Cartoon Animation Outpainting with Region-guided Motion Inference

Huisi Wu, *IEEE Senior Member,* Hao Meng, Chengze Li, Xueting Liu, *IEEE Senior Member,* Zhenkun Wen, *IEEE Member,* and Tong-Yee Lee, *IEEE Senior Member*

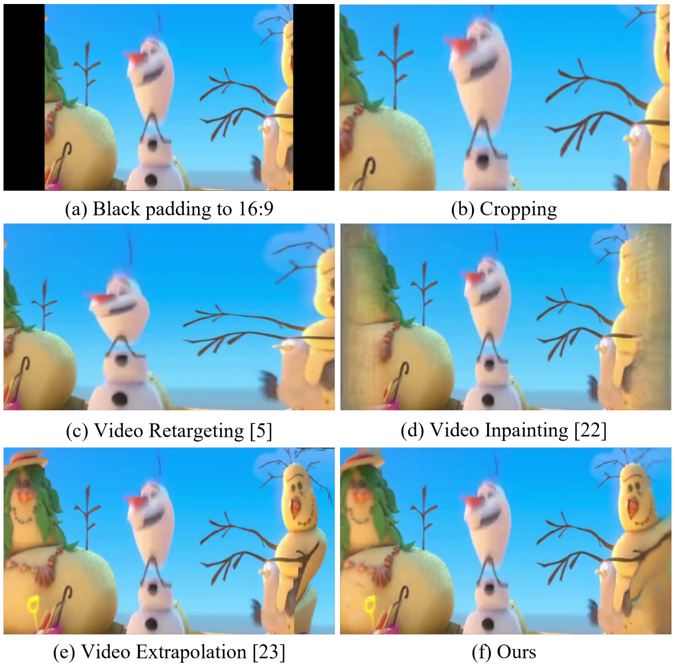
Abstract—Cartoon animation video is a popular visual entertainment form worldwide, however many classic animations were produced in a 4:3 aspect ratio that is incompatible with modern widescreen displays. Existing methods like cropping lead to information loss while retargeting causes distortion. Animation companies still rely on manual labor to renovate classic cartoon animations, which is tedious and labor-intensive, but can yield higher-quality videos. Conventional extrapolation or inpainting methods tailored for natural videos struggle with cartoon animations due to the lack of textures in anime, which affects the motion estimation of the objects. In this paper, we propose a novel framework designed to automatically outpaint 4:3 anime to 16:9 via region-guided motion inference. Our core concept is to identify the motion correspondences between frames within a sequence in order to reconstruct missing pixels. Initially, we estimate optical flow guided by region information to address challenges posed by exaggerated movements and solid-color regions in cartoon animations. Subsequently, frames are stitched to produce a pre-filled guide frame, offering structural clues for the extension of optical flow maps. Finally, a voting and fusion scheme utilizes learned fusion weights to blend the aligned neighboring reference frames, resulting in the final outpainting frame. Extensive experiments confirm the superiority of our approach over existing methods.

Index Terms—Cartoon animations, video outpainting, optical flow, deep learning.

1 Introduction

Anime is a worldwide popular visual entertainment and art form with significant market demand. Though many new titles have been created, there are still a large amount of legacy and classic animations popular and enjoyed by audiences. However, the old cartoon animations were usually produced in 4:3 aspect ratio resolutions, which do not match the commonly used 16:9 aspect ratio or even wider screens at present. To display a classic 4:3 cartoon animation video on a wider screen, the animation is typically placed at the center of the screen, with two wide black stripes on the left and right sides, respectively(Fig. 1 (a)). This significantly harms the visual experiences of the audience.

To provide a visually pleasing experience, it is necessary to remove the two wide black stripes and display the cartoon animation in the full screen area. To do so, a straightforward approach is to crop and stretch a 16:9 rectangular area in the 4:3 animation. However, cropping may lead to evident information loss (Fig. 1 (b)). Retargeting the cartoon animation frames from 4:3 to 16:9 (Fig. 1 (c)) may lead to distortion of the content, even with the state-of-the-art retargeting methods [1]–[5]. Therefore, the animation companies still rely on manual labor to remake the classic anime videos, such as the movie of *Monkey King*, *Ronin Warriors*, and *Fullmetal Alchemist*. As the artists will preserve the original content and paint out the content on the two sides that originally do not exist, the quality of the output is significantly better than cropping or retargeting.



(a) Black padding to 16:9

(b) Cropping

(c) Video Retargeting [5]

(d) Video Inpainting [22]

(e) Video Extrapolation [23]

(f) Ours

Fig. 1. Various approaches to extend the field-of-view.

Nonetheless, manually outpainting a cartoon animation is extremely tedious and labor-intensive. An automatic

Huisi Wu (hswu@szu.edu.cn), Hao Meng, Xueting Liu and Zhenkun Wen are with the College of Computer Science and Software Engineering, Shenzhen University, 518060, Shenzhen, Guangdong, China.
Chengze Li is with the School of Computing and Information Sciences, Caritas Institute of Higher Education, Hong Kong, China.
Tong-Yee Lee is with the Department of Computer Science and Information Engineering, National Cheng-Kung University, Taiwan, R.O.C.

This work was supported partly by National Natural Science Foundation of China (No. 62002232 and No. 62273241), the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. UGC/FDS11/E03/22), and the National Science and Technology Council under nos. 111-2221-E-006-112-MY3, Republic of China (ROC), Taiwan.

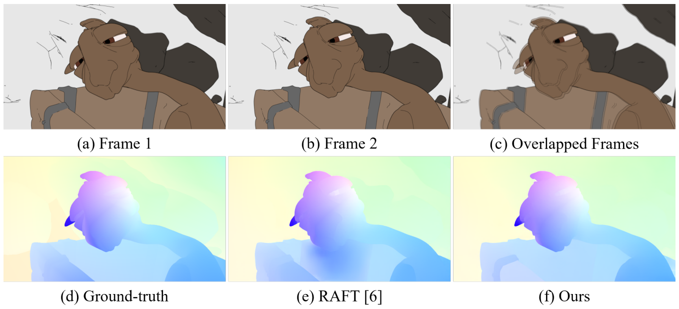


Fig. 2. Existing optical flow estimation methods on cartoon animations. The existing motion estimation methods tailored for natural videos generally cannot work well in cartoon animations.

cartoon animation outpainting method is highly in demand. However, such a task is not easy. Firstly, the outpainted cartoon animations frames should be temporally smooth and consistent for visual pleasantness. This suggests that the content to be outpainted at the two sides should be guided by not only the current frame, but the neighboring frames as well. Secondly, the movement of objects in cartoon animations may be exaggerated and not obey the physical laws, therefore, the existing motion estimation methods tailored for natural videos generally cannot work well in cartoon animations [6], [7] (Fig. 2). Finally, cartoon animations frequently use solid-color regions to depict objects, leading to a general fact of lack of texture, which further complicates the motion estimation and makes existing natural video tailored methods unsuccessful. As shown in Fig. 2, the existing methods are quite error-prone when handling solid-color regions which are quite common in cartoon animations.

In this paper, to resolve the mentioned problems, we propose a novel cartoon animation outpainting framework via a deep learning approach with motion inference. The key idea of our method is to find the motion correspondences of a cartoon animation frame against a long sequence of neighbor frames in the video so that we can have as much information as possible to reconstruct the missing pixels on the two sides of the cartoon animations. Then all frames are warped and aligned to this frame so that the information from different frames can be integrated for the outpainting. Therefore, our system is designed to have three stages. In the first stage, we propose to estimate the motion of the objects via optical flow estimation. Since the existing optical flow methods are error-prone due to the lack of textures in cartoon animations, we novelly propose to adopt the guidance of regions in optical flow estimation. With the guidance of regions, our method can well handle the optical flow in flat-color regions. In the second stage, we stitch the frames in sequence to produce a pre-filled guide frame. This guidance frame can provide structural clues to extend the field of view of the optical flow maps from 4:3 to 16:9. After the reconstructed 16:9 optical flows, in the third stage, we align all neighbor frames in the sequence to this frame to form a series of reference frames. Finally, the reference frames are blended with a novel voting and fusion scheme where the fusion weights are generated by a deep learning network with a stacked channel attention module. Extensive experiments have been conducted to validate the effectiveness of our method (Fig. 1 (f)). The main contributions of this paper are summarized as follows:

- We propose a novel cartoon animation outpainting framework based on region-guided motion inference. Remarkably, our proposed method is totally different from traditional retargeting methods to enlarge the input video.
- Due to the lack of texture, we proposed a novel cartoon animation tailored optical flow estimation method guided by regions.
- We propose the voting and fusion scheme to outpaint the missing pixels based on the estimated optical flow.

2 Related Work

In this section, we study the related work to our task. We mainly categorize them into three approaches and briefly describe them in this session.

2.1 Image and Video Retargeting

Image and video retargeting are the technologies used to adaptively change the aspect ratio of images and video content based on viewing devices [5], [8]–[10]. For image retargeting, it is required to preserve the structural features of the subject as much as possible during the change of aspect ratio. Seam carving [11] is one of the most widely used methods for image retargeting, but it is difficult to achieve consistent deformation across multiple frames of the video. Moreover, it is hard to find a good energy function to estimate the structure of cartoon animations. Cho et al. [12] proposed a CNN-based image retargeting method to determine the important part of the image and ensure the content is remaining in the output through a pre-trained classification module. Tan et al. [9] proposed to generate both narrow and enlarged results at the same time and gradually optimize by keeping the results consistent across multiple iterations. All three methods above are image-based without the capability to maintain temporal consistency for videos. To deal with video, Cho and Kang [13] proposed a foreground-aware video extrapolation method with dynamic sensing of the foreground to extend the video boundary. It reduced the unpleasant deformation during the retargeting process and maintained temporal consistency. Yan et al. [5] constructed a new energy function that considered both spatial and temporal constraints in video retargeting. Kim et al. [14] proposed a deep neural network model that can aggregate temporal features while maintaining temporal consistency. This model can be used for fast video inpainting and video retargeting tasks. Although video retargeting methods can change the aspect ratio of video contents with flexibility, they usually result in uneven content scaling [13]–[16]. In contrast, we aim to preserve the composition of the original cartoon animations within the 4:3 field-of-view.

