Video Inpainting by Frame Alignment with Deformable Convolution

Yusuke HARA†a, Nonmember, Xueting WANG†, and Toshihiko YAMASAKI†, Members

SUMMARY Video inpainting is a task of filling missing regions in videos. In this task, it is important to efficiently use information from other frames and generate plausible results with sufficient temporal consistency. In this paper, we present a video inpainting method jointly using affine transformation and deformable convolutions for frame alignment. The former is responsible for frame-scale rough alignment and the latter performs pixel-level fine alignment. Our model does not depend on 3D convolutions, which limits the temporal window, or troublesome flow estimation. The proposed method achieves improved object removal results and better PSNR and SSIM values compared with previous learning-based methods.

Key words: video inpainting, deformable convolution, deep learning, computer vision

1. Introduction

Video inpainting is a task of filling missing regions in videos with visually consistent and plausible content in the context of visible areas. It is useful in various video editing tasks such as object removal, caption removal, damaged video restoration, or retargeting. What makes it more difficult than single image inpainting is that we need to properly borrow visible patches in other frames and generate temporally coherent results. Thus, it is important to efficiently manage information of multiple frames.

Although a traditional optimization-based method, Huang et al. [1], is still the state-of-the-art, many deep learning-based approaches [2]–[8] have been recently proposed. They often employ spatio-temporal networks (e.g., 3D convolution) [2], [3], [7], or use 2D convolutions with explicit frame alignment processes such as flow warping [4], [8] or affine transformation [5]. They take advantage of generalization ability of neural networks and run much faster than optimization-based methods. Their results, however, often have artifacts due to a limited temporal window, flow estimation failure, or insufficient alignment. Moreover, flow-based methods require additional supervision of flow fields whose accurate ground truth is difficult to prepare.

In this paper, we propose a video inpainting method by jointly using affine transformation and deformable convolutions for frame alignment. For video inpainting, affine transformation was first used in Copy-and-Paste Networks (CP-Net) [5]. This operation requires only six parameters and they can be easily estimated by Convolutional Neural Networks (CNN) with low computational cost. Thus, a large time span of frames can be utilized to fill the missing regions. Compared to using 3D convolutions or flow-based warping, it can efficiently deal with larger or more slowly moving masked regions. However, affine transformation cannot perform pixel-level alignment, and local mismatches often occur resulting in blurred or striped output. To address this issue, we introduce deformable convolutions [9], [10] to further refine the alignment.

Deformable convolution is a technique that can adaptively change its receptive field by learnable offsets. Recent studies [11], [12] of video super-resolution have shown that it can perform feature-level and robust frame alignment without explicit flow estimation. Inspired by them, we apply this approach into video inpainting. For video inpainting, however, the missing regions may interfere with the deformable convolution alignment. To avoid this problem, we provisionally fill the alignment target frame before the deformable convolution alignment. This operation is essential to make a clear alignment target so that reference frames are properly integrated.

We conducted experiments to evaluate our method on the DAVIS dataset [13], [14]. We show that the feature-level alignment of deformable convolutions significantly improves reconstruction ability compared with other learning-based methods [5]–[7]. We also evaluated our method qualitatively on object removal tasks. User studies show that our network produces more plausible results than other learning-based methods [5]–[7]. Furthermore, our method achieves comparable results to the optimization-based state-of-the-art method, Huang et al. [1], while ours runs considerably faster.

Our contributions are as follows.

- We organically combine affine transformation and deformable convolution to efficiently and accurately collect information from a wide range of frames. Our model does not depend on difficult flow estimation and can be trained without any additional supervision.
- We propose a technique to provisionally fill the alignment target frame so that the deformable convolution alignment works well even in the missing regions.
- Our method achieved superior performance in quantitative and qualitative evaluations to existing learning-based approaches. Moreover, our results are visually at a near level to the optimization-based state-of-the-art.
method while saving computational time. Our source code will be available at https://github.com/y60/dconv-vi.

