while ignoring the static and non-moving areas.

Unlike traditional images, event images are obtained by accumulating events in specific time interval or a constant number of events. Therefore, the adjustable range of the time resolution of event image sequences is very wide. However, the high temporal resolution and high spatial resolution of the event image are contradictory. In the case of high temporal resolution, only a few hundred pixels of the event image record brightness change information. At the same time, the event camera will not produce any events on the occluded area or stationary objects. As a result, some spatiotemporal information in the scene we are interested in cannot be caught (see Fig.1(a)).

Specifically, in DHP19[2], in order to obtain complete human motion information (such as human pose), 7 500 events need to be accumulated in each frame, thus losing the fast response characteristic of the event camera (the frame rate is consistent with that of the traditional camera). If human motion information with high temporal resolution is needed, it can only be obtained by reducing the spatial resolution of the image. Therefore, it is necessary to fill in missing or increasing the interesting spatiotemporal information in the event image.

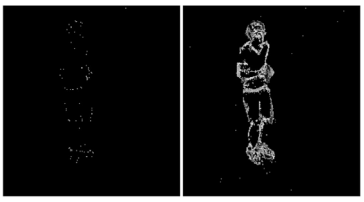

Fig.1 Event frames: (a) 100 events per frame; (b) 7 500 events per frame

In the aspect of information enhancement, a large number of results of restoring events to intensity images have been obtained. In Ref.[3], the intensity images are used as complementary information, but the resulting video is blurred. Wang et al[4] proposed multi-stage adversarial training and generated high-quality super-resolution

* This work has been supported by the National Key Research and Development Program of China (No.2018YFB1305200), and the Science Technology Department of Zhejiang Province (No.LGG19F020010).

** E-mail: edliu@zjut.edu.cn

intensity images in an unsupervised manner. Reconstructing events directly into intensity images will retain too much redundant information (such as the redundant static background), which will seriously affect the inference speed of downstream tasks (such as human pose estimation). Therefore, this paper only fills the missing spatiotemporal information based on event information.

For traditional images, deep learning has achieved significant performance improvements in image inpainting[5,6] and video inpainting[7,8]. These schemes use the learned data distribution to fill in missing pixels. However, these works have the following shortcomings for event images: event images only contain low-level visual information (motion contour), and the redundancy of existing model parameters is too large, which leads to the inference speed cannot meet the application of event cameras with high time resolution; most of the work uses custom masks to simulate missing areas and uses masks as a known prior. When missing areas are unknown, there is nothing to do, which limits the application of existing restoration algorithms to event images.

To tackle the aforementioned problems, we probe the event-inpainting network. To the best of our knowledge, this is the first work for event image sequences inpainting. We design a lightweight 3D CNNs completion network to generate coherent structures in the missing regions' event image. In addition, unlike image inpainting, video inpainting has to be temporally coherent. Therefore, we add a temporal sequence discriminator to enhance the time consistency. Besides, the frame discriminator keeps an eye on the spatial feature coherence of observation frames. We evaluate our proposed model on MVSEC[9] and DHP19. The results obtained prove the effectiveness of our method. We also compare the performance of our model against current state-of-the-art schemes. Finally, we evaluate performance on DHP19 for the human pose estimation, and show its potential for high frame rate.

This paper presents a lightweight GAN for event inpainting, and Fig.2 shows the general framework. The framework mainly includes a generator G, an event frame discriminator D_f and an event sequence discriminator D_s. The method can input event image sequences with high temporal resolution and low spatial resolution, and can also input event image sequences with normal temporary resolution but impaired spatial resolution, and output filled event image sequences after passing through the generator. Then, the filled event image sequence and the real label are sent to two discriminators, which discriminate the authenticity and feed the results back to the generator, thus ensuring the time consistency and image quality of the filled event image sequence.

Event cameras are bio-inspired sensors that asynchronously and independently report logarithmic intensity changes[10]. Thus, the output of an event-based camera can be viewed as a continuous stream of events $\{e_i\}$, $i \in N$. Each event e_i can be represented using the following form:

$$e_i=(x_i, y_i, t_i, p_i), \tag{1}$$

where (x_i, y_i) denotes the spatial location of the pixel generating the event, t_i represents temporal coordinates of a brightness change, and $p_i \in \{-1, 1\}$ indicates the positive or negative changes in intensity at the pixel causing the event.