2.2 Video Inpainting

Video inpainting aims to repair the missing areas in the video while maintaining temporal and spatial consistency [17]–[22]. Their inference of the image-level content is usually from the hole's surrounding pixels, while the temporal consistency is generally maintained through optical flow estimation and maintenance. For example, Xu et al. [19] used a coarse-to-fine strategy to refine optical flow in hole area, then used optical flow to guide the propagation of pixels between frames. Gao et al. [23] extracted and repaired the line information in the optical flow to generate a sharper motion boundary to contribute a higher quality result through non-local pixel propagation. Some methods used the attention mechanism to capture the frame correlations. For example, Onion-Peel Network [24] used pixel-level attention to fill holes in the video gradually. Lee et al. [21] used the attention between frames to align and fuse the content of multiple frames to obtain the content in the hole. STTN [22] introduced the transformer model to encode the temporal and spatial information of the video sequence to achieve the balance of performance and efficiency. Some methods formulated video inpainting as a constrained image generation problem and used GANs to generate the content of the missing regions. For example, Chang et al. [25] used temporal SN-PatchGAN [26] and temporal-shift modules to repair irregular-shaped masks. E2FGVI [27] proposed an end-to-end optical flow-based video inpainting method by applying optical flow warps to image features and embedding them in the network. FGT [28] proposed a new flow-guided Transformer method that used the motion differences of optical flow to guide attention retrieval and achieve high-fidelity video restoration with improved efficiency through window partitioning strategies and flow weighting modules. Although these methods have achieved good results in video inpainting, the solid-color region in cartoon animation videos often lead to the failure of optical flow estimation and the confusion of motion boundaries. It brings difficulties to these methods in cartoon animation video. This paper proposes an image enhancement method that can improve the performance of optical flow models trained on natural image data when applied to cartoon animation videos.

Additionally, there were a few works dedicated to image inpainting in the cartoon field, such as Seamless Manga Inpainting [29], which inpainted bubbles in comics by decomposing the comic and inpainting the layers separately before merging them. However, it is targeted for manga images, which are still very different from cartoon images. The open source project Anime Inpainting [30] used an edge connection [31] model retrained with a cartoon dataset to perform inpainting of anime characters. However, the Edge-Connection model was designed for natural image inpainting, so it was difficult to address the target area of cartoon animation outpainting, and it was also difficult to achieve temporal consistency requirements. Sketch-based Hairstyle Editing [32] focused on inpainting and editing the hairstyle of anime characters based on sketch lines, but it was difficult to cope with the various application scenarios of cartoon animation. These methods provide special optimization ideas for manga or cartoon images, but they are all challenging to use for the task of cartoon animation outpainting since all of these method were tailored for inpainting task instead of outpainting. For inpainting tasks, surrounding information could be utilized to fill in the pixels of a hole, while neighboring information of the sides is less useful. This is mainly because the information to support outpainting is much less than hole-filling. With the extension of the outpainting boundaries, these models will become less and less confident in recovering the contents.

2.3 Video Extrapolation and outpainting

The task of video extrapolation is to use a given video sequence to predict the peripheral information of the sequence that has not yet appeared in the video [33], [34], or outside the visible area of the video. To extend to a wider field of view with a given video, the method of [35]–[37] used blurred pixels to fill the surrounding area of the video content. This method was effective under the visual assumption that the viewer's sight would not deviate from the main screen. However, some studies have shown that viewers were more inclined to look around when viewing wide-field content [38]. Based on this consideration, the surrounding area of wide-field content should still require a filling of visually natural content. Lee et al. [21] used the 3D scene information recovered from the video to guide the sampling and blending of different frames regions. The surrounding area was filled with image blocks obtained from neighboring frames. Ma et al. [39] also used 3D scene information to expand the field of view while introducing the attention mechanism and uncertainty analysis to improve the accuracy of the results and enable the results to meet the requirements of downstream tasks. Dehan et al. [40] processed the foreground and background of the video separately, so they obtained good outpainting results for the background area. However, they still resulted in unbearable dissonance when dealing with foreground objects moving to the boundary. And the extrapolated optical flow based on the constant gradient also doesn't have fine-grain motion information.

This category of research is considered closest to our objective. However, these methods are tailored for natural videos and usually require a very precise estimation of camera motion and object locations. They do not allow apparent object motions across frames to minimize the mismatch in frame warping and blending. Unfortunately, in cartoons with hand-drawn content, we cannot obtain precise camera parameters or 3D reconstruction of objects, thus the existing solutions generally fail when applied on cartoons and animations.

3 Methodology

In this work, we propose to solve the challenge of field-of-view (FOV) outpainting in cartoon animations using deep learning. The key motivation is to find motion correspondences of frame $\mathcal{S}_i$ at time i against its neighbor frames in the sequence $\mathcal{S}$ and align these frames with time i by motion-based warping. Due to camera and object motion, those pixels that do not appear in $\mathcal{S}_i$ may present in the other frames in $\mathcal{S}$. Suppose that these pixels are warped outside the original 4:3 FOV of $\mathcal{S}_i$ after alignment; they shall contain additional information than $\mathcal{S}_i$. Thus, we can blend them to construct a much wider FOV of the input.

To collect these useful pixels scattered in other frames, some methods like [21], [22], [24], [41] use neural network

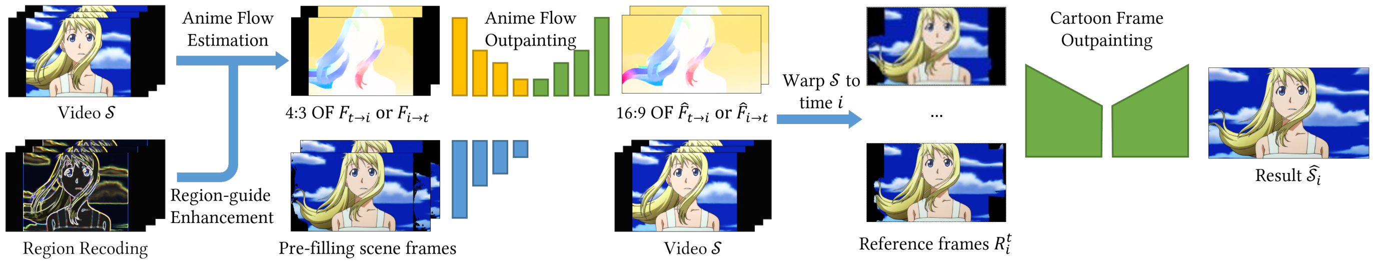


Fig. 3. Method Overview. Our method consists of three main stages: anime optical flow estimation, optical flow outpainting, and cartoon frame outpainting.

to perceive the correlation and differences between frames, and infer the content that can be filled in the target area. However, limited by the model capacity, these methods are difficult to deal with the situation where the clues are scattered in far away frames, like video outpainting. Therefore, during video outpainting, extracting pixel correspondence and filling target area are better separated as two stages, which can more flexibly use the correspondence between pixels and enable every single module to focus more on different tasks.

Under such principle, we have our framework with a three-stage design. In contrast to natural videos, the estimation of motion information in cartoon animations often encounters more errors. To address such challenges, our first stage is dedicated to cartoon animation optical flow extraction. Specifically, we propose a region-based method for cartoon animation optical flow extraction, which enhances the performance by employing different enhancements on lines and regions. In the second stage, we aim at the wide-FOV motion outpainting of 16:9 bi-directional optical flows from adjacent 4:3 frame pairs in $\mathcal{S}$. The FOV extension of optical flows helps to stabilize the frame alignment to $\mathcal{S}_i$, especially at the boundary locations, as the 4:3 optical flow cannot sufficiently estimate the motion of disappearing pixels. After that, in the third stage, we align neighbor frames of $\mathcal{S}_i$ in $\mathcal{S}$ to time i by SoftSplat warping [42]with the outpainted 16:9 optical flows and form a series of reference frames $\mathcal{R}_i$. Commonly, the reference frames farther away from time i shall contain more information on the target areas, but the motion may contain more errors to affect its alignment and vice versa. To blend the reference frames with precision, we propose our method of cartoon frames outpainting. The method compares all reference frames to $\mathcal{S}_i$ and estimates the reliability to use a certain reference frame to fill in the extra FOV. We convert the reliability into pixel-wise weights to blend all reference frames to complete the FOV outpainting process. The whole process is illustrated in Fig. 3 and we process all frames in $\mathcal{S}$ individually as input to complete the whole sequence.

We will introduce these three stages in detail as follows.

3.1 Anime Optical Flow Estimation

The ideal content of the area have been repaired should be consistent with that appeared in the video. To obtain information from neighboring frames, some existing methods [21], [22], [24], [41] proposed to feed multiple frames instead of a single frame to the network. However, without explicit analysis of the motion of the objects, these methods fail to reconstruct the large areas on the sides (Fig. 1(d)). Some video extrapolation and inpainting methods [23], [27], [28] use inter-frame optical flow to estimate the motion of the objects and use it as the guidance for pixel propagation (Fig. 1 (e)). So it is necessary to extract the motion between frames to correspond pixels from known to unknown. In previous studies, methods that perceive motion based on tools like attention are often difficult to achieve good results in outpainting, while methods based on optical flow can gradually perceive far-frame information through the motion information stored in optical flow. And avoids the limitation of the model capacity. Therefore, our method is based on optical flow for cartoon animation outpainting.

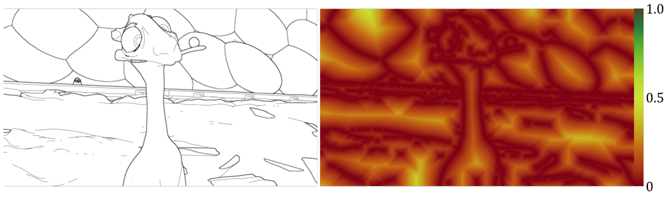


Fig. 4. Our proposed region encoding method. The left part displays the input sketch line image, the right part demonstrates the visualization of the generated region encoding.

In this stage, we focus on estimating the wide high-quality optical flow from cartoon animations.

In recent years, deep learning methods have been applied to motion estimation resulting in the development of several excellent optical flow estimation methods, such as FlowNet [43], RAFT [6], PWC-Net [44], and GMFlow [7]. However, all these methods are designed for the natural style videos, while they are not suitable for cartoon animations. Unlike natural videos, cartoon animations are typically produced by first drawing sketch lines as a structural framework and then filling colors in different regions. As a result, cartoon animations exhibit thicker structural lines and contain large areas of solid colors. The former leads to unclear motion boundaries, while the latter often results in erroneous feature matching.

In order to reduce the optical flow mismatch caused by the thick sketches and the lack of texture in animation frames, we first perform feature enhancement on animation frames by incorporating additional information from regions and lines.

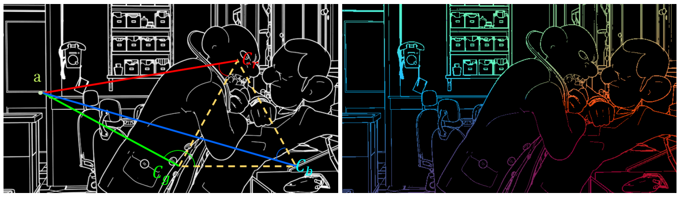


Fig. 5. Our proposed sketch line encoding method. The left image is the input sketch line, where C_r, C_g, and C_b separately represent the mass centers in the original image of the channel R, G and B, and a represents a point in the image that needs to be encoded. The encoding of a is composed of $\angle aC_rC_g, \angle aC_gC_b$, and $\angle aC_bC_r$. The right image shows the visualization of the generated line encoding.

