RNA: Video Editing with ROI-based Neural Atlas

Jaekyeong Lee*, Geonung Kim*, and Sunghyun Cho

POSTECH
{jaekyeong,k2woong92,s.cho}@postech.ac.kr
https://jaekyeongg.github.io/RNA

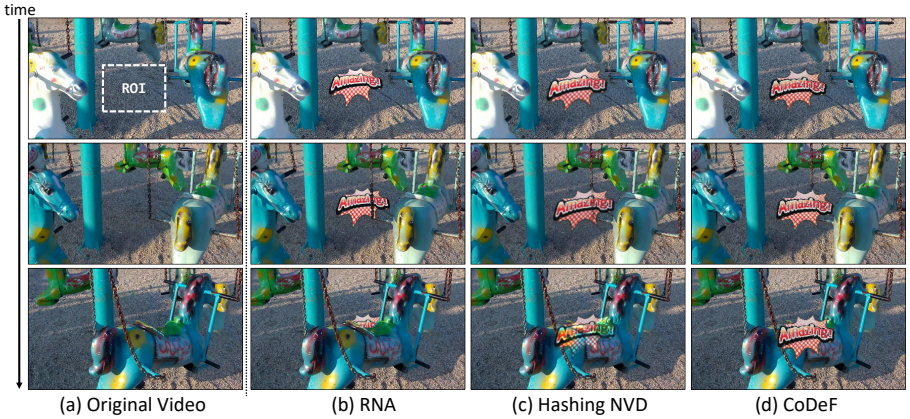

Fig. 1: (b) Our video editing achieves natural editing outcomes, successfully considering the occlusions from the thin chain (2nd row) and the toy horse of the carousel (3rd row). In contrast, (c) Hashing NVD [4] results in ghosting artifacts, and (d) CoDeF [13] neglects the occlusion from moving objects, failing to produce natural editing results.

Abstract. With the recent growth of video-based Social Network Service (SNS) platforms, the demand for video editing among common users has increased. However, video editing can be challenging due to the temporally-varying factors such as camera movement and moving objects. While modern atlas-based video editing methods have addressed these issues, they often fail to edit videos including complex motion or multiple moving objects, and demand excessive computational cost, even for very simple edits. In this paper, we propose a novel region-of-interest (ROI)-based video editing framework: ROI-based Neural Atlas (RNA). Unlike prior work, RNA allows users to specify editing regions, simplifying the editing process by removing the need for foreground separation and atlas modeling for foreground objects. However, this simplification presents a unique challenge: acquiring a mask that effectively handles occlusions in the edited area caused by moving objects, without relying on an additional segmentation model. To tackle this, we propose a novel mask refinement approach designed for this specific challenge. Moreover, we introduce a soft neural atlas model for video reconstruction to ensure high-quality editing results. Extensive experiments show that RNA offers a more practical and efficient editing solution, applicable to a wider range of videos with superior quality compared to prior methods.

* equal contribution

1 Introduction

With the recent growth of video-based Social Network Service (SNS) platforms such as YouTube Shorts, there has been an explosive increase in the demand for video editing among common users. However, video editing is a challenging task since videos have temporally-varying characteristics caused by various factors such as camera movement and moving objects, and editing such videos must involve addressing these tricky factors in a temporally-consistent way.

For temporally-consistent video editing, atlas-based video editing methods such as Layered Neural Atlas (LNA) [9], Hashing Neural Video Decomposition (Hashing NVD) [4], and Content Deformation Fields (CoDeF) [13] have been proposed. These methods operate under an assumption, in which each video frame consists of sprite layers representing the background and foreground objects, and the appearance of the sprites are constant, while their positions and shapes change by their motions, which can be represented, e.g., by optical flow [7]. Based on this assumption, the typical procedure of atlas-based approaches is as follows. First, they separate an input video into foreground objects and the background using instance segmentation, then estimate the atlases and motions of the separated regions. Finally, a user edits these atlases, and then an edited video is reconstructed from the edited atlases and the motion data.