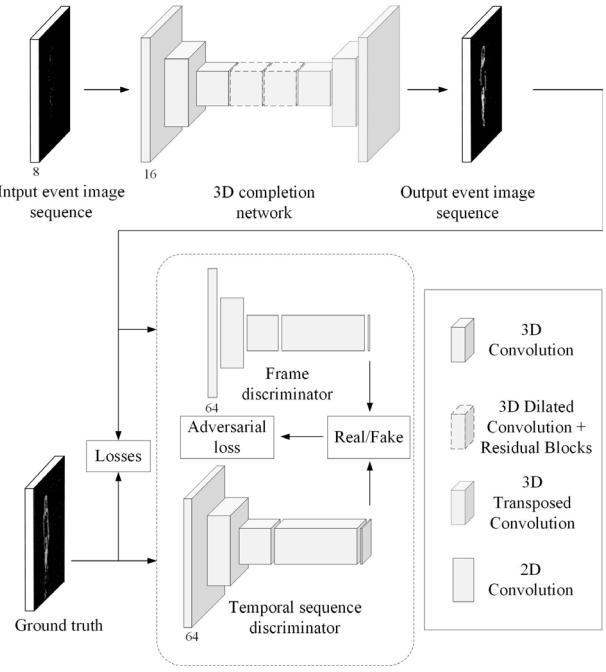

Fig.2 Overview of the proposed framework

Following Stefano[11], during an exposure time interval $\Delta t=[t, t+\tau]$, an event frame $F_\tau(t)$ is obtained by summing up all events between time t and $t+\tau$ at a pixel-wise level. Formally, an event frame can be expressed as:

$$F_\tau(t)=\sum_{e_i \in E_{t,\tau}} p_i, \tag{2}$$

in which $E_{t,\tau}=\{e_i | t_i \in [t, t+\tau]\}$. In this manner, an event frame could be represented as a gray-level image of size $1 \times w \times h$, which integrates all events occurred in a certain time interval in a single channel.

Given the ground truth of event image sequences I_{gt} and its corresponding binary mask M_s (value 0 for known pixels and 1 denotes unknown ones), the input event image sequences are defined as $I_{in}=I_{gt} \odot (1-M_s)$, the output I_{pred} predicted by the network.

The encoder-decoder structure is adopted in generator in this paper. Traditional image restoration network is too deep for event images, which leads to slow inference speed. In addition, the large-capacity networks are easier to overfit the sparse data, and lightweight architecture is more suitable for the sparsity of the event image, which has been verified in Ref.[12]. Therefore, the encoder in this paper only down-samples the image twice, and the corresponding decoder only up-samples the image twice. At the same time, because of the sparsity of the event image, the shallow network will not make the generation

quality of the event image too low, so only two residual blocks are used between the encoder and decoder[12]. In order to preserve more spatiotemporal information, 3D convolution is used instead of 2D convolution. Considering the influence of the receptive field, the regular convolution is replaced by the expansion convolution with an expansion factor of 2 in the residual layer. In order to improve the generalization ability, we use case standardization[13] on all layers of the network. Finally, the total number of parameters of the generator is about 781k, which is far less than the traditional image inpainting model, and can meet the application requirements of the high frame rate.

As for the discriminators, we have two kinds of discriminators, both of which use the 70×70 PatchGAN[14] architecture, which determines whether or not overlapping image patches of size 70×70 are real. In addition, the recently proposed spectral normalization[15] is applied to the discriminator to improve the training stability. The frame discriminator D_{f} uses 2D convolution in order to pay attention to the spatial feature coherence of event frames. Although using 3D convolution in the generator can keep more spatiotemporal information, it will also make the image edges appear blurred. Therefore, an event sequence discriminator D_{s} is introduced, that is, using 3D convolution to improve the quality of the generated image. The event sequence discriminator D_{s} focuses on the time dependence and coherence of pixel changes.

The whole loss function contains four terms, which are L_1 loss, perceptual loss[16], style loss[17] and adversarial loss[18]. L_1 loss is often used in image inpainting and video inpainting, so we mainly introduce the latter three loss functions.