Anime Region Enhancement: To improve the optical flow estimation due to the lack of textures, we first use the line drawings extracted from the cartoon animation frames to compute the segmentation boundaries for different regions in the input. Subsequently, we compute the distance transformation of the line drawings for each region as the texture coding for that target region. This texture encoding is solely dependent on the shape and size of the region, effectively marking unique pixels within the region with minimal computational requirements, as shown in Fig. 4. The distance transformation is computed as follows:

$$E_r\left(a\right) = \frac{||a - proj\left(a, L\right)||}{\sqrt{H * W}} + \epsilon \tag{1}$$

Where a denotes a point that inner the region, L denotes the point set of the image sketch line, $proj\left(a, L\right)$ denotes the projection of point a on the point set L, H and W denotes the height and width of the image respectively, ϵ denotes a base constant number.

Sketch Line Enhancement: The thick sketch lines in animation frames tend to cause pixel mismatching and tearing artifacts near the line boundaries when computing the feature correspondence. To mitigate this problem, we propose to re-encode line pixels within the animation frame. Based on our practical and preliminary experiments, we find out that there exists a motion-invariant coordinate system that is relatively static to the objects inside the scene, and can move along with almost the exact camera motion. This enables a semantically identical point on the boundary to be consistently encoded on different frames.

During the process of motion, the movement of the image mass center often approximates the scene's motion, thus the image mass center is a suitable anchor point for this coordinate system. Based on a single anchor point, the coordinates of each point on the image can be encoded using an angle value and a distance value (similar to polar coordinates). However, due to the direct correlation between the distance value and the image size, this encoding scheme fails to maintain consistency during common motion patterns such as image scaling. Therefore, we extend the number of anchor points and uniquely encode points on the sketch lines using a set of angle values to multiple anchor points, as shown in Fig. 5.

Specifically, we calculate the mass center separately for the different channels of the input frame in RGB color space. The mass centers form a triplet C_r, C_g, and C_b as anchor points to form a triangle. Denote the three sides of the triangle as C_rC_g, C_gC_b, and C_bC_r, for any point a in the image, the position of the point can be described by the angle between the line aC_r, aC_g, aC_b and C_rC_g, C_gC_b, C_bC_r. Here, we select $\angle aC_rC_g$, $\angle aC_gC_b$, and $\angle aC_bC_r$ to describe the position of a, as illustrated in Fig. 5.

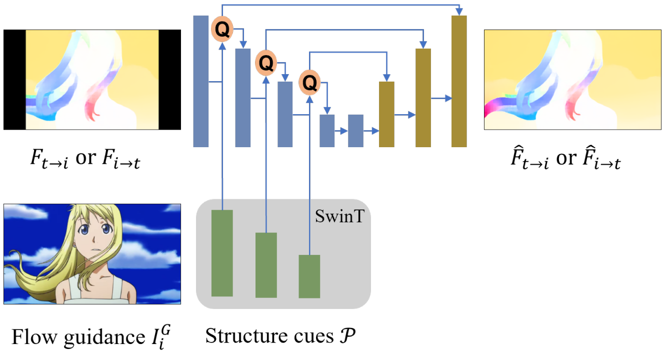

$F_{t\to i}$ or $F_{i\to t}$
Q
Q
Q
$\hat{F}_{t\to i}$ or $\hat{F}_{i\to t}$
SwinT
Flow guidance I_i^G
Structure cues $\mathcal{P}$

Fig. 6. Cartoon Animation Optical Flow Outpainting.

We employed the following formula to combine the results of texture enhancement and line encoding into the original image:

$$I_{line} = M_{line}\left(\alpha * I^{ori} + (1-\alpha) * E_l\left(I^{ori}\right)\right) \tag{2}$$

$$I_{region} = \widetilde{M}_{line}\left(\beta * I^{ori} + (1-\beta) * E_r\left(I^{ori}\right)\right) \tag{3}$$

$$I^{ehc} = I_{line} + I_{region} \tag{4}$$

where I^{ori} denotes the original input frame, M_{line} is the binary line mask to the input image, $E_l\left(I^{ori}\right)$ is the new line encoding, $E_r\left(I^{ori}\right)$ is the distance field generated from M_{line}, I_{line} denotes the line enhanced image, I_{region} denotes the region texture enhanced image, I^{ehc} is the final enhanced image, α and β are the blending constant number.

So, we can use above enhancement combined with the other optical flow model based on natural style data to obtain more accurate optical flow estimate results on animations. In our experiments, the animation optical flow $F_{1\to 2}$ is performed using the optical flow estimation model RAFT [6]:

$$F_{1\to 2} = \text{RAFT}\left(I_1^{ehc}, I_2^{ehc}\right) \tag{5}$$

3.2 Optical Flow Outpainting

After the bidirectional optical flow of the cartoon animation is obtained, content alignment can be performed frame by frame to expand the field of view for each frame. Unfortunately, during the propagation of pixels from distant frames to the current frame, the lack of optical flow vector guidance in the target area results in pixel loss, limiting the expansion of the field of view to a range close to the known regions.

To avoid losing motion guidance in the target area during pixel propagation, it is necessary to widen the extracted bidirectional narrow field-of-view optical flow F. Previous methods have often used Poisson filling (FGVC [23]) or coarse-to-fine approaches based on neighborhood pixel inference (DFCS [19]), but Poisson filling does not consider structural information in the image, and neighborhood pixel inference methods struggle to generate reliable structure for areas far from the boundaries.

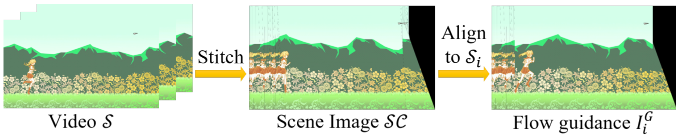


Fig. 7. Cartoon Animation stitching and align to $\mathcal{S}_i$

We think that structural cues in the input frame sequence $\mathcal{S}$ are useful to guide optical flow outpainting. Building on this idea, we complete optical flow outpainting through three steps: first, we stitch the cartoon frames together and pre-filling the target region of the current frame. Next, we employ a pre-trained image encoder to extract cues $\mathcal{P}$ from the pre-filled guidance frame I^G. Finally, a U-shaped network is employed to receive the cues $\mathcal{P}$ and predict optical flow information in the target region, as shown in Fig. 6. The details of the three parts are described as follows.

Frame Stitching and Pre-filling: Due to the rigid nature of objects during motion, the motion boundaries in optical flow often exhibit a high degree of consistency with the structural information in the image. Such structural cues can be extracted from the input frame sequence $\mathcal{S}$, which can effectively guide the restoration of optical flow F in the target region. However, the self-contained nature of video frames renders much of the structural information of the scene and object in $\mathcal{S}$ redundant. Therefore, we think it is critical to eliminate this redundancy and distill the crucial visual cues that contribute to reconstructing the optical flows beyond the original FOV. This is accomplished by initially conducting homography-based stitching for $\mathcal{S}$, allowing for a rough but efficient affine alignment of structures. This preliminary transformation serves as a beneficial initialization for subsequent optical flow reconstruction.

As shown in Fig. 7, our method first uses the middle frame $\mathcal{S}_{mid}$ ($mid = \frac{1}{2}length\left(\mathcal{S}\right)$) as the basis and stitches all cartoon frames together by homography based on feature matching. This can remove most redundant structural information from cartoon frames at a low cost, and generate a rough scene frame $\mathcal{SC}$. During the process of optical flow outpainting for a frame $\mathcal{S}_i$, the scene frame $\mathcal{SC}$ can be aligned to the current frame and components of a pre-filled guidance image I_i^G.

Structure Cues Extraction: Due to the unreasonable structures or seams in the pre-filled guide frames, it is necessary to predict the long-range relationship between reliable content in the center area and the pre-filled content on both sides. The CNN-based architecture is not suitable for this situation. Therefore, we fine-tune a pre-trained Swin-Transformer [45] encoder to extract structural cues. Specifically, we use a pre-trained Swin-Transformer [45] classification encoder to extract features at multiple scales ($1/2,\ 1/4$ and $1/8$) from the pre-filled guidance image I_a^G and compose them as a guidance cue pyramid $\mathcal{P}$ for optical flow outpainting.

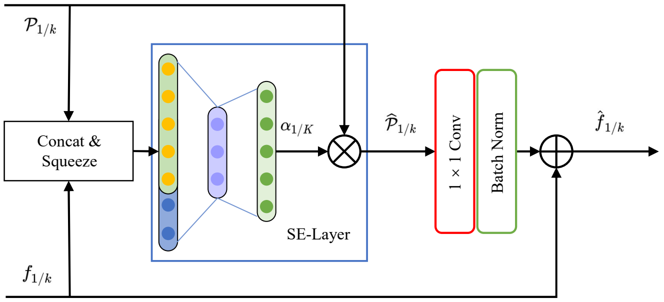


Fig. 8. The structural cue query block. The input is the optical flow feature $f_{1/k}$ and the structural guidance cue $\mathcal{P}_{1/k}$. The output is the queried and the fused feature $\hat{f}_{1/k}$. $\otimes$ represents multiplication with broadcasting. $\oplus$ represents element-wise addition.

Optical Flow Outpainting: Considering that the structural guidance cues $\mathcal{P}$ contains all object structural information, not all of these structural features will become motion boundaries in the flow vector set $F_{a\to b}$. Moreover, potential incorrect matching during the generation of the scene frame may lead to structural errors in the structural guidance cue. To better select the most relevant information in $\mathcal{P}$, we propose a structural cue query block to select and filter appropriate features for $F_{a\to b}$, as shown in Fig. 8.

To extend the FOV of optical flow, we first use the U-Net downscaling blocks to compute the multiscale feature $f_{1/k}, k \in (1/2, 1/4, 1/8)$ for the 4:3 optical flow $F_{a\to b}$. For each scale of feature f, the query block searches in the corresponding level of the structural cue $\mathcal{P}_{1/k}$ and finds the best features through a modified Squeeze-and-Excitement (SE) channel attention [46]. The SE block concatenates $\mathcal{P}_{1/k}$ and $f_{1/k}$ along the channel direction and squeezes each channel into a single point representation. Two dense layers further activate the squeezed vector to construct the excitement weight vector $\alpha_{1/k}$ for $\mathcal{P}_{1/k}$. The weight vector $\alpha_{1/k}$ works as the channel weight that amplifies the information in $\mathcal{P}_{1/k}$ that are directly related to the query $f_{1/k}$, to form the queried structural cue $\hat{\mathcal{P}}_{1/k}$. Finally, the queried structural cues are fused with $f_{1/k}$ as the U-Net feature at the scale of $1/k$. We illustrate the functionality of the feature query block in Fig. 8. The final output of the stage is refined and outpainted 16:9 version of the input optical flow. We denote it as $\hat{F}_{a\to b}$.