2. Related Work

2.1 Image Inpainting

Image inpainting is a task of filling missing regions in single images. Traditional methods for image inpainting [15], [16] complete a missing region by borrowing usable patches outside the missing region and fitting them. Such methods, however, often require large computational cost for searching or optimization processes. Moreover, the patch-based strategy cannot generate novel views.

Recently, CNN-based methods [17]–[23] are proposed to achieve high-quality generative inpainting. Large training data allows more natural completion than simply finding patches from outside the missing area. In [19], for impressive restoration, a generative adversarial network (GAN) [24] was introduced. Reference [17] extended the idea and used two discriminators to create globally and locally natural results. Reference [22] proposed a coarse-to-fine network architecture and a contextual attention module for searching relevant patches. References [18] and [23] pointed out that convolutions without distinction of visible and missing regions result in artifacts. To solve this problem, they employed the partial convolution and the gated convolution, respectively. In [20], to prevent object boundaries from being blurred, they introduced an image segmentation process before the completion. For the same problem, Ref. [21] used an edge prediction model.

Although these various techniques made great achievements on image inpainting, simply applying them to each video frame produces temporally inconsistent results, which are fatal to naturalness. Furthermore, they cannot utilize visible patches in other frames. It is necessary to efficiently manage temporal information for video inpainting.

2.2 Video Inpainting

Methods of video inpainting can be divided into optimization-based ones and learning-based ones.

In optimization-based methods [1], [25]–[27], patches from visible regions are borrowed and fit to the missing regions. Though they produce significantly plausible results in some cases, they cannot perform in real-time because they often include heavy optimization processes in searching and fitting patches. For example, Huang et al. [1] optimized both optical-flow and color in the masked area by minimizing an objective function that encourages spatio-temporal consistency and flow smoothness.

On the other hand, learning-based methods run faster by taking advantage of high generalization ability of neural networks. The first deep learning-based method [7] used 3D convolutions in the deep feature space to collect information from other frames and generate temporally coherent results. Another method [4] performed flow-based warping and a single 3D convolution layer to compute temporally aggregated feature of neighbor frames. Following them, Ref. [2] pointed out that training only with the L1 loss causes blurred results and they proposed the first video inpainting method in the framework of GANs [24]. They adversarially trained an encoder-decoder network and a discriminator, both of which are composed of 3D convolutions, to tackle with complex scenes or masks. After [2], to avoid high computational cost of 3D convolutions, Ref. [3] proposed the learnable temporally shifting module to efficiently collect content from neighboring frames. However, these methods [2]–[4], [7] have limitations on the temporal window size and they cannot or do not efficiently consider distant frames.

To avoid the difficult flow estimation and efficiently obtain a large temporal window, Onion-Peel Networks (OPNet) [6] employs a spatio-temporal attention module in the deep feature space, which can search the input video for applicable patches in a non-local manner. It has great capability to fill never visible areas by borrowing patches from visible regions. However, its results tend to lack sufficient temporal consistency.

To address the same issue, Copy-and-Paste Networks (CPNet) [5] uses affine transformation to align other frames with the completion target frame. An affine matrix between two frames can be estimated easily and fast by CNNs. Thus, it can collect more reference frames to fill each frame than 3D convolutions or flow-based warping. However, blurred or striped results are often caused by alignment insufficiency due to absence of explicit pixel-level alignment.

Our method uses deformable convolutions for pixel-level alignment and affine transformation for rough alignment. This combination makes it possible to efficiently and accurately gather information covering distant frames. Furthermore, our network is independent of flow estimation, which has difficulty in complex scenes and often requires additional supervision.

In parallel with our research, other methods [28]–[30] have been proposed. Zeng et al. [29] employed non-local attention modules like OPNet [6]. Liu et al. [30] used only deformable convolutions for frame alignment. Gao et al. [28] proposed a flow-based approach with improved flow completion and pixel propagation.