In order to address the blurry results caused by L_1 loss, we adopt the perceptual and style loss to preserve the image contents. The perceptual loss regularizes the generated target image I_{pred} to be closer to the ground truth I_{gt} in VGG[19] subspace. Its formulation is given as follows:

$$L_{\mathrm{perc}}=E[\Sigma_i 1/N_i\|\phi_i(I_{\mathrm{gt}})-\phi_i(I_{\mathrm{pred}})\|_1] \ , \tag{3}$$

where is the activation map of the ith layer of a pretrained network. For our work, ϕ_i corresponds to activation maps from layers relu1_1, relu2_1, relu3_1, relu4_1 and relu5_1 of the VGG-19 network. For event inputs, we modify the number of input channels in the first layer, and randomly initialize the weights of this layer.

For better recovery of detailed textures, different from perceptual loss, style loss is firstly applied an auto-correlation (Gram matrix) to the features. Style loss measures the differences between covariances of the activation maps which is also used VGG to compute. Given feature maps of sizes $C_i \times H_i \times W_i$, style loss is computed by:

$$L_{\mathrm{style}}=E[\|\boldsymbol{G}_j^{\phi}(I_{\mathrm{gt}})-\ \boldsymbol{G}_j^{\phi}(I_{\mathrm{pred}})\|_1], \tag{4}$$

where $\boldsymbol{G}_j^{\phi}$ is a $C_j \times C_j$ Gram matrix constructed from activation maps ϕ_j.

Adversarial loss simply constrains the generated events to follow the same distribution as the real ones and avoids directly constraining the network to memorize the trajectories seen at training time. As shown below, we formulate the adversarial loss as:

$$L_{\mathrm{g}}=E_{I_{\mathrm{in}}\sim P_{\mathrm{data}}(I_{\mathrm{in}})}\left[\log D_{\mathrm{f,s}}\left(G\left(I_{\mathrm{in}}\right)\right)\right], \tag{5}$$

$$L_{D_{\mathrm{f,s}}}=E_{I_{\mathrm{gt}}\sim P_{\mathrm{data}}(I_{\mathrm{gt}})}\left[\log D_{\mathrm{f,s}}\left(I_{\mathrm{gt}}\right)\right]+ E_{I_{\mathrm{in}}\sim P_{\mathrm{data}}(I_{\mathrm{in}})}\left[\log\left(1-D_{\mathrm{f,s}}\left(G\left(I_{\mathrm{in}}\right)\right)\right)\right], \tag{6}$$

where $\log D_{\mathrm{f,s}}(I_{\mathrm{gt}})$ is the probability of being a real frame and $\log(1-D_{\mathrm{f,s}}(\boldsymbol{G}(I_{\mathrm{in}})))$ is the probability to be a synthesized frame, f stands for frame discriminator and s represents the sequence discriminator. Our overall loss function is:

$$L_{\mathrm{G}}=\lambda_1 L_1+\lambda_{\mathrm{p}} L_{\mathrm{perc}}+\lambda_{\mathrm{s}} L_{\mathrm{style}}+\lambda_{\mathrm{g}} L_{\mathrm{g}}, \tag{7}$$

$$L_D=\lambda_{D_{\mathrm{s}}} L_{D_{\mathrm{s}}}+\lambda_{D_{\mathrm{f}}} L_{D_{\mathrm{f}}} \ . \tag{8}$$

where λ_1, λ_{p}, λ_{s}, λ_{g} , $\lambda_{D\mathrm{s}}$ and $\lambda_{D\mathrm{f}}$ are the weights for L_1 loss, perceptual loss, style loss, generator loss, sequence discriminator loss and frame discriminator loss respectively.

Eight images are used as an input sequence to train the network, and batch size is 3. The model is optimized using Adam optimizer[20] with β_1=0.1 and β_2=0.9. Generator $\boldsymbol{G}$ is trained with learning rate 10^{-4} until the losses plateau. Discriminators are trained with a learning rate one tenth of the generator's. As for the hyperparameters of the loss function, we choose λ_1=1, $\lambda_{\mathrm{g}}=\lambda_{\mathrm{p}}$=0.1, λ_{s}=250, $\lambda_{D\mathrm{s}}=\lambda_{D\mathrm{f}}$=0.5. 100 000 iterations were carried out on MVSEC and 50 000 iterations were carried out on DHP19. The experiments were all made on one NVIDIA TITAN RTX GPU.