Training Objective: We apply the smooth L1 loss [47] as the training objective in this stage. Note that we only use this loss for bootstrapping purposes and will remove it during the joint training of the whole framework, as we do not obtain the ground truth 16:9 optical flow for our training animations. The loss is defined as:

$$L^{OF}_{\text{outpainting}} = \begin{cases} 0.5\left(\hat{F} - F^{GT}\right)^2 & \text{if } |\hat{F} - F^{GT}| < 1 \\ |\hat{F} - F^{GT}| - 0.5 & \text{otherwise} \end{cases} \tag{6}$$

where $\hat{F}$ and F^{GT} denotes the predicted optical flow and ground truth, respectively.

3.3 Cartoon Frames Outpainting

Once the 16:9 optical flow is obtained, utilizing information from distant frames, expansion can be achieved at a considerable scale. Based on the outpainted 16:9 optical flow, we align all neighboring frames $\mathcal{S}_i^t$ to time i into a list of reference frames $\mathcal{R}_i^t$, where t represents the time difference. The alignment reveals the out-of-FOV pixels of

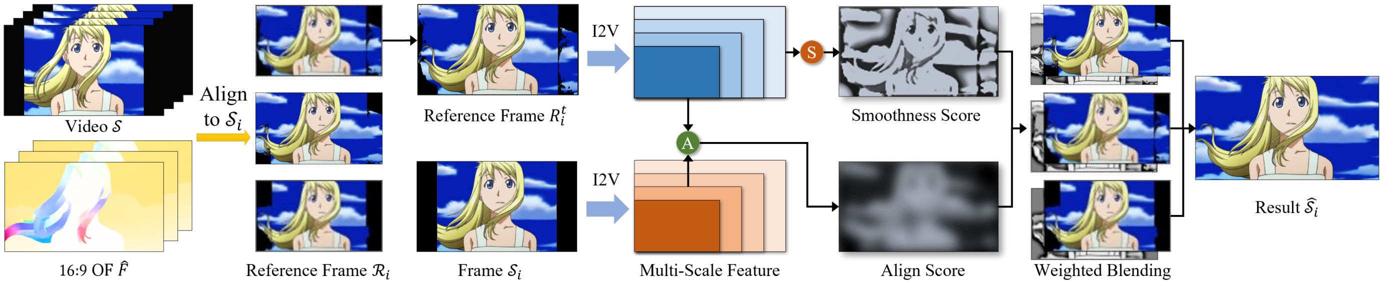


Fig. 9. Cartoon Frame Outpainting.

the original 4:3 input $\mathcal{S}_i$, which is the key source of the frame outpainting in the next step.

We apply the SoftSplat model [42] for frame warping based on optical flow. Note that we only compute the bidirectional optical flow of adjacent frames in the previous step. To warp the frame more than 1 time step to $\mathcal{S}_i^t$, we warp it multiple times instead of accumulating the optical flow for smoothness. We shall discuss the way to assemble the reference frames with $\mathcal{S}_i^t$ to reconstruct the full 16:9 FOV frame $\hat{\mathcal{S}}_i$ in the following part.

Due to the error in optical flow estimation and warping, the reference frames cannot be directly used for field-of-view outpainting. A direct blending of the reference frames may cause visual artifacts such as blurring and noticeable seams, especially when the reference frame is far away from the input. For the reference frames that are close to time i, their optical flow estimation with $\mathcal{S}_i$ are usually of good quality due to the limited motion. However, the small motion may not provide sufficient information to complete the whole 16:9 FOV for outpainting. Those reference frames far away from time i contain more information for frame outpainting but may suffer from imprecise motion estimation. Based on this finding, we propose the third stage of cartoon frame outpainting to assemble all reference frames and the input frame $\mathcal{S}_i$ for precise FOV outpainting. In this stage, we first learn two feature-level scores for each reference frame to validate its reliability for outpainting. We then use a deep neural network to convert the reliability score to the linear fusion weights of reference frames to blend the outpainted area. The pipeline of this stage is illustrated in Fig. 9.

Reference Frame Reliability Estimation: We compute two multiscale features to reflect the reference frame reliability: an alignment feature μ^a to estimate the alignment quality of the reference frame $\mathcal{R}_i^t$ to the input and a smoothness feature μ^s to estimate the visual quality and completeness of the reference frame. Both features are computed in the illustration2vec (I2V) [48] encoder domain. Compared to direct pixel-level difference estimation, the feature-level estimation enables a higher level of semantic understanding and comparison of image contents [27]. Additionally, the pixel-level comparison cannot handle incomplete image compositions, for example, for the regions outside the 4:3 FOV of the input where no ground truth alignments exist. In comparison, the feature-level estimation can still approximate the alignment because of the larger receptive field. Moreover, the I2V features are tailored to recognize illustrations and cartoons, which are more suitable for our task than the VGG model [49]. We extract features from the $relu2\_1$, $relu3\_2$ and $relu4\_2$ layers for further processing. These features are on three different scales, and we represent the feature extraction operator as $f^{I2V}_{1/k}(\cdot)$, where $k \in \{2, 4, 8\}$.

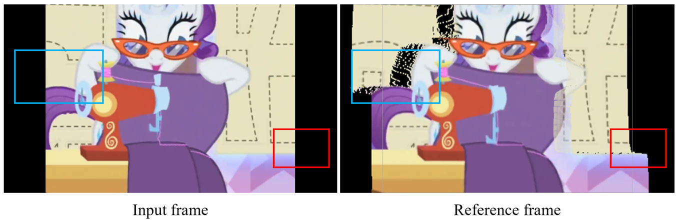


Fig. 10. Visualization of the estimation of the reference frame reliability. The blue box marks the ripping artifact which could be detected by the smoothness score. The red box marks a successful alignment around the 4:3 FOV boundary.

For each reference image $\mathcal{R}_i^t$, we compute the alignment feature μ^a based on a nonlinear mapping of the feature differences under the I2V encoding, as:

$$\mu^a_{1/k}\left(\mathcal{R}_i^t\right) = NL^a_{1/k}\left(f^{I2V}_{1/k}\left(\mathcal{R}_i^t\right) - f^{I2V}_{1/k}\left(\mathcal{S}_i\right)\right), \tag{7}$$

where $NL^a_{1/k}$ is a two-block composition of $Conv\text{-}BN\text{-}ReLU$ weights for scale $1/k$. We choose a higher level of features $k = 4$ and 8 in the I2V encoder for a larger receptive field for alignment estimation.

Additionally, due to the error in the previous stage of optical flow outpainting, some pixels may be distorted or ripped off after warping, causing discontinued local image neighborhood (e.g., the blue box in Fig. 10). We compute the smoothness score μ^s by comparing the difference between the pixel and the mean of its local window:

$$\mu^s_{1/k}\left(\mathcal{R}_i^t\right) = NL^s_{1/k}\left(f^{I2V}_{1/k}\left(\mathcal{R}_i^t\right) - Avg_{3\times3}\left(f^{I2V}_{1/k}\left(\mathcal{R}_i^t\right)\right)\right) \tag{8}$$

where $Avg_{x\times y}$ is the average pooling kernel sized $x \times y$ and $NL^s_{1/k}$ is another set of nonlinear mapping layers. We choose the lower level of the I2V features by setting $k = 2$, because the smoothness estimations are more fine-grained. Especially, we add a pixel-level smoothness feature μ^s_1 as the lowest level of μ^s:

$$\mu^s_1\left(\mathcal{R}_i^t\right) = NL^s_1(\mathcal{R}_i^t - Avg_{3\times3}(\mathcal{R}_i^t)), \tag{9}$$

We set all NL^a and NL^s output the same number of channels as 16.

Reference Frame Voting and Fusion: To convert the multichannel feature-based score μ^s and μ^a back to the linear blending weight for the reference frame $\mathcal{R}_i^t$, we first integrate the two scores into a single score c for each feature scale:

$$c_{1/k}\left(\mathcal{R}_i^t\right) = NL^c_{1/k}\left(\mu^s_{1/k}\left(\mathcal{R}_i^t\right), \mu^c_{1/k}\left(\mathcal{R}_i^t\right)\right), \tag{10}$$

where the NL^c is a 1x1 convolution block with 16 output channels. In some certain feature level k, there could be only the μ^s or the μ^a feature existing. In that case, we simply halve the input shape of the NL^c layer for that scale and keep the other settings remained.

With the multiscale reliability feature $c_{1/k}$ obtained, we first upsample it to the original image scale, then fuse all scales by linear sum. Then we compare the score c across all reference frames $\mathcal{R}_i$ to compose the weight $w\left(\mathcal{R}_i^t\right)$ for each specific reference frame $\mathcal{R}_i^t$. Suppose that we have n reference frames, we will have a total of 16 × n channels to represent the reliability of all reference frames. We first concatenate them along the channel direction and apply multiple stacked *ChannelAttn-Conv-ReLU* blocks, or, in short, the CCR blocks. The CCR blocks help to emphasize the differences in reliability between frames for all image locations with the channel attention mechanism [46]. We apply a total number of CCR blocks, and the output dimensions are $16\times, 32\times, 32\times$, and $1\times$ of n. Finally, we obtain n different scores $w\left(\mathcal{R}_i^t\right)$ for the reference frames by replacing the activation function of the last block with Softmax, which normalizes these weights. In this stage, we only perform the fusion outside the 4:3 FOV, as the pixels inside the 4:3 FOV have ground truths and are not meant to be changed.

Under rare conditions, the fusion weights may not fully cover the whole 16:9 frame size because the motion in the neighbor frames of $\mathcal{S}_i$ is too subtle to provide essential information, or the motion is too extreme to be well aligned. Under such circumstances, we can perform image inpainting methods such as [50] to complete the unfilled regions that could be computed by estimating a mask $\hat{M}_i$ to represent the filling status of a pixel:

$$\hat{M}_i = 1 \text{ if } \prod_{n=-t}^{t} (1-m_i^n)(1-\hat{w}_i^n) = 1, \tag{11}$$

where m_i^n denotes the valid warped pixels of the reference image $\mathcal{R}_i^t$ and the multiplication is the hadamard product.

Training Objective: After the bootstrapping of the second stage, we jointly train the whole framework with the image-level MSE loss as the only and the ultimate supervision:

$$L^{frame}_{\text{outpainting}} = \mathbb{E}_{\hat{M}_i}\left(\hat{\mathcal{S}}_i - \mathcal{S}_i^{GT}\right)^2/3, \tag{12}$$

where $\hat{\mathcal{S}}_i$ is the fused output frame and $\mathcal{S}_i^{GT}$ is the ground truth 16:9 FOV frame. We compute this loss over the valid area in $\hat{M}_i$.

3.4 Temporal Consistency Processing

After processing frames one by one, the field of view (FOV) of the cartoon animation is expanded, and the structural consistency of each frame is maintained by the motion information. However, due to potential inconsistencies in the color of non-local frames in the input and errors introduced by single-frame processing, there may be some flickering in the restored region. We use a simplified network from blind filter [51] to eliminate this inconsistency and apply a loss of temporal consistency [51] to constrain the training process.