2.3 Video Super-Resolution

The demand for efficient management of multi-frame information and temporal consistency of outputs is common
Fig. 1  Overview of the proposed method. We extend Copy-and-Paste Networks (CPNet) [5] and introduce an additional alignment module using deformable convolutions (Red).

to video super-resolution. Frame-Recurrent Video Super-Resolution (FRVSR) [31] uses flow-based warping. By recurrently taking the current frame and the warped previous frame, it can generate temporally consistent results while saving the computational cost. Another video super-resolution method [32] adopts a GAN framework, where a discriminator distinguishes generated frame sequences from real ones.

Temporally Deformable Alignment Network (TDAN) [11] is the first method employing a frame alignment module using deformable convolutions. Deformable convolution is a convolution that can dynamically transform the receptive field by learnable parameters. In TDAN [11], to align neighboring frames (reference frames) with the target frame, each of their feature is processed by deformable convolutions whose parameters are determined considering the target feature. This enables feature-level alignment which is more flexible and adaptive than explicit flow-based warping. Following TDAN [11], Ref. [12] proposed a PCD alignment module consisting of a coarse-to-fine structure of deformable convolutions to improve the alignment performance.

Inspired by these methods [11], [12], we introduce the alignment approach by deformable convolutions into video inpainting for adaptive, pixel-level, and feature-level frame alignment.

3. Proposed Method

3.1 Overview

The inpainting framework of our method is based on CPNet [5], which uses only affine transformation for frame alignment, and we add a DConv (deformable convolution) module for further alignment (Fig. 1). Specifically, the additional module is based on the Pyramid, Cascading, and Deformable convolutions (PCD) proposed for video super-resolution by [12]. We first briefly review CPNet [5] and then discuss how to combine the deformable convolution module.

3.2 Copy-and-Paste Networks

For ease of explanation, we formulate video inpainting as a task to process $T$ consecutive video frames ($X = \{X_1, \cdots, X_T\}$) with binary masks ($M = \{M_1, \cdots, M_T\}$) indicating the missing pixels in each frame into completed frames ($\hat{Y} = \{\hat{Y}_1, \cdots, \hat{Y}_T\}$). CPNet [5] consists of an alignment network and a completion network.

3.2.1 Alignment Network

In the alignment network, to complete each frame $X_t$ (target frame), many other frames (reference frames) are aligned to $X_t$ by affine transformation. The affine matrix between the target frame and each reference frame is estimated by a CNN. It is trained with a self-supervised L1 loss considering visible areas only:

$$V_t(x, y) = 1 - M_t(x, y),$$

$$L_{align} = \sum_{t=1}^{T} \sum_{r} \|V_{t-r} \odot V_t \odot (X_t - X_{t-r})\|_1,$$

where $V_t$ is the visibility map of $X_t$, and $X_{t-r}$ denotes the $r$th reference frame aligned with the target frame $X_t$. $V_{t-r}$ is the aligned visibility map of the $r$th reference frame $V_t$ in the same way. $\odot$ is pixel-wise multiplication.

3.2.2 Completion Network

Following the alignment network, the completion network fills the target frame using the aligned frames in an encoder-decoder manner. First, the shared encoder extracts the feature $F_t$ from the target frame and $F_{t-r}$ from each aligned frame. To aggregate the information of the reference frames, the weighted sum $F_{t-r}$ of $F_{t-r}$’s is computed (Eq. (3)). The
 pixel-wise weight \( C_{r}^{\text{match}} \) is based on \( \theta_{r \rightarrow t} \) (Eq. (4)), which is a global similarity between \( F_{r \rightarrow t} \) and \( F_t \) calculated in the visible areas, where \( V_t(x, y) = V_{r \rightarrow t}(x, y) = 1 \) (Eq. (5)).