Since there is currently no dataset for event inpainting, this paper constructs based on the existing event dataset and simulates two different types of mask. Meanwhile, the mask in this paper is only used to generate damaged event sequences, and will not participate in other processes of the algorithm as a priori.

MVSEC is a collection of data designed for the development of novel 3D perception algorithms for event-based cameras. In this paper, only outdoor car scenes recorded during the day are used. From there, we choose the first 80% as the training set and the following as the test (20%) set. All frames are cropped and resized to 256×256.

DHP19 is the first human pose dataset with data collected from DVS event cameras. We only use session2 and session4 which totally have 199 video sequences from the view of camera 2. The way of split the dataset is the same as DHP19. Meanwhile, the data has no background, we cropped and resized all frames to 256×256.

The Moving-Bar generates a vertical white bar crossing the sequence, very roughly mimicking a fence or any similar obstacle and only used for MVSEC.

The mask above is an artificial occlusion image, which cannot simulate the real situation. However, real large-scale completion datasets based on event cameras

is difficult to obtain, so we consider another case. In the DHP19 dataset, we integrate 100 events into one frame as input, make 7 500 events into one frame as ground truth, and use 3D joints label to get alignment one after another. When the 100 events frames are aligned with the 7 500 events frame, the time resolution is the same as that of the 7 500 events frame, and vice versa.

We are the first to propose the task of event image sequences inpainting thus there are no directly comparable methods. So, we compare our method to the state-of-the-art free-form intensity video inpainting LGTSM[5].

We verified the effectiveness of our Event-inpainting Network with two datasets. Fig.3 shows a sample image sequence generated by our model. Our model can generate event-realistic results with a fraction of image structures. It is also important that the inpainting images show minimal blurriness. In the DHP19 dataset, our model can fill out complete contours of extremely sparse event image sequences. In the MVSEC dataset, our method shows the ability to complete the occlusion area of the event image sequences.

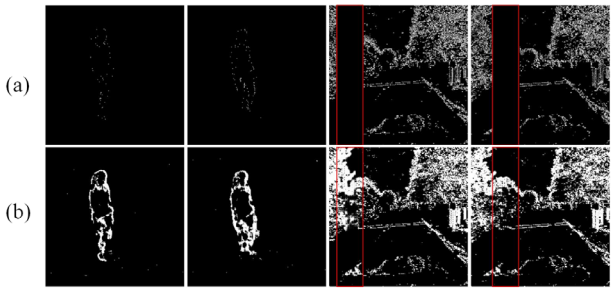

Fig.3 Event-inpainting on DHP19 and MVSEC: (a) Input image sequences; (b) Generated image sequences (The moving bar is marked by a red box.)

Fig.4 compares images generated by our method with the state-of-the-art technique LGTSM. In the MVSEC dataset, as shown in the yellow box, our method retains more complete structure. However, there are more holes in the dense area, which we suspect is caused by dilated convolutions. In the DHP19 dataset, LGTSM can produce smoother edges, but it is more distorted. The image generated by our method is more realistic, closer to the ground truth and the noise distribution is also learned.

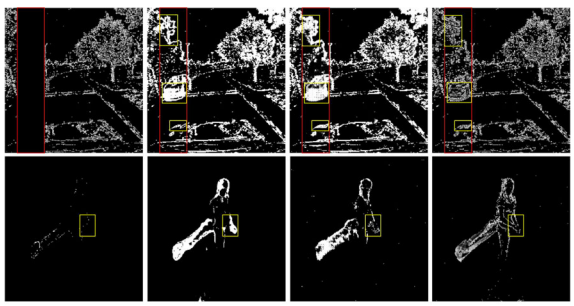

Fig.4 Visual comparison: (a) Input images; (b) Inpainting images by LGTSM; (c) Inpainting images by our method; (d) Ground truth

We use the following metrics to measure the quality of results: the peak signal-to-noise ratio (*PSNR*) in dB (logarithmic scale), the structural similarity index (*SSIM*) with a window size of 11, and the perceptual similarity (*LPIPS*) as a metric to evaluate the similarity of the high-level features in two images (lower the value, more the similarity). Use mean per joint position error (*MPJPE*) for downstream task.