Unfortunately, the atlas-based video editing approaches have inherent limitations caused by their strategy that explicitly models each foreground object independently. Firstly, accurate instance segmentation and atlas estimation of multiple foreground objects are challenging especially when foreground objects have complex motions. Failures in segmentation and atlas estimation result in error-prone editing outcomes, e.g., ghosting artifacts or failures of handling occlusions in Fig. 1 (c) and (d). Secondly, for the segmentation of foreground objects, the atlas-based video editing approaches require users to specify all the foreground objects including those that the users do not intend to edit. This requirement significantly diminishes the user-friendliness of the video editing interface, particularly when the input video contains multiple foreground objects. Lastly, in practical scenarios, users frequently aim to modify only a specific region of a video, such as introducing a new object or altering the texture of an existing one. Despite this, prior atlas-based video editing methods, such as LNA [9] and Hashing NVD [4], necessitate the complete reconstruction of the entire video, leading to substantial computational resource requirements. In particular, they demand memory space and computation time that scale with the number of objects. This results in computation times spanning several hours for videos containing multiple objects.

This paper proposes a novel region-of-interest (ROI)-based video editing framework: ROI-based Neural Atlas (RNA). Unlike previous methods, RNA allows users to specify a region where editing will occur and then optimizes a single atlas exclusively for the specified region. After the optimization, editing is performed directly on the atlas. Finally, RNA reconstructs an edited video using the edited atlas for the ROI, and the original pixel values for the non-edited regions. Our ROI-based approach enjoys a couple of advantages over prior

methods. Firstly, our approach does not need foreground separation performed by segmentation models, which are often unreliable and cumbersome for users. Additionally, by focusing solely on an ROI atlas, it avoids the necessity of atlas estimation for all moving foreground objects, thus enabling editing of videos with complex motions without incurring ghosting artifacts, and maintaining constant computational resources regardless of the number of foreground objects.

Eliminating the foreground separation process introduces a unique challenge that sets it apart from previous methods: acquiring a mask that effectively addresses occlusions in the edited area caused by moving objects, without depending on an additional segmentation model. To this end, we propose a novel approach for mask refinement tailored to the specific challenge. Specifically, after estimating the atlas and mask similarly to previous methods [4, 9], we further refine the imperfect mask using a novel self-supervised method. Additionally, we introduce a novel soft neural atlas model, which employs a soft mask for handling boundaries between occluding objects and an edited atlas for high-quality video reconstruction.

Our main contributions can be summarized as follows.

- We first propose a novel ROI-based video editing framework that utilizes a single atlas for editing regions, which removes requirement for the foreground separation process and atlas modeling for all foreground objects.
- To estimate an accurate mask without foreground separation, we propose a novel mask refinement method.
- We also introduce a novel soft neural atlas model for more natural-looking video reconstruction.
- Extensive experiments demonstrate the efficiency and effectiveness of our video editing framework, underscoring its practicality and versatility, compared to previous state-of-the-art methods.

2 Related Work

Atlas-based Video Editing Video editing via 2D atlas images was first introduced by Unwrap Mosaics [14], which decomposes a video into a series of 2D texture maps termed "unwrap mosaics", and establishes a mapping from these mosaics to video frames. Editing is then directly applied to the unwrap mosaics. To enhance this approach, which involves complex optimization, naïve layering with binary segmentation masks, and limited adaptability for non-rigid objects, LNA [9] introduces an end-to-end self-supervised method that reconstructs videos using neural atlases with alpha blending. To expedite the optimization, Hashing NVD [4] and CoDeF [13] adopt hash encoding [11] to represent input videos. To facilitate more advanced editing, Text2Live [1] and StableVideo [3] incorporate text-based image editing into the LNA [9] framework. In another direction, Deformable Sprites [19] estimates sprite images and their warping parameters to reconstruct an input video.

While atlas-based editing methods enable temporally-consistent video editing, their capabilities are often limited to simple videos featuring a single fore-

ground object with mild movement. It is primarily due to the unreliable foreground separation process performed using an off-the-shelf segmentation model [5, 6], and atlas optimization for all moving foreground objects. Meanwhile, RNA eliminates the need for foreground separation and employs a single atlas where the editing will occur, thereby achieving robust video editing that is applicable to a broader range of videos.