\[
F_{r \rightarrow t} = \sum_r C_r^{\text{match}} \odot F_{r \rightarrow t}, \quad (3)
\]

\[
C_{r}^{\text{match}}(x, y) = \begin{cases} \exp \left( \theta_{r \rightarrow t}(x, y) \right) / \sum_r \exp \left( \theta_{r \rightarrow t}(x, y) \right) & V_{r \rightarrow t}(x, y) = 1, \\ 0 & \text{otherwise,} \end{cases} \quad (4)
\]

\[
\theta_{r \rightarrow t} = \frac{(V_r \odot F_t) \cdot (V_{r \rightarrow t} \odot F_{r \rightarrow t})}{V_r \cdot V_{r \rightarrow t}}. \quad (5)
\]

\( \cdot \) denotes an inner product of two flattened vectors. The aggregated mask \( C_{r}^{\text{mask}}(x, y) \) indicating the areas that are never visible in the reference frames is also computed as:

\[
C_{r}^{\text{mask}}(x, y) = 1 - \sum_r C_r^{\text{match}}(x, y). \quad (6)
\]

Finally, the decoder takes \( F_t, F_{r \rightarrow t}, \) and \( C_{r}^{\text{mask}} \) to generate the completed frame \( \hat{Y}_t \). Repeating this process, the video is completed in temporal order and reverse order using already completed frames to maintain temporal consistency. The final output is the weighted sum of the two completion results.

In our experiments, however, the actual values of \( \theta_{r \rightarrow t} \)’s do not converge in a proper scale to eliminate useless reference frames. It results in significantly blurred outputs. To counter this problem, we introduce the following normalization:

\[
\theta'_{r \rightarrow t} = \frac{M_0}{\max(\theta_{r \rightarrow t})} \theta_{r \rightarrow t}, \quad (7)
\]

where all \( \theta_{r \rightarrow t} \)’s are multiplied by the same value so that the maximum of \( \theta'_{r \rightarrow t} \) is normalized to a constant value \( M_0 \). We use \( \theta'_{r \rightarrow t} \) instead of \( \theta_{r \rightarrow t} \) in Eq. (4). We can change \( M_0 \) freely in accordance with the number of input frames. Our model is trained with \( M_0 = 12 \) and \( T = 5 \), while \( M_0 \) is set to 100 in completing DAVIS [13], [14] videos whose number of frames, \( T \), is from 50 to 100.

3.3 DConv Alignment for Video Inpainting

The additional module including PCD [12] is introduced for further alignment in the completion network. As described in the following, it is not just a combination of the two approaches, but their organic combination is carefully designed.

3.3.1 Provisional Filling

For video inpainting tasks, large missing regions may interfere with PCD [12]. To address this issue, we provisionally fill the target feature \( F_t \) to prepare an explicit alignment target \( F_{t}^{\text{pf}} \) and then align each reference feature \( F_{r \rightarrow t} \) to \( F_{t}^{\text{pf}} \) through the PCD. To fill each pixel in the missing region of the target feature, the value of the nearest visible reference frame is borrowed. In other words, reference features are taken from the nearest and copied to fill the target feature as shown in Fig. 2. Our method does not specify which frame at the same distance (e.g., (1) or (2) in Fig. 2) should be treated as the nearer.

3.3.2 PCD Alignment

After the provisional filling, the PCD module takes each reference feature \( F_{r \rightarrow t} \) with the target feature \( F_{t}^{\text{pf}} \) and performs pixel-level alignment using deformable convolutions [9], [10] (Eq. (8)).

\[
F_{r \rightarrow t} = \text{PCD} \left( F_{r \rightarrow t}, F_{t}^{\text{pf}} \right). \quad (8)
\]

Deformable convolution [9], [10] can adaptively change its receptive field by learnable offsets. Given that \( F_{\text{in}} \) and \( F_{\text{out}} \) denote the input and output features, a \( 3 \times 3 \) deformable convolution is formulated as:

\[
F_{\text{out}}(x) = \sum_{k=1}^{K} w_k F_{\text{in}}(x + p_k + \Delta p_k) \Delta m_k, \quad (9)
\]

where \( w_k \) and \( p_k \in \{-1, -1\}, \{-1, 0\}, \ldots, \{1, 1\} \) are the weight and the fixed offset at the \( k \)-th sampling location. \( \Delta p_k \) and \( \Delta m_k \) are the additional offsets and the modulation computed by some learnable normal convolutions. PCD [12] is designed to process a reference feature through deformable convolutions whose deformation parameters are predicted by using a target feature in addition to the reference feature. In our method, the PCD module processes each \( F_{r \rightarrow t} \) with \( F_{t}^{\text{pf}} \) and outputs \( F_{r \rightarrow t}^{\text{pf}} \).