The quantitative results over MVSEC and DHP19 datasets are reported in Tab.1. We could see that our model is on par with the state-of-the-art method LGTSM in terms of *SSIM* and *PSNR* with only 6.5% of parameters. As the number of LGTSM network layers (17 layers) is deeper than the number of our network layers (8 layers), LGTSM is better than our method in terms of LPIPS. Although increasing the convolution dimension will increase the amount of calculation, by reducing the number of layers and channels, the amount of calculation can be balanced with the fast response characteristics of the event camera. In the end, the inference speed can reach 500 fps.

Tab.1 Comparison of metric values on two datasets

Dataset	Mask	Metric	LGTSM	Ours
MVSEC	Moving bar	*SSIM*	**0.797 1**	0.763 9
		PSNR	**36.79**	35.87
		LPIPS	**0.179 3**	0.210 8
DHP19	Free-form	*SSIM*	**0.863 3**	0.843 7
		PSNR	**41.86**	41.40
		LPIPS	**0.281 1**	0.324 1
Number of parameters (Generator)			12M	**781k**
Inference fps			75	**500**

At the same time, we performed the downstream task human pose estimation on the image after DHP19 completion. The dataset is obtained by aligning the 100 events frame with the 7 500 events frame. And the training method is similar to DHP19. We use the hourglass[21] model to train the dataset with 25 epochs. We have trained and tested two kinds of time resolution data. Qualitative results for human pose estimation can be found in Fig.5. We could see that the result of human pose estimation before inpainting tends to random and irregular. After completion, the prediction is more stable. The quantitative results of human pose estimation after event inpainting are given in Tab.2, which further shows that our model can improve the performance of downstream tasks after event inpainting and is expected to be applied to high frame rate scenes.

In order to quantify the importance of the sequence discriminator, we have also carried out additional ex-

periments. In experiments, we remove the sequence component from our model, that is, we removed the temporal sequence discriminator D_s and replaced 3D generator with 2D generator. Tab.3 shows that our model improves temporal quality by reducing LPIPS. In Fig.6, comparing (a) and (b), the image will be blurred after removing D_s; comparing (b) and (c), the 3D generator retains more details than 2D generator. Therefore, our method can effectively improve time consistency and event image sequences inpainting quality.

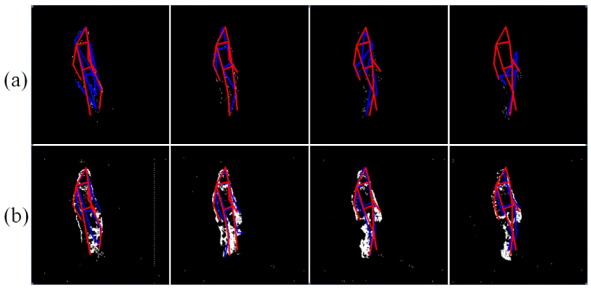

Fig.5 Human pose estimation on DHP19: (a) 100 events per frame; (b) 100 events per frame after inpainting (Predictions are in blue, and ground truth are in red.)

Tab.2 Comparisons of human pose estimation errors before and after event inpainting (pixel)

Parameter	Event100	Inpainting event100
100 events per frame align to 7 500 (30 fps)	6.250	**4.524**
Event100 (2 000 fps)	5.369	**4.054**
Average	5.809	**4.289**

Tab.3 Ablation study on DHP19

Metric	2D generator+D_f	3D generator+D_f	3D generator+D_f+D_s
SSIM	0.844 6	0.837 2	**0.847 3**
PSNR	40.73	39.69	**41.40**
LPIPS	0.346 1	0.368 2	**0.324 1**

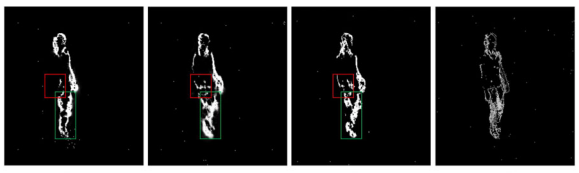

Fig.6 Visual comparison with ablation study on the DHP19: (a) 2D generator+D_f; (b) 3D generator+D_f; (c) 3D generator+D_f+D_s; (d) ground truth

In this paper, we proposed the Event-inpainting, which is the first learning-based fast event image sequences inpainting model. Event-inpainting comprises of a light-weight 3D generator, following a frame discriminator and a temporal sequence discriminator. We demonstrate that the temporal sequence discriminator plays an important role in enhancing the temporal consistency and video quality. Our method can inpaint event image sequences with occluded areas and low spatial resolution. In addition, our model only uses 6.5% of LGTSM parameters, and is 6 times faster than LGTSM. Therefore, it is possible to apply our method to high frame rate motion.