Propagation-based Video Editing Video propagation refers to general techniques that address video-related problems in a temporally-consistent manner. It involves selecting a key frame from a video, applying a specific action to that frame, and then propagating it to the remaining video frames. For example, Jampani et al. [8] and Oh et al. [12] propose video object mask segmentation methods, and Wang et al. [17] introduce a more general label propagation method that covers object masks, textures, and human poses. For video editing, Meyer et al. [10] propose a phase-based modification transfer, and Texler et al. [16] introduce an editing method using patch-based training with a few shots. These methods often show limited editing capabilities, since they largely rely on a single frame rather than exploiting multiple frames. For example, Meyer et al.'s method is applicable only to static videos, and Texler et al.'s is limited to video stylization.

3 ROI-based Neural Atlas

In this section, we first present our ROI-based neural atlas model, which serves as the foundation for our video editing framework (Sec. 3.1). Then, we explain the three main components of our video editing framework: atlas estimation (Sec. 3.2), mask refinement (Sec. 3.3), and video reconstruction using a soft neural atlas model (Sec. 3.4).

3.1 ROI-based Neural Atlas Model

For temporally-consistent editing of a local region in a video, our method takes an input video, and an ROI specified by a user in a reference frame as input. Then, our method estimates a 2D atlas representing the temporally-invariant appearance of the ROI, and the mapping from each video frame to the atlas. Then, a user edits the 2D atlas using an image editing software such as Adobe Photoshop. Finally, an edited video is reconstructed from the edited atlas and the original video input. We assume that the ROI can be of any arbitrary shape, and that the ROI in the reference frame has no occluded pixels, while they can still be occluded by other objects in other frames.

To this end, we propose an ROI-based neural atlas model defined as:

$$\hat{c}_p = \mathbb{M}(p)\mathbb{L}(\mathbb{T}(p), t)\mathbb{A}(\mathbb{T}(p)) + (1 - \mathbb{M}(p))\, c_p, \tag{1}$$

where $p = (x, y, t)$ is a coordinate indicating the spatial position (x, y) at the t-th frame. c and $\hat{c}$ are the input and its reconstructed video, respectively. c_p and

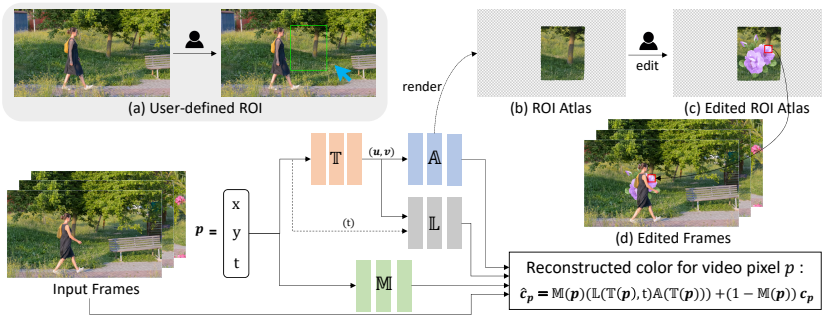

Fig. 2: Overall framework of RNA. For video editing, (a) a user selects a reference frame from an input video and specifies an ROI where they want to edit. (b) For the specified ROI, our method estimates a 2D atlas representing its temporally-invariant appearance. (c) Then, the user edits the 2D atlas. (d) Finally, an edited video is reconstructed from the edited atlas and the input video.

$\hat{c}_p$ are the pixel values of c and $\hat{c}$ at p, respectively. $\mathbb{M}(p)$ is a mask indicating whether the pixel (x, y) at the t-th frame belongs to the ROI, i.e., $\mathbb{M}(p) = 1$ if p belongs to the ROI, and $\mathbb{M}(p) = 0$ otherwise. $\mathbb{A}$ is an atlas representing the color values of the ROI. $\mathbb{A}$ is defined as a mapping from a 2D coordinate (u, v) to an RGB value. $\mathbb{T}$ is a mapping from p to a coordinate (u, v) on the atlas. $\mathbb{L}$ is a scaling function to model the spatial and temporal illumination change [4]. Specifically, $\mathbb{L}(\mathbb{T}(p), t)$ is a 3×3 diagonal matrix whose diagonal entries consist of scaling factors for the RGB color channels.