The PCD module is followed by the aggregation stage, that is, \( F_{r \rightarrow t} \) in Eq. (3) is replaced with \( F_{r \rightarrow t}^{\text{pf}} \). Note that the aggregation weight \( C_{r}^{\text{match}} \) of each frame is still calculated from \( F_t \) and \( F_{r \rightarrow t} \) (as derived in Eqs. (4) and (5)) because the target feature \( F_t \) is never processed by the PCD.

3.4 Loss Functions

The entire model is trained in an end-to-end manner according to CPNet [5] with a loss function including the following L1 loss, \( L_{\text{l1}} \), between the prediction \( \hat{Y}_t \) and the ground truth \( Y_t \):
They are computed separately in the missing regions and the others. In the missing regions, the losses are further separated considering visibility in the reference frames using \( C_{\text{mask}} \). In contrast to CPNet [5], we assign a larger weight to losses in the missing regions than to those in the non-missing regions, to focus on collecting and aligning information properly from reference frames.

The perceptual loss \( L_{\text{precp}} \) [33], the style loss \( L_{\text{sty}} \) [33], and the total variation loss \( L_{\text{tv}} \) [34] are included in the same way as CPNet [5] to generate visually plausible results and prevent from the checkerboard artifacts.

The learning rate is set to 10^{-5} after about 260,000 iterations.

We set the batch size to 8, that is, 8 sequences are input for the object segmentation masks, however, makes it difficult to train the objective function on the data used for training is summarized in the first row of Table 1.

| Frame Mask | Training | Mask |
|------------|----------|------|
| YouTube-VOS [37] | Irregular Mask Dataset [18] |
| Places2 [38] | Rectangular |

Table 1: The data used for training and evaluation

Contains random and complex shapes. We apply random rotation, cropping, and dilation while limiting the missing region from 5\% to 20\% of the frame size. The second type consists of square masks. The size and position of each square is randomly determined. The third type is also composed of square masks. The beginning and last masks are randomly generated, and the rest of masks are created by linear interpolation.

The frame type and the mask type of each sequence are randomly chosen with equal probability. The combination of the data used for training is summarized in the first row of Table 1.

4.2 Training Details

We set the batch size to 8, that is, 8 sequences are input for each iteration. The parameters are updated with the Adam optimizer [39] in the condition of \( \beta_1 = 0.9 \) and \( \beta_2 = 0.999 \). The learning rate is set to \( 10^{-4} \) at the beginning and reduced to \( 10^{-5} \) after about 260,000 iterations.

4.3 Evaluation Methods

To confirm the effectiveness of the proposed method, we compare it with existing video inpainting methods by both quantitative and qualitative evaluation.

4.3.1 Comparison Targets

We first compared our method with the existing state-of-the-art methods including the optimization-based method (Huang et al. [1]) and several latest learning-based approaches OPNet [6], CPNet [5], and LGTSM [3], which are introduced and discussed in Sect. 2.2. Moreover, we conducted detailed ablation study comparing the proposed network with modified versions including the network without PCD, and the network whose number of parameters are equalized.

4.3.2 Quantitative Evaluation

We conducted quantitative evaluation using the widely used evaluation metrics PSNR and SSIM on the DAVIS dataset [13], [14]. DAVIS [13], [14] consists of 90 videos with object masks for video object segmentation task. We selected 40 videos from DAVIS [13], [14]. Directly using the object segmentation masks, however, makes it difficult...
to treat the original frames as the ground truth. Thus, we shuffled the pairs of frames and object masks among the 40 videos, which is the same way as some previous methods [5], [6] did. Using the shuffled pairs, we calculated PSNR and SSIM values between the original frames and the completion results.