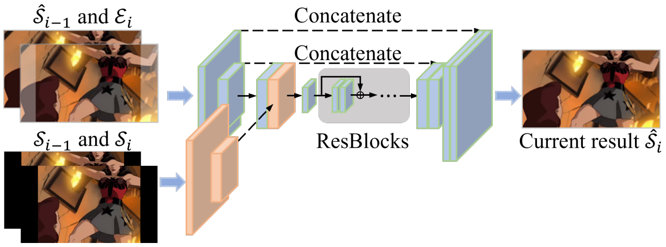

$\hat{\mathcal{S}}_{i-1}$ and $\mathcal{E}_i$
Concatenate
Concatenate
$\mathcal{S}_{i-1}$ and $\mathcal{S}_i$
ResBlocks
Current result $\hat{\mathcal{S}}_i$

Fig. 11. Temporal consistency processing to remove the flickering artifact.

Specifically, we extract and smooth the temporal information between two adjacent frames. Denote the unsmoothed and time-smoothed results by $\mathcal{E}$ and $\hat{\mathcal{S}}$, respectively. The $\left(\hat{\mathcal{S}}_{i-1}, \mathcal{E}_i\right)$ and the input frame pair $\left(\mathcal{S}_i, \mathcal{S}_{i-1}\right)$ comprise the input of this model. These two image pairs individually go through two convolutional layers to extract image features and temporal differences. Parameters on two paths do not share weights. Next, we use a convolutional layer to blend the features of the two branches and then pass five consecutive residual blocks for processing. Finally, the temporally smoothed output frame $\hat{\mathcal{S}}_i$ is obtained through two convolutional layers with upsampling. To maintain the outpainted structure information after processing, skip-connections are added between the layers shown in Fig. 11.

The temporal smooth loss [51] we used is:

$$L_{\text{TS}} = \sum_{i=2}^{SL} \left(\lambda_t * L_{\text{t}} + \lambda_p * L_{\text{p}}\right) \tag{13}$$

where i denotes the time stamp on the input sequence $\mathcal{S}$, SL is the length of the sequence, λ_t and λ_p are the weights for the temporal loss L_{t} and the perceptual loss of the content L_{p}, respectively. Specifically, $\lambda_t = 80$ and $\lambda_p = 1$. This combination effectively balances the reduction of temporal flickering and the minimization of perceptual distance. A lower λ_t/λ_p ratio causes the network's optimization to be predominantly driven by perceptual loss, which may induce temporal flickering in the reconstructed regions. On the other hand, increasing the λ_t/λ_p ratio tends to result in excessive blurring of the output videos, thus increasing the perceptual distance of the processed videos. This has been demonstrated in paper [51] dealing with temporal consistency.

We compute the temporal loss as the warping error between the output frames:

$$L_{\text{t}} = M_{i\Rightarrow i-1}\left(\hat{\mathcal{S}}_i - f\left(\hat{\mathcal{S}}_{i-1}\right)\right)^2 \tag{14}$$

$$M_{i\Rightarrow i-1} = \exp(-\alpha\|\mathcal{S}_i^{GT} - f\left(\mathcal{S}_{i-1}^{GT}\right)\|) \tag{15}$$

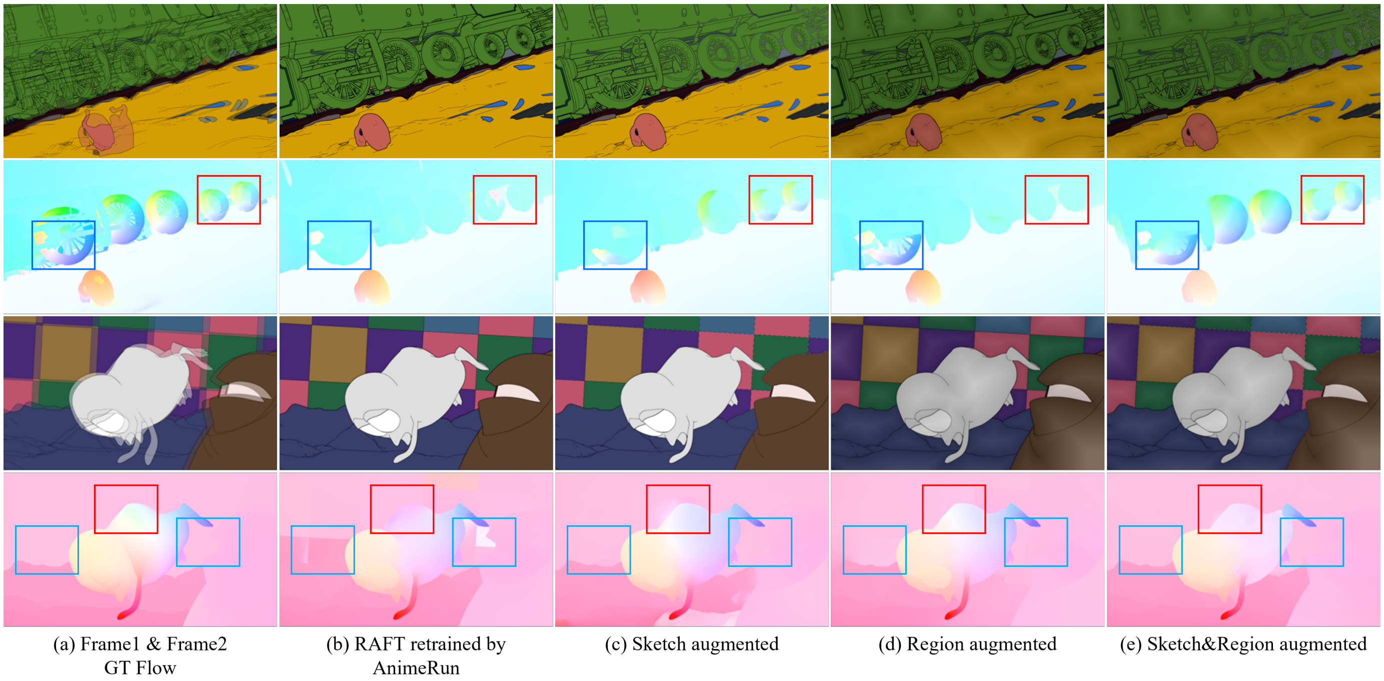


Fig. 12. Ablation Study of anime augmentation method. Improving input images with line and region-based methods can enhance the ability of optical flow models to process anime images in various ways. And the best results are achieved when both enhancement methods are combined, as demonstrated in the red and blue box areas of the image.

where $f\left(\hat{\mathcal{S}}_{i-1}\right)$ is the frame $\hat{\mathcal{S}}_{i-1}$ warped by optical flow $F_{i\Rightarrow i-1}$, and $M_{i\Rightarrow i-1}$ is the occlusion mask calculated from the warping error between the ground truth frames $\mathcal{S}_i^{GT}$ and the warped ground truth frame $f\left(\mathcal{S}_{i-1}^{GT}\right)$. The optical flow $F_{i\Rightarrow i-1}$ is the backward flow between $\mathcal{S}_{i-1}^{GT}$ and $\mathcal{S}_i^{GT}$. We use bilinear sampling to warp frames and empirically set $\alpha = 50$ (with pixel range between $[0, 1]$)

Since the style of cartoon animation is different from natural style images, we use an image encoder [48] dedicated to the classification of cartoon illustrations to calculate the perceptual loss L_{p}.

$$L_{\mathrm{p}} = \sum_{l} \left\| \phi_l\left(\hat{\mathcal{S}}_i\right) - \phi_l\left(\mathcal{S}_t^{GT}\right) \right\|_1 \tag{16}$$

where l denotes the scale level in the I2V encoder, symbol $\phi_l(\cdot)$ denotes the feature activation at the l-th layer of the I2V encoder, $\hat{\mathcal{S}}_i$ is the temporally smoothed result and $\mathcal{S}_i^{GT}$ is the ground truth frame.

4 Experiments

4.1 Experiment Details

Experiment Platform: Our experimental platform has an Intel i7-7700K CPU, 32GB DDR4 memory, and NVIDIA Geforce RTX 3090 GPU with 24GB memory.

Dataset: We collect our training data from the internet. The dataset contains more than 1000 cartoon animation clips of more than 40 different titles of animations. Cartoon animations have various origins. They come from different countries and are created in different eras. More importantly, we carefully choose the animation clips to cover a vast diversity of motion types, including but not limited to camera zooming, panning, yaw and pitch translation, subjects running, dancing, flying, and many other visual effects. In our experiments, the frame size is downscaled to 480 × 270, while we can still process larger frames if necessary. To simulate 4:3 FOV, we apply intra-frame zero-padding at the boundaries. The training and test set are separated with a 7:3 split. Given the fact that the essence of our approach is to leverage information from neighboring frames to assist in outpainting the target frame, achieving high-quality results usually necessitates a balanced level of motion, neither static nor overly drastic. When the motion falls within this optimal range, our dataset size can ensure a stable outpainted result.

Data Augmentation: To improve the model's robustness, we conducted several data augmentations during training. These included random horizontal and vertical image flipping, sequence order reversal, and randomly selected outpainting ratios ranging from 1.3 to 2.0.

Training Details: We implemented the whole framework in PyTorch 1.8.0 with CUDA 11.0. The batch size during training has been set to 16. We compute a total number of n reference frames for voting and fusion for each input frame. We employ the SGD optimizer throughout the training process. The second stage was bootstrapped for 300 epochs, and the learning rate was reduced from $3e$-2 to $2e$-5 with an exponential decay strategy. Then we train the complete framework for 200 epochs, and the learning rate is gradually reduced $1e$-2 to $1e$-7 with the same exponential decay.