Fig. 2 illustrates our ROI-based neural atlas model in Eq. (1). The ROI can be occluded by other objects that were originally outside of the ROI at the reference frame, but moved into the ROI later, e.g., a person passing in front of the ROI specified on the background as shown in Fig. 2. $\mathbb{M}$ allows for handling such occlusions without explicitly modeling occluding objects. For the mappings $\mathbb{M}$, $\mathbb{A}$, $\mathbb{T}$, and $\mathbb{L}$, we adopt multi-layer perceptrons (MLPs). Also, we use hash encoding [11] for $\mathbb{M}$ and $\mathbb{A}$ following the recent neural atlas-based approaches [4, 13].

To edit a video using the model in Eq. (1), we first estimate the mappings $\mathbb{M}$, $\mathbb{A}$, $\mathbb{T}$, and $\mathbb{L}$ for a given video in an end-to-end supervised manner (Sec. 3.2). Then, we perform an additional mask refinement process to more accurately consider occlusions caused by foreground objects in motion (Sec. 3.3). Following this, we render a discretized version of the atlas $\mathbb{A}$. Users then edit this atlas as they desire, and obtain an edited atlas image $\mathbb{A}_{edit}$. Finally, an edited video is reconstructed using the video reconstruction method based on a novel soft neural atlas model, which will be described in Sec. 3.4.

3.2 Atlas Estimation

To edit a video, all the mappings $\mathbb{M}$, $\mathbb{A}$, $\mathbb{T}$, and $\mathbb{L}$ in Eq. (1) are estimated in an end-to-end self-supervised manner prior to editing. The estimation is performed

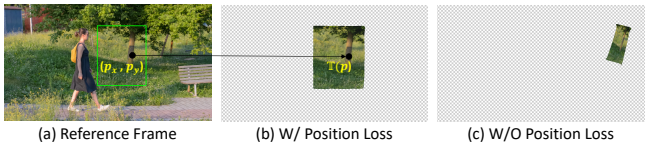

Fig. 3: (b) Atlas estimation with $\mathcal{L}_{pos}$ provides a user-friendly interface for editing, (c) while, without $\mathcal{L}_{pos}$, the estimated 2D atlas can be severely distorted, leading to less intuitive editing.

by minimizing a loss $\mathcal{L}$, which is defined as:

$$\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{rigid} + \mathcal{L}_{pos} + \mathcal{L}_{corr} + \mathcal{L}_{mask} + \mathcal{L}_{illum} \tag{2}$$

where $\mathcal{L}_{recon}$, $\mathcal{L}_{rigid}$, $\mathcal{L}_{pos}$, $\mathcal{L}_{corr}$, $\mathcal{L}_{mask}$, and $\mathcal{L}_{illum}$ represent the reconstruction, rigidity, position, correspondence, mask, and illumination losses, respectively. In the following, each loss will be elaborated in detail.

Reconstruction loss The reconstruction loss $\mathcal{L}_{recon}$ is used for estimating the mappings that can accurately reconstruct the input video. $\mathcal{L}_{recon}$ is defined as:

$$\mathcal{L}_{recon} = \sum_{p \in \mathcal{P}} \|\hat{c}_p - c_p\|_2^2, \tag{3}$$

where $\mathcal{P}$ is a set of pixels p in an input video, including those both inside and outside the ROI. During the estimation process, we randomly sample pixels for $\mathcal{P}$ in every epoch.

Rigidity loss Estimating the mappings only with the reconstruction loss may result in a severely distorted atlas, making editing challenging. To tackle this, we adopt the rigidity loss proposed by Kasten et al. [9], which is defined as:

$$\mathcal{L}_{rigid} = \lambda_{rigid} \sum_{p \in \mathcal{P}} (\|J_p^T J_p\|_F + \|(J_p^T J_p)^{-1}\|_F), \tag{4}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, J_p is the Jacobian matrix of the local transformation at p obtained from $\mathbb{T}$, and λ_{rigid} is a balancing weight for $\mathcal{L}_{rigid}$. The rigidity loss encourages local transformations of $\mathbb{T}$ to be as rigid as possible by enforcing the singular values of the Jacobians to be close to 1. We refer the readers to the Supplemental Document for more details of J_p.