### 4.3.3 Qualitative Evaluation

We conducted user studies to evaluate the object removal performance of our method for qualitative evaluation. We used DAVIS [13], [14] whose pixel-wise object annotations can be used as masks for object removal. Furthermore, Huang et al. [1] provides object masks covering object shadow for some DAVIS [13], [14] videos. If available, we used them. We used the same 40 videos as for the quantitative evaluation. Note that they were selected so that most of the videos the recent studies [4]–[6] used for qualitative evaluation are contained. The combination of the data used for qualitative evaluation is summarized in the third row of Table 1.

To compare our method with each existing method, the object removal results on each video were shown to 20 workers on a cloudsourcing service. They were asked to select the better one from the two results of the proposed method and each previous method, or answer “indistinguishable” while referring to the original video. The 40 videos were separated into four groups and the workers were different among the groups.

### 5. Results

#### 5.1 Quantitative Results

The quantitative comparison results are shown in Table 2. We can observe that the proposed method achieved the highest performance on both PSNR and SSIM metrics, outperforming previous video inpainting methods [1], [3], [5], [6] with notable margins. The fact that our scores are higher than those of CPNet [5] shows that the combination of the deformable convolution module (PCD) can effectively improve the frame alignment and the reconstruction performance.

Table 3 shows the average execution time over the 40 object removal results. Our method is much faster than that of Huang et al. [1]. Our approach runs at a comparable speed with CPNet [5] and OPNet [6]. LGTSM [3] is considerably faster, but the quality of its results is not as good as the others (Sect. 5.2).

In order to further confirm the effect of PCD, we show the difference of the learning curves of 1) the proposed network, 2) the proposed network without PCD, and 3) the proposed network whose number of parameters is adjusted to the same as 2) in Fig. 3. The parameter size adjustment is done by reducing the channel numbers of the entire network. This comparison indicates that the alignment using deformable convolutions significantly improves the completion capability, and it is more powerful than simply enlarging the network size.

#### 5.2 Qualitative Results

For the qualitative evaluation, Fig. 4 shows the detailed voting distributions of the user studies of pairwise comparison between the proposed method and each previous method. Moreover, to summarize the results, the percentage of the workers who selected ours as the better one is calculated for each video. Table 4 shows the average of the percentages among all videos for each existing method [1], [3], [5], [6]. The workers who answered “indistinguishable” are ignored. From the table, we can find that more workers preferred our method than all previous learning-based methods [3], [5], [6]. The comparison with CPNet [5] especially indicates that the proposed alignment using deformable convolutions greatly helps to generate vi-
Fig. 4 The detailed results of the user studies for comparison with previous methods [1], [3], [5], [6]. Specifically, it shows the workers’ voting distribution in the pairwise comparison for each video. The green bar shows the number of workers who selected ours as the better one, the blue bar represents each previous method, and the orange bar represents the selection of “indistinguishable.”

Usually plausible completion results. Moreover, our method is not seriously worse than the optimization-based state-of-the-art method [1]. Considering that our method runs more than ten times faster than [1], there is a reasonable trade-off between time and quality.

The examples of the object removal results are shown in Fig. 5. Compared with other learning-based methods [3], [5], [6], the proposed method deals better with complex scenes including objects moving differently from backgrounds and objects with complicated textures. In the video...
Fig. 5 Qualitative comparison of existing video inpainting methods [1], [3], [5], [6] and ours on the DAVIS dataset [13], [14]. The individual results are zoomed for a better view of the important areas. The video titles are above the corresponding results.
Fig. 6  Failure cases. Ours often produces blurred results when it has to reconstruct areas that are invisible through all the frames.

swing, for example, our method can properly process pixels around the wooden frame moving independently from the background. In other examples, high reconstruction capability of ours can be observed in complex objects such as the grass (in swing), the letters “ASVZ” (in tennis), and the bars (in horsejump-low).