4.2 Ablation Study

4.2.1 Animation Frame Enhancement

To verify the effectiveness of the proposed frame enhancement method for the estimation of optical flow in anime, we performed ablation experiments on the

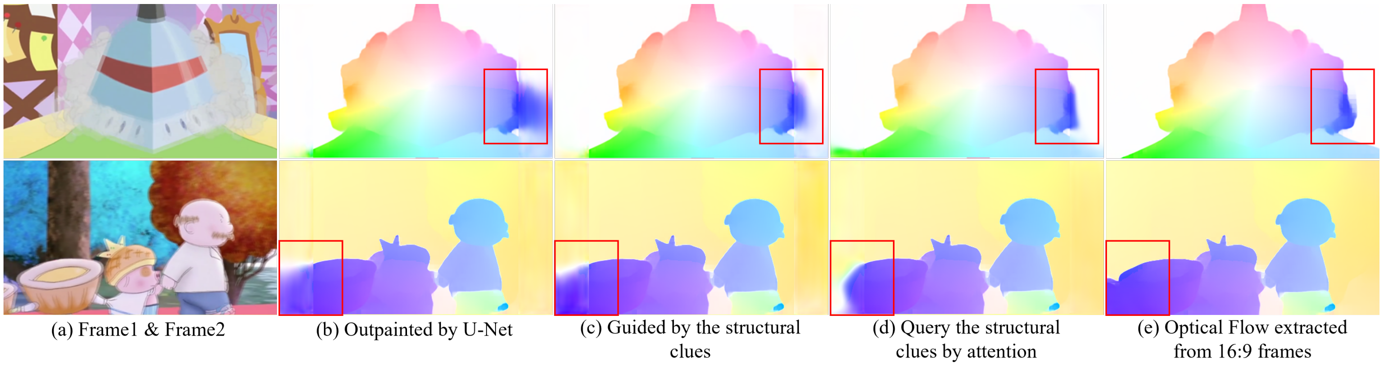

(a) Frame1 & Frame2
(b) Outpainted by U-Net
(c) Guided by the structural clues
(d) Query the structural clues by attention
(e) Optical Flow extracted from 16:9 frames

Fig. 13. Ablation Study of optical flow outpainting model. The first column shows the superposition of the two frames. The second column shows the optical flow outpainted only using the U-Net [52] network. The third column shows the results generated by the Unet and directly using the structural clues. The fourth column shows the result obtained by adding the proposed structural clue query block while using structural clues. The fifth column shows the pseudo-labels extracted from 16:9 input frames.

AnimeRun [53] dataset using RAFT [6] as the baseline method.

We retrained the RAFT model on the original AnimeRun dataset (Fig. 12 (b)), the AnimeRun dataset with sketch line enhancement (Fig. 12 (c)), the AnimeRun dataset with region enhancement (Fig. 12 (d)), and the AnimeRun dataset with both sketch line and region enhancement (Fig. 12 (e)), respectively.

In the first example, RAFT's output contained numerous detail errors in the wheel areas due to the unique challenges presented by anime frames (shown in red and blue boxes). By enhancing the image data based on sketch lines, our method was able to better highlight the motion details in the smaller wheel area (shown in red box), and by region-based enhancement, the results can show good emphasis of the motion details in the larger wheel area (shown in blue box). In the final results, both enhancement methods complemented each other and achieved better results than using either method alone.

In the second example, the motion boundary of the foreground extracted by RAFT is not clear (red box area), which has two reasons. First, the special line characteristics of anime images make this type of boundary area difficult to match correctly. After enhancing the image with sketch lines, the situation of motion boundaries is improved(red box). Second, due to the pure color and low-texture areas in the background, it is difficult to locate the features, and similar areas can be found in other parts of the background(blue box). After applying region-based enhancement to the image, the motion boundary in the red box and the motion situation of the pure color area in the background are improved(blue box).

4.2.2 Optical Flow Outpainting

During the outpainting stage of optical flow, we extract optical flow from 4:3 input cartoon frames and expand the field of view of the optical flow based on structural clues provided by the reference frame. To verify the effectiveness of the model, we tested its performance under conditions with and without structural clues.

As shown in Fig. 13, it can be seen that under experimental conditions, structural cues can effectively guide the optical flow restoration.

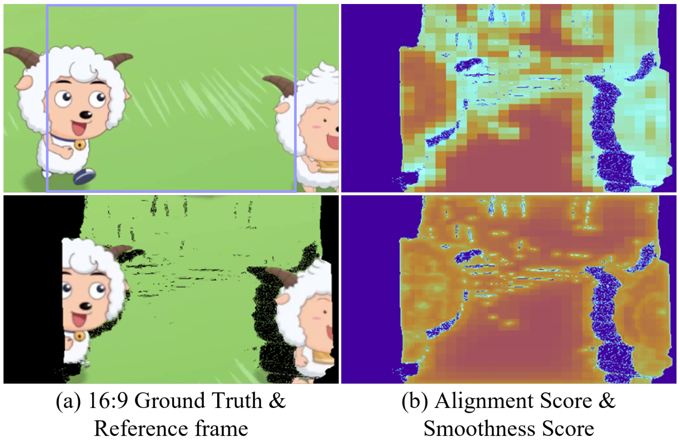

(a) 16:9 Ground Truth & Reference frame
(b) Alignment Score & Smoothness Score

Fig. 14. Visualization of two different reliability scores. The top left image shows the 16:9 ground truth frame of our current target frame. The blue box parts are cropped as 4:3 input and fed into the network. The bottom left image shows a reference frame generated by aligning neighboring frames to the input. The top right image displays the alignment score, while the bottom right image shows the smoothness score.

4.2.3 Reference Frames Reliability Estimation

As shown in Fig. 14, there are some alignment errors, pixel splashing, and small holes in a single aligned frame. Therefore, we propose two reliability scores for this: an alignment score and a smoothness score, which are demonstrated on the right side of Fig. 14. The alignment score assigns higher activation levels to areas with alignment errors, while the smoothness score assigns higher activation levels to areas with pixel splashing and hole edges. Furthermore, we pre-use a pixel valid mask to mark large hole areas.

4.3 Comparison Experiments

4.3.1 Visual Comparison

We conduct a visual comparison on our test dataset with three main competitors: the video inpainting method E2FGVI [27], the extrapolation method FGVC [23] and the video outpainting method [40]. The comparison results are illustrated in Fig. 15.

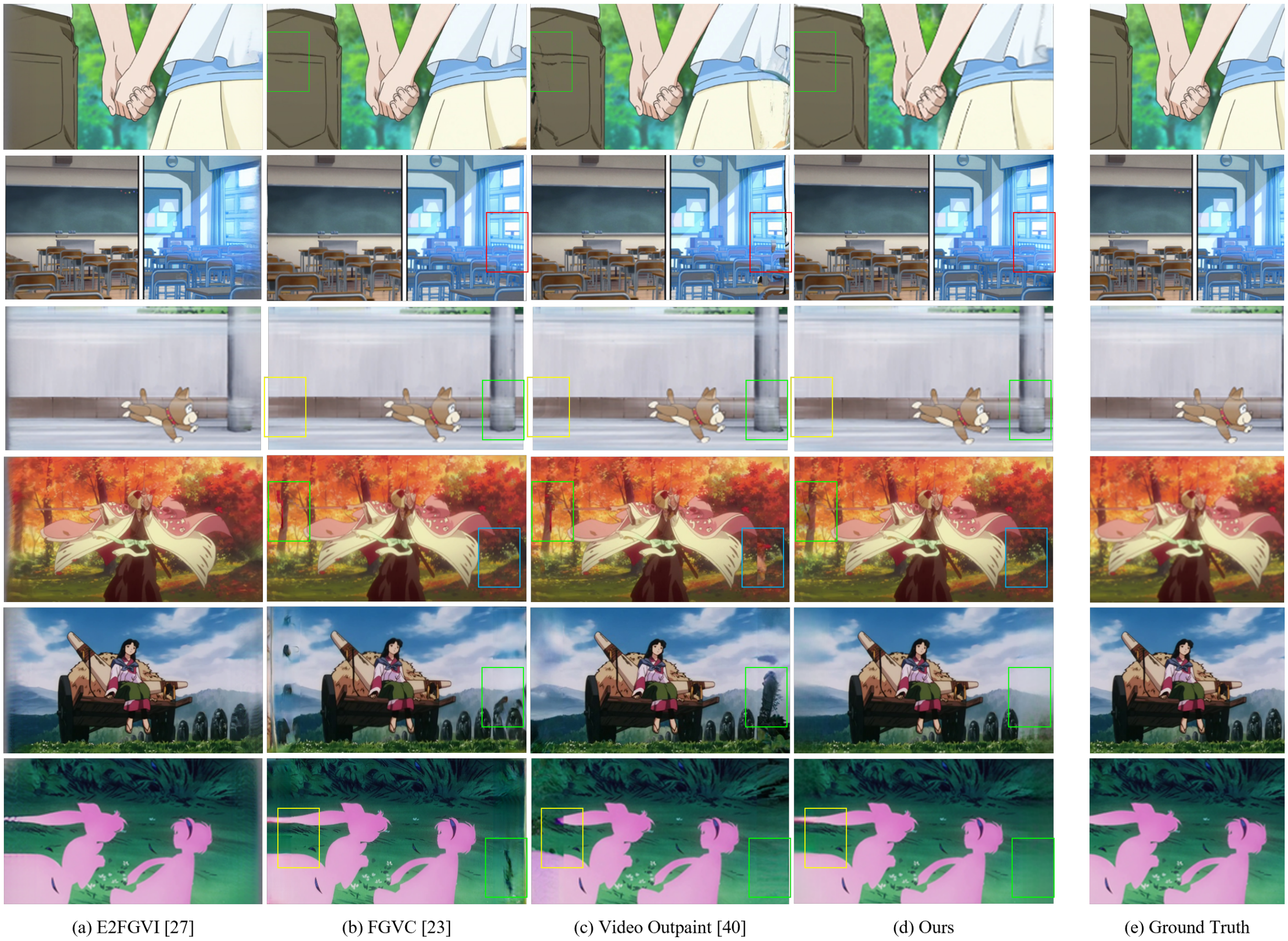


Fig. 15. Visual comparison against our competitors. The examples shown in the first through third rows demonstrate the results of cropping 16:9 cartoon animations to 4:3 as inputs. The examples in the fourth through sixth rows show the results of classic 4:3 cartoon animations as inputs (without 16:9 ground truth). The green, red, blue, and yellow boxes mark some important areas.

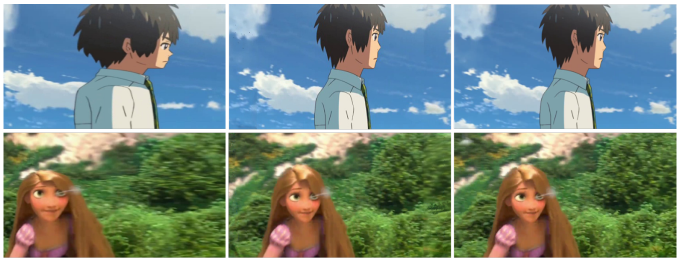
We first compare our method to the video inpainting method E2FGVI [27], and we find that their "inpainted" results cannot fully complete the 16:9 field of view. As shown in Fig. 15(a), most of their results suffer from the blurring and scratch artifacts as if the field of view are covered by a mystery mask. We believe this is because the E2FGVI method is designed for hole filling based on the information of surrounded pixels. In our task, the method may find it difficult to guess the external regions by growing the boundary pixels only.

FGVC [23] achieves competitive results among our competitors. Their results are usually free from blurring artifacts and manage to repair the missing FOVs for some videos. However, we see apparent tearing, slicing and trailing artifacts around the 4:3 FOV boundaries (as shown in the second and the fifth examples in Fig. 15 (b)). Moreover, their results may introduce extra unwanted objects copied from the subject, which breaks the original video composition and may confuse the audience, as shown in the fifth example in Fig. 15 (b). We believe this is due to its limited motion estimation ability when processing video sequences of enormous motion.