References

[1] Gallego G., Rebecq H. and Scaramuzza D., A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and optical Flow Estimation, IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[2] Enrico C., Gemma T., Christopher A.E., Sophie S., Federico C., Luca L., Kynan E. and Tobi D., DHP19: Dynamic Vision Sensor 3D Human Pose Dataset, IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[3] Cedric S., Nick B. and Robert M., Continuous-Time Intensity Estimation Using Event Cameras, Asian Conference on Computer Vision, 308 (2018).

[4] Kamyar N., Eric Ng, Tony J., Faisal Z. Qureshi and Mehran E., EventSR: From Asynchronous Events to Image Reconstruction, Restoration, and Super-Resolution via End-to-End Adversarial Learning, IEEE Conference on Computer Vision and Pattern Recognition, 2020.

[5] Jing-yuan Li, Ning Wang, Le-fei Zhang, Bo Du and Da-cheng Tao, Recurrent Feature Reasoning for Image Inpainting, IEEE Conference on Computer Vision and Pattern Recognition, 7760 (2020).

[6] Yi W., Ying-cong C., Xin T. and Jia-ya J., VCNet: A Robust Approach to Blind Image Inpainting, IEEE Conference on Computer Vision and Pattern Recognition, 2020.

[7] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee and Winston Hsu, Free-Form Image Inpainting with Gated Convolution, International Conference on Computer Vision, 4470 (2019).

[8] Chen Gao, Ayush Saraf, Jia-Bin Huang and Johannes Kopf, Flow-edge Guided Video Completion, European Conference on Computer Vision, 2020.

[9] Alex Zihao Zhu, Dinesh T., Tolga O., Bernd P., Vijay K. and Kostas D., IEEE Robotics and Automation Letters **3**, 2032 (2018).

[10] Patrick L., Christoph P. and Tobi D., J Solid-State Circuits **43**, 566 (2008).

[11] Pini S., Borghi G. and Vezzani Ro., International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications **4**, 37 (2020).

[12] Alex Zihao Zhu, Liang-zhe Yuan, Kenneth C. and Kostas D., EV-FlowNet: Self-Supervised Optical Flow

Estimation for Event-based Cameras, Robotics: Science and Systems, 2018.

[13] Kai-ming He, Xiang-yu Zhang, Shao-qing Ren and Jian Sun, Deep Residual Learning for Image Recognition, IEEE Conference on Computer Vision and Pattern Recognition, 770 (2016).

[14] Dmitry U, Andera V and Victor L, Improved Texture Networks: Maximizing Quality and Diversity in Feed-Forward Stylization and Texture Synthesis, IEEE Conference on Computer Vision and Pattern Recognition, 4105 (2017).

[15] Phillip I., Jun-Yan Zhu, Ting-hui Zhou and Alexei A. E., Image-to-Image Translation with Conditional Adversarial Networks, IEEE Conference on Computer Vision and Pattern Recognition, 5967 (2017).

[16] Miyato T., Kataoka T., Koyama M. and Yoshida Y., Spectral Normalization for Generative Adversarial Networks, International Conference on Learning Representations, 2018.

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio, Generative Adversarial Nets, Annual Conference on Neural Information Processing Systems, 2672 (2014).

[18] Johnson J, Alahi A and Fei-Fei L., Perceptual Losses for Real-Time Style Transfer and Super-Resolution, European Conference on Computer Vision, 694 (2016).

[19] Leon A.G., Alexander S.E and Matthias B., Image Style Transfer Using Convolutional Neural Networks, IEEE Conference on Computer Vision and Pattern Recognition, 2414 (2016).

[20] Kingma DP and Ba J., Adam: A Method for Stochastic Optimization, International Conference on Learning Representations, 2015.

[21] Newell A., Yang K. and Deng J., Stacked Hourglass Networks for Human Pose Estimation, European Conference on Computer Vision, 483 (2016).