Figure 6 shows failure cases of our method. Ours often fails when it has to reconstruct areas that are invisible through all the frames. For goat, ours fails to recover the background hidden behind the goat through the entire video. This is because our model does not explicitly borrow patches from visible regions to fill such areas as OPNet [6] and Huang et al. [1] do. We also show bmx-bumps in Fig. 6 as a case where ours loses to CPNet [5] in the user study.

Ours sometimes fails when a missing region is crossing the border of the frame. Though its reason is not clarified, we suspect that the amount of such scenes in the training data is still insufficient in our implementation.

6. Conclusions

We have presented an approach of video inpainting combining affine transformation and deformable convolutions for frame alignment. We showed that the deformable convolution alignment can significantly improve completion performance without explicit flow estimation. In quantitative and qualitative experiments, our method outperformed existing deep learning-based methods and obtained comparable results to the optimization-based state-of-the-art approach.

References

[1] J.-B. Huang, S.B. Kang, N. Ahuja, and J. Kopf, “Temporally coherent completion of dynamic video,” ACM ToG, vol.35, no.6, pp.1–11, 2016.
[2] Y.-L. Chang, Z.Y. Liu, K.-Y. Lee, and W. Hsu, “Free-form video inpainting with 3D gated convolution and temporal patchgan,” ICCV, pp.9065–9074, 2019.
[3] Y.L. Chang, Z.Y. Liu, K.Y. Lee, and W. Hsu, “Learnable gated temporal shift module for deep video inpainting,” BMVC, 2019.
[4] D. Kim, S. Woo, J.-Y. Lee, and I.S. Kweon, “Deep video inpainting,” CVPR, pp.5785–5794, 2019.
[5] S. Lee, S.W. Oh, D. Won, and S.J. Kim, “Copy-and-paste networks for deep video inpainting,” ICCV, pp.4412–4420, 2019.
[6] S.W. Oh, S. Lee, J.-Y. Lee, and S.J. Kim, “Onion-peel networks for deep video completion,” ICCV, pp.4402–4411, 2019.
[7] C. Wang, H. Huang, X. Han, and J. Wang, “Video inpainting by jointly learning temporal structure and spatial details,” AAAI, vol.33, no.1, pp.5232–5239, 2019.
[8] R. Xu, X. Li, B. Zhou, and C.C. Loy, “Deep flow-guided video inpainting,” CVPR, pp.3718–3727, 2019.
[9] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, “Deformable convolutional networks,” ICCV, pp.764–773, 2017.
[10] X. Zhu, H. Hu, S. Lin, and J. Dai, “Deformable convnets v2: More deformable, better results,” CVPR, pp.9300–9308, 2019.
[11] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, “TDAN: Temporally-deformable alignment network for video super-resolution,” CVPR, pp.3360–3369, 2020.
[12] X. Wang, K.C.K. Chan, K. Yu, C. Dong, and C.C. Loy, “EDVR: Video restoration with enhanced deformable convolutional networks,” CVPRW, pp.1954–1963, 2019.
[13] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, “A benchmark dataset and evaluation methodology for video object segmentation,” CVPR, pp.724–732, 2016.
[14] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, “The 2017 davis challenge on video object segmentation,” arXiv preprint arXiv:1704.00675, 2017.
[15] C. Barnes, E. Shechtman, A. Finkelstein, and D.B. Goldman, “Patchmatch: A randomized correspondence algorithm for structural image editing,” ACM ToG, vol.28, no.3, pp.1–11, 2009.
[16] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, “Summarizing visual data using bidirectional similarity,” CVPR, pp.1–8, 2008.
[17] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” ACM ToG, vol.36, no.4, pp.1–14, 2017.
[18] G. Liu, F.A. Reda, K.J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions,”
ECCV, pp.89–105, 2018.

[19] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A.A. Efros, “Context encoders: Feature learning by inpainting,” CVPR, pp.2536–2544, 2016.