Among the competitors, the video outpainting method [40] is same to our task. It expands the field of view by separately processing the foreground and background. However, the method uses only gradient minimization to solve the flow value when completing the optical flow. This leads to difficulties in predicting the motion of the repaired area, particularly when the foreground object moves to the edge of the frame (as shown in the third example in Fig. 15 (c)).

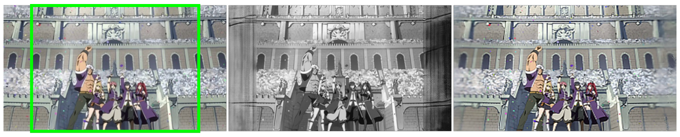
In sharp contrast, our method features a more advanced cartoon animation optical flow estimation that is guided by region information to better predict motion, which leads to higher quality after the final blending in the extra FOV. Thanks to the high-quality motion estimation, our results are free from ghosting and blurring artifacts. Moreover, the cartoon frame outpainting stage helps ensure the sharpness and reduces the tearing and distortion artifacts in the results (as shown in Fig. 15 (d)).

Retargeting Methods. There has been some research on retargeting methods for changing the aspect ratio of media content. However, most of these methods use energy functions to evaluate the importance of each pixel and perform uneven scaling of the objects in the image, which can damage the structural features of the content. Compared



(a) EFVR [5] (b) Ours (c) Ground Truth

Fig. 16. The results generated by video retargeting method EFVR [5] and our methods.



(a) Ground Truth (b) Seamless Manga Inpainting[29] (c) Ours

Fig. 17. The outpaint results of the Seamless Manga Inpainting [29] in cartoon fields.

to methods of this kind, our method has significant technical advantages. In this section, we have chosen EFVR [5] as a representative method for retargeting to display the results. As shown in Fig. 16, the repositioning method is challenging to fully preserve the structural characteristics of the objects when changing the aspect ratio.

Cartoon Fields Methods. There have been some studies on image restoration techniques for cartoon. And the research on manga inpainting is relatively active. However, applying these techniques to our task is not effective due to significant differences between the features of manga and anime. In this section, we chose Seamless Manga Inpainting [29] as a representative method for manga inpainting to showcase its effectiveness in Fig. 17.

Another anime inpainting project uses anime data to retrain image inpainting methods, such as EdgeConnect [31], which were designed for natural images. However, due to the differences between images and videos, these methods cannot maintain temporal consistency when applied to video content. Therefore, we conducted a comparative experiment using a video inpainting method [27] designed for natural image videos but retrained on our anime datasets. The results are shown in Fig. 15.

4.3.2 *Quantitative Comparison*

The superiority of our approach is also proven with a quantitative study shown in Table. 1. We choose PSNR, SSIM [54], MSE, LPIPS [55], VFID [56], and flow warping error E_{warp} [51] to evaluate the performance of the relevant methods. Specifically, PSNR, SSIM, and MSE are utilized for distortion-oriented video assessment, while LPIPS and VFID are employed for evaluating perceptual similarity from the perspectives of images and videos, respectively. Moreover, the flow warping error E_{warp} is used to measure temporal stability. We achieve the best scores in all metrics except flow warping error. We believe the advantage does not only come from our high visual quality but also a more precise assembly of reference frames. In comparison, the competitors may not fully recover the motion and may tend to guess the information in the outpainting area and thus fail to recover the ground truth frame faithfully. Furthermore, we argue that there is an equilibrium between perceptual distance and temporal consistency (Eq. 13), even though the Video Outpaint method [40] has achieved the best flow warping error, its other metrics have been much worse than ours, which means that to ensure temporal consistency, the model sacrifices quite a lot of image quality and faithfulness in preserving the original image contents from the motion priors.

TABLE 1
Quantitative evaluation of the video outpainting quality. ${E_{warp}}^*$ denotes $E_{warp} \times 10^{-2}$.

	PSNR ↑	SSIM ↑	MSE ↓	LPIPS ↓	VFID ↓	${E_{warp}}^*$ ↓
Video Outpaint [40]	24.63	0.9688	0.0661	0.0072	0.216	**0.0519**
E2FGVI [27]	25.27	0.9806	0.0622	0.0038	0.211	0.0526
FGVC [23]	24.91	0.9761	0.0614	0.0058	0.266	0.0529
Ours	**27.04**	**0.9827**	**0.0498**	**0.0036**	**0.188**	0.0575

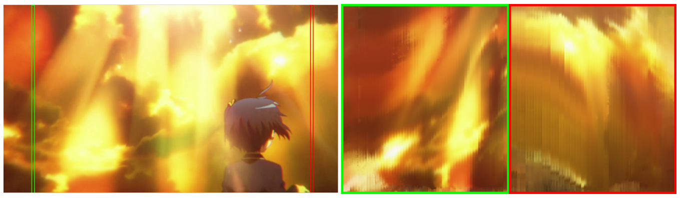
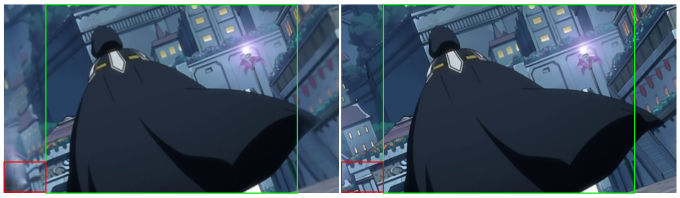


Fig. 18. The y-t slices of the frame sequence results are shown, where the left side displays the ground truth frame and the green and red boxes indicate the sampling positions for the slices. The right side displays the slice results of the sequence, where the left half corresponds to the green box area and the right half corresponds to the red box area.



(a). Result (b). Ground Truth

Fig. 19. Artifacts caused by inpainting

4.4 Temporal Consistency

To further evaluate the performance of the method on the entire video sequence, we experiment with video sequences with various motion styles. Fig. 18 shows the results of the y-t slice of the video, and more video results are shown in the supplementary material. It can be seen that our method has the ability to preserve temporal consistency.

4.5 Computational Efficiency Analysis

We calculate the computational efficiency on a dataset of approximately 600 frames of cartoon animation in 10 videos. The image size of each frame is 360 × 270. The computation time for the three subprocedures: anime optical flow estimation, optical flow outpainting, and cartoon frames outpainting is shown in Table 2.

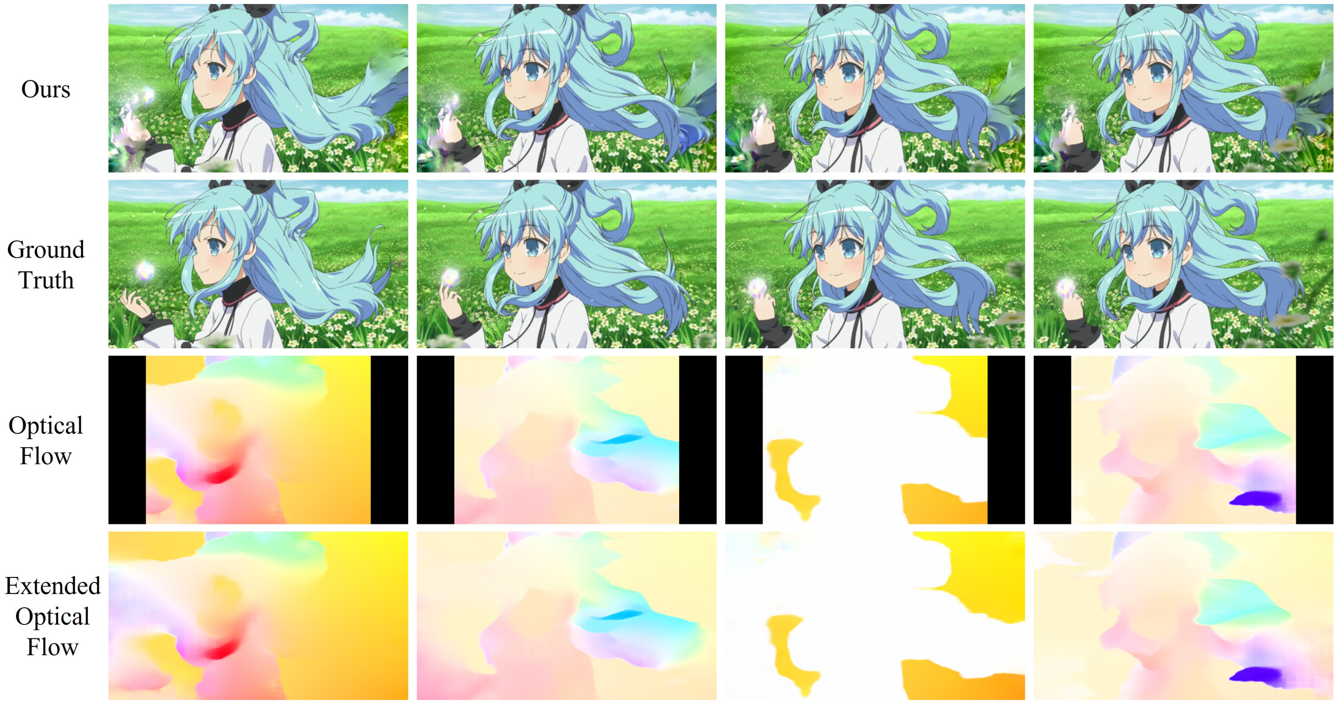


Fig. 20. Outpainting results of the cartoon video with complex camera movement. In this example, we demonstrate a challenging case with a combination of character movement and a fast rotating camera. The frames are consecutively sampled without skipping. Our results are obtained by the proposed outpainting approach from the center-cropped 4:3 area. The last two lines are the forward optical flow obtained from the anime flow estimation stage and the extended forward optical flow obtained from the anime flow estimation stage, respectively.

TABLE 2
Computational efficiency analysis.

Component	Time (second)
Anime optical flow estimation	0.1049
Optical flow outpainting	0.7230
Cartoon frames outpainting	18.6846
Total	19.5125

4.6 Discussions

4.6.1 Limitations

Although our approach manages to outpaint the FOV for most of the videos, there is still a small portion of the videos that may not be fully outpainted, as described in Sec. 3. Firstly, if the video does not provide sufficient information to fix the target area of all frames, we used image-based inpainting to complete the unfilled regions. This may cause human-observable temporal inconsistency at the corner locations, such as Fig. 19. In addition, we find that our method is relatively weak in processing the beginning and ending of sequences, as the outpainting can only receive clues from one temporal direction. Human assistance may still be required in some of those cases.

Secondly, extending animations in a sequential manner can be influenced by the results of earlier stages. Suppose the predicted or extended optical flow is inaccurate, it may result in the loss of guidance from optical flow vectors. Consequently, this prevents the stable propagation of pixels from neighboring frames to the current frame, hindering the achievement of the desired outpainting results. As illustrated in Fig. 20, When encountering rapid non-linear camera movements, optical flow predictions sometimes fall short of accuracy (as shown in the first frame of Fig. 20, where the optical flow estimation on the left-hand side of the character is problematic). This incorrect optical flow estimation causes the error to propagate to the extended optical flow outside the original 4:3 area, resulting in textures and structures mistakenly appearing in the exterior area, in addition to some distortion issues depicted in this figure.