Position loss While the rigidity loss can prevent severe local distortions, resulting atlases may still be smoothly distorted. Moreover, it does not prevent global scaling or rotation of the atlas, and may result in a too small and rotated atlas, which is difficult to edit, as shown in Fig. 3 (c). Thus, to support more user-friendly and convenient editing, we introduce a position loss $\mathcal{L}_{pos}$, which promotes the positions of the content in the atlas to be similar to those of the

user-specified ROI in the reference video frame, so that it can prevent global scaling and rotation as shown in Fig. 3 (b). $\mathcal{L}_{pos}$ is defined as:

$$\mathcal{L}_{pos} = \lambda_{pos} \sum_{p \in \mathcal{P}_{ROI}} \|\mathbb{T}(p) - [x, y]^T\|_2, \tag{5}$$

where λ_{pos} is a balancing weight for $\mathcal{L}_{pos}$, $\mathcal{P}_{ROI}$ is the set of pixels in the ROI in the reference frame, and $p = (x, y, t)$.

Correspondence loss The correspondence loss $\mathcal{L}_{corr}$ ensures that the corresponding pixels across different video frames in the ROI are mapped to the same UV coordinate in the atlas. To this end, $\mathcal{L}_{corr}$ is defined as:

$$\mathcal{L}_{corr} = \lambda^{(1)}_{corr} \sum_{(p,p_r)\in\mathcal{P}_r} \mathbb{M}(p)\|\mathbb{T}(p) - \mathbb{T}(p_r)\|_2 + \lambda^{(2)}_{corr} \sum_{(p,p_a)\in\mathcal{P}_a} \mathbb{M}(p)\|\mathbb{T}(p) - \mathbb{T}(p_a)\|_2, \tag{6}$$

where $\lambda^{(1)}_{corr}$ and $\lambda^{(2)}_{corr}$ are balancing weights. $\mathcal{P}_r$ and $\mathcal{P}_a$ are sets of corresponding pixel pairs. Specifically, $\mathcal{P}_r = \{(p, p_r)\}$ is a set such that p is a pixel in the input video and p_r is its match in the reference frame. Similarly, $\mathcal{P}_a = \{(p, p_a)\}$ is a set such that p is a pixel in the input video and p_a is its match in an adjacent frame. $\mathcal{P}_r$ and $\mathcal{P}_a$ can be estimated using an off-the-shelf optical flow model [15]. For more accurate estimation of correspondences between distant frames, we aggregate multiple optical flow estimations. Further details can be found in the Supplemental Document. The first term on the right hand side in Eq. (6) ensures that the UV coordinate of a point in any given frame matches the UV coordinate of its corresponding point at the reference frame, while the second term is adopted for improving the temporal consistency of the UV coordinates between adjacent frames.

Mask loss The goal of the mask loss $\mathcal{L}_{mask}$ is to train the mask network $\mathbb{M}$ to output 1 for non-occluded pixels within the ROI and 0 otherwise. We define $\mathcal{L}_{mask}$ as:

$$\begin{aligned}\mathcal{L}_{mask} = -\lambda^{(1)}_{mask} \left\{ \sum_{p\in\mathcal{P}_{ROI}} \log \mathbb{M}(p) + \sum_{p\in\mathcal{P}^c_{ROI}} \log\left(1 - \mathbb{M}(p)\right) \right\} \\ + \lambda^{(2)}_{mask} \sum_{(p,p_r)\in\mathcal{P}_r} |\mathbb{M}(p) - \mathbb{M}(p_r)| + \lambda^{(3)}_{mask} \sum_{(p,p_a)\in\mathcal{P}_a} |\mathbb{M}(p) - \mathbb{M}(p_a)|,\end{aligned} \tag{7}$$

where $\lambda^{(1)}_{mask}$, $\lambda^{(2)}_{mask}$, and $\lambda^{(3)}_{mask}$ are balancing weights. In the first term, $\mathcal{P}^c_{ROI}$ is a set of the pixels outside the ROI in the reference frame. The first term on the right hand side is a binary cross-entropy loss to train M to have 1 if the pixel is inside the ROI and 0 otherwise. The second and third terms propagate the mask values to different frames in the input video based on the correspondence $\mathcal{P}_r$ and $\mathcal{P}_a$. As we assume that the ROI in the reference frame has no occluded pixels, we enforce M to be 1 for all the pixels inside the ROI in the reference frame using the first term. Then, M is trained to be 0 for the occluded pixels inside the ROI in different frames by the reconstruction loss and the second and third terms of the mask loss.