[20] Y. Song, C. Yang, Y. Shen, P. Wang, Q. Huang, and C.C.J. Kuo, “Spg-net: Segmentation prediction and guidance network for image inpainting,” BMVC, 2018.

[21] W. Xiong, J. Yu, Z. Lin, J. Yang, X. Lu, C. Barnes, and J. Luo, “Foreground-aware image inpainting,” CVPR, pp.5833–5841, 2019.

[22] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T.S. Huang, “Generative image inpainting with contextual attention,” CVPR, pp.5505–5514, 2018.

[23] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T.S. Huang, “Freeform image inpainting with gated convolution,” ICCV, pp.4471–4480, 2019.

[24] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” NeurIPS, vol.2, pp.2672–2680, 2014.

[25] Y. Wexler, E. Shechtman, and M. Irani, “Space-time completion of video,” TPAMI, vol.29, no.3, pp.463–476, 2007.

[26] M. Granados, J. Tompkin, K. Kim, O. Grau, J. Kautz, and C. Theobalt, “How not to be seen — object removal from videos of crowded scenes,” Computer Graphics Forum, Wiley Online Library, vol.31, no.2pt1, pp.219–228, 2012.

[27] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, “Video inpainting of complex scenes,” SIAM Journal on Imaging Sciences, vol.7, no.4, pp.1993–2019, 2014.

[28] C. Gao, A. Saraf, J.-B. Huang, and J. Kopf, “Flow-edge guided video completion,” ECCV, pp.713–729, 2020.

[29] Y. Zeng, J. Fu, and H. Chao, “Learning joint spatial-temporal transformations for video inpainting,” ECCV, pp.528–543, 2020.

[30] R. Liu, Z. Weng, Y. Zhu, and B. Li, “Temporal adaptive alignment network for deep video inpainting,” IJCAI, pp.927–933, 2020.

[31] M.S.M. Sajjadi, R. Vemulapalli, and M. Brown, “Frame-recurrent video super-resolution,” CVPR, pp.6626–6634, 2018.

[32] Y. Xie, E. Franz, M. Chu, and N. Thuerey, “tempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow,” ACM ToG, vol.37, no.4, pp.1–15, 2018.

[33] L.A. Gatys, A.S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” CVPR, pp.2414–2423, 2016.

[34] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” ECCV, pp.694–711, 2016.

[35] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” CVPR, pp.248–255, 2009.

[37] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang, “YouTube-VOS: Sequence-to-sequence video object segmentation,” ECCV, pp.603–619, 2018.

[38] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” TPAMI, vol.40, no.6, pp.1452–1464, 2018.

[39] D.P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” ICLR, 2015.

Yusuke Hara received his B.E. degree from the University of Tokyo in 2020. He is a graduate student with the Department of Information and Communication Engineering in the University of Tokyo. He is a member of IEEE and ITE.

Xueting Wang is currently a Project Assistant Professor in the Department of Information and Communication Engineering, the University of Tokyo, Japan. She received the M.S. degree and Ph.D. degree in Information Science from Nagoya University, in 2015 and 2018, respectively. She has received the best paper award at ISM2014. Her research interests focus on multimedia processing, video editing and summarization, social media analysis, and recommendation. She has been the primary organizer of MMAsia2019 Tutorial on “Multimedia Attractiveness Computing”, and will be Local Arrangement Co-Chair for MIPR2021. She has served as a TPC member of IJCAI, ACM, MMAsia. She is a member of ACM, IEEE, IEICE, IPSJ, and ITE.

Toshihiko Yamasaki received his B.S., M.S., and Ph.D. degrees from the University of Tokyo. He is currently an Associate Professor at Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo. He was a JSPS Fellow for Research Abroad and a visiting scientist at Cornell University from Feb. 2011 to Feb. 2013. His current research interests include attractiveness computing based on multimedia big data analysis and fundamental problems in pattern recognition and machine learning. He is a member of IEEE, ACM, AAAI, IEICE, IPSJ, and ITE.