Finally, our proposed method is still feasible when dealing with natural videos; however, our tailored approaches such as sketch enhancement and region enhancement are not optimized for natural scenes and thus may not achieve human-preferred results on natural videos.

4.6.2 Future Work

Considering that the latest diffusion models have made significant progress in the fields of image and video generation, we posit that the application of diffusion methodologies to cartoon animation outpainting represents a highly promising avenue for research. In our future research, we plan to explore and develop this method in depth to further improve the quality of cartoon animation outpainting.

5 Conclusion

We propose a novel cartoon animations outpainting framework that extends the field of view of 4:3 cartoon animations to 16:9 without any prior knowledge of the camera or the objects. The key insight of the FOV inference from motion directs us to construct the three-stage design of anime optical flow estimation, optical flow outpainting and cartoon frames outpainting. We have achieved high-quality artifact-free outpainting for a vast diversity of cartoon animations with the three-stage design. Both qualitative and quantitative experiments show that our approach has achieved the highest output quality amongst all state-of-the-art methods.

Acknowledgments

This work was supported partly by National Natural Science Foundation of China (Nos. 62002232 and 62273241), the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. UGC/FDS11/E03/22), and the National Science and Technology Council under Nos. 111-2221-E-006-112-MY3, Republic of China (ROC), Taiwan.

References

[1] Yu-Shuen Wang, Hui-Chih Lin, Olga Sorkine, and Tong-Yee Lee. Motion-based video retargeting with optimized crop-and-warp. In *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, New York, NY, USA, 2010. Association for Computing Machinery.

[2] Weiming Dong, Fuzhang Wu, Yan Kong, Xing Mei, Tong-Yee Lee, and Xiaopeng Zhang. Image retargeting by texture-aware synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 22(2):1088–1101, 2015.

[3] Sung In Cho and Suk-Ju Kang. Temporal incoherence-free video retargeting using foreground aware extrapolation. *IEEE Transactions on Image Processing*, 29:4848–4861, 2020.

[4] Shih-Syun Lin, Chao-Hung Lin, I-Cheng Yeh, Shu-Huai Chang, Chih-Kuo Yeh, and Tong-Yee Lee. Content-aware video retargeting using object-preserving warping. *IEEE Transactions on Visualization and Computer Graphics*, 19(10):1677–1686, 2013.

[5] Bo Yan, Binhang Yuan, and Bo Yang. Effective video retargeting with jittery assessment. *IEEE Transactions on Multimedia*, 16(1):272–277, 2014.

[6] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020.

[7] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8121–8130, 2022.

[8] Shih-Syun Lin, Chao-Hung Lin, Yu-Hsuan Kuo, and Tong-Yee Lee. Consistent volumetric warping using floating boundaries for stereoscopic video retargeting. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(5):801–813, 2015.

[9] W Tan, B Yan, C Lin, and X Niu. Cycle-IR: Deep Cyclic Image Retargeting. *IEEE Transactions on Multimedia*, 22(7):1730–1743, 2020.

[10] Shih-Syun Lin, Chao-Hung Lin, Shu-Huai Chang, and Tong-Yee Lee. Object-coherence warping for stereoscopic image retargeting. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(5):759–768, 2013.

[11] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. *ACM Transactions on Graphics*, 26(99):10, 2007.

[12] Donghyeon Cho, Jinsun Park, Tae-Hyun Oh, Yu-Wing Tai, and In So Kweon. Weakly- and Self-Supervised Learning for Content-Aware Deep Image Retargeting. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[13] Sung In Cho and Suk-ju Kang. Temporal Incoherence-Free Video Retargeting. 29:4848–4861, 2020.

[14] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5792–5801, 2019.

[15] Ya Zhou, Zhibo Chen, and Weiping Li. Weakly supervised reinforced multi-operator image retargeting. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(1):126–139, 2021.

[16] Seung Joon Lee, Siyeong Lee, Sung In Cho, and Suk-Ju Kang. Object detection-based video retargeting with spatial–temporal consistency. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(12):4434–4439, 2020.

[17] Zhong Ji, Jiacheng Hou, Yimu Su, Yanwei Pang, and Xuelong Li. G2lp-net: Global to local progressive video inpainting network. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(3):1082–1092, 2023.

[18] Zhiliang Wu, Changchang Sun, Hanyu Xuan, Kang Zhang, and Yan Yan. Divide-and-conquer completion network for video inpainting. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(6):2753–2766, 2023.

[19] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3718–3727, 2019.

[20] Chaoqun Wang, Xuejin Chen, Shaobo Min, Jiaping Wang, and Zheng-Jun Zha. Structure-guided deep video inpainting. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(8):2953–2965, 2021.

[21] Sungho Lee, Seoung Wug Oh, Daeyeun Won, and Seon Joo Kim. Copy-and-paste networks for deep video inpainting. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:4412–4420, 2019.

[22] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 528–543, Cham, 2020. Springer International Publishing.

[23] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 713–729, Cham, 2020. Springer International Publishing.

[24] Seoung Wug Oh, Sungho Lee, Joon Young Lee, and Seon Joo Kim. Onion-peel networks for deep video completion. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:4402–4411, 2019.

[25] Ya Liang Chang, Zhe Yu Liu, Kuan Ying Lee, and Winston Hsu. Free-form video inpainting with 3D gated convolution and temporal patchGAN. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:9065–9074, 2019.

[26] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019.

[27] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17562–17571, June 2022.

[28] Kaidong Zhang, Jingjing Fu, and Dong Liu. Flow-guided transformer for video inpainting. In *European Conference on Computer Vision*, pages 74–90. Springer, 2022.

[29] Minshan Xie, Menghan Xia, Xueting Liu, Chengze Li, and Tien-Tsin Wong. Seamless manga inpainting with semantics awareness. *ACM Transactions on Graphics (SIGGRAPH 2021 issue)*, 40(4):96:1–96:11, August 2021.

[30] youyuge34. Anime-inpainting. https://github.com/youyuge34/Anime-InPainting, March 2019.

[31] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[32] Shuyang Luo, Haoran Xie, and Kazunori Miyata. Sketch-based anime hairstyle editing with generative inpainting. In *2021 Nicograph International (NicoInt)*, pages 7–14. IEEE, 2021.

[33] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4463–4471, 2017.

[34] Shuai Huo, Dong Liu, Bin Li, Siwei Ma, Feng Wu, and Wen Gao. Deep network-based frame extrapolation with reference frame alignment. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(3):1178–1192, 2021.

[35] Laura Turban, Fabrice Urban, and Philippe Guillotel. Extrafoveal Video Extension for an Immersive Viewing Experience.

IEEE Transactions on Visualization and Computer Graphics, 23(5):1520–1533, 2017.

[36] Tamar Avraham and Yoav Y. Schechner. Ultrawide foveated video extrapolation. *IEEE Journal on Selected Topics in Signal Processing*, 5(2):321–334, 2011.

[37] Naoki Kimura and Jun Rekimoto. Extvision: Augmentation of visual experiences with generation of context images for peripheral vision using deep neural network. *Conference on Human Factors in Computing Systems - Proceedings*, 2018-April:1–10, 2018.

[38] Jungjin Lee, Sangwoo Lee, Younghui Kim, and Junyong Noh. Screenx: Public immersive theatres with uniform movie viewing experiences. *IEEE Transactions on Visualization and Computer Graphics*, 23(2):1124–1138, 2016.

[39] Liqian Ma, Stamatios Georgoulis, Xu Jia, and Luc Van Gool. FoV-Net : Field-of-View Extrapolation Using. 6(3):4321–4328, 2021.

[40] Loïc Dehan, Wiebe Van Ranst, Patrick Vandewalle, and Toon Goedemé. Complete and temporally consistent video outpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 687–695, June 2022.

[41] Cheng Xu, Wei Qu, Xuemiao Xu, and Xueting Liu. Multi-scale flow-based occluding effect and content separation for cartoon animations. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[42] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 5436–5445, 2020.

[43] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017.

[44] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[45] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *International Conference on Computer Vision (ICCV)*, 2021.

[46] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):2011–2023, 2020.

[47] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[48] Masaki Saito and Yusuke Matsui. Illustration2Vec: A Semantic Vector Representation of Illustrations. In *SIGGRAPH Asia 2015 Technical Briefs*, SA '15, New York, NY, USA, 2015. Association for Computing Machinery.

[49] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015.

[50] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019.

[51] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 170–185, 2018.

[52] Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9351(Cvd):12–20, 2015.

[53] Li Siyao. AnimeRun : 2D Animation Visual Correspondence from Open Source 3D Movies. (NeurIPS 2022):1–12.

[54] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[55] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[56] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 1152–1164, 2018.

Huisi Wu received his B.E. and M.E. degrees both in Computer Science from the Xi'an Jiaotong University (XJTU) in 2004 and 2007, respectively. He obtained his Ph.D. degree in Computer Science from The Chinese University of Hong Kong (CUHK) in 2011. He is currently a Professor in the College of Computer Science and Software Engineering, Shenzhen University. His research interests include computer graphics and computer vision.

Hao Meng received his B.Eng. degree from Shanxi University (SXU) in 2019 and is now a graduate student in the College of Computer Science and Software Engineering at Shenzhen University (SZU). His research interests include computer graphics, computer vision, machine learning, and deep learning.

Chengze Li received their B.Eng. degree from University of Science and Technology of China in 2013, and Ph.D. degree in Computer Science and Engineering from the Chinese University of Hong Kong in 2020. He is currently an Assistant Professor in the School of Computing and Information Sciences, Caritas Institute of Higher Education. Their research interests include 2D non-photorealistic media analysis and processing, computational photography, and computer graphics.

Xueting Liu received her BE degree in Computer Science and Technology from Tsinghua University and Ph.D. degree in Computer Science from The Chinese University of Hong Kong in 2009 and 2014, respectively. Her research interests include computer animation, computer graphics, computer vision, and deep learning.

Zhenkun Wen received his M.Sc. degree in Science and Technology from Tsinghua University in 1999. Since 1987, he has been engaged in computing research and teaching in Shenzhen University. He is currently a professor of computing and software, and director of the Science and Technology Department of Shenzhen University. His research interests are in video tampering detection and location, video information security, and information management system design and implementation.

Tong-Yee Lee (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Washington State University, Pullman, in May 1995. He is currently a Chair Professor in the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan, ROC. He leads the Computer Graphics Group, Visual System Laboratory, National Cheng-Kung University (http://graphics.csie.ncku.edu.tw). His current research interests include computer graphics, non-photorealistic rendering, medical visualization, virtual reality, and media resizing. He is a Senior Member of the IEEE and a Member of the ACM. He also serves on the editional boards of the IEEE Transaction on Visualization and Computer Graphics.