Fig. 4: Magnified masks and editing results with and without mask refinement. Without the mask refinement, the mask inaccuracy is significant at the boundaries of occluding object.

Illumination loss The illumination loss $\mathcal{L}_{illum}$ ensures that the mapping $\mathbb{L}$ accurately models changes in lighting over time. For $\mathcal{L}_{illum}$, we adopt the residual regularization loss of Chan et al. [4]. Specifically, assuming that the illumination does not change much from that of the reference frame, we define $\mathcal{L}_{illum}$ as:

$$\mathcal{L}_{illum} = \lambda_{illum} \sum_{p \in \mathcal{P}} \|\mathbb{L}(\mathbb{T}(p), t) - [1, 1, 1]^T\|_2^2, \tag{8}$$

where λ_{illum} is a balancing weight.

3.3 Mask Refinement

Fig. 5: A point p, which is outside of the ROI, is mapped into a plausible position within the ROI by $\mathbb{T}$.

While our atlas estimation process described in Sec. 3.2 accurately estimates $\mathbb{T}$, $\mathbb{A}$, $\mathbb{M}$, and $\mathbb{L}$ for most pixels, $\mathbb{M}$ can still contain errors, particularly around the boundaries of occluding objects, as shown by the red arrow in Fig. 4(b). This artifact occurs when non-occluded pixels are erroneously identified as occluded ones due to the difficulty in reliably estimating a mask for these pixels. To handle such boundary errors and to obtain a more accurate mask, we propose a novel mask refinement method, which is performed after atlas estimation. To this end, we leverage well-estimated $\mathbb{A}$ and $\mathbb{T}$, as shown in the example in Fig. 5. In the example, the point p, which was originally inside the ROI at the reference frame, is occluded by a foreground object. Nevertheless, $\mathbb{T}$ still learns to map p to a position inside the ROI due to the non-occluded counterparts of p at other frames. This is because $\mathbb{T}$ is modeled as an MLP, which is a piece-wise continuous function. Based on this property, we perform an additional occlusion test. Specifically, we test whether c_p is close enough to $\mathbb{A}(\mathbb{T}(p))$ to detect occluded pixels, and update the mask $\mathbb{M}$ accordingly.

Specifically, in the first stage of the refinement step, we first find a set of pixels in the input video potentially belonging to the ROI. To this end, we build a binary mask M_{roi} for the reference frame such that $M_{roi}(x, y) = 1$ if (x, y) is inside the ROI at the reference frame, and $M_{roi}(x, y) = 0$ otherwise. Then, we warp M_{roi} using the transformation $\mathbb{T}$ for the reference frame, and obtain a mask M^w_{roi} that is aligned to the atlas $\mathbb{A}$. Using M^w_{roi}, we find a set of pixels

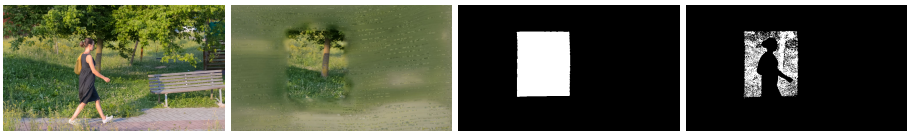

Fig. 6: (b), which is rendered by $\mathbb{L}(\mathbb{T}(p), t)\mathbb{A}(\mathbb{T}(p))$, shows an appearance without the moving foreground object that exists in the input frame at t. (d) Using this property, we can effectively identify the target points, $\mathcal{P}_{no}$, for mask refinement.

potentially belonging to the ROI as $\mathcal{P}_{pot} = \{p | M^{w}_{roi}(\mathbb{T}(p)) = 1\}$ (Fig. 6 (c)). We then identify non-occluded pixels among the pixels in $\mathcal{P}_{pot}$. Specifically, we find a set of non-occluded pixels $\mathcal{P}_{no}$ as $\mathcal{P}_{no} = \{p | \|\mathbb{L}(\mathbb{T}(p), t)\mathbb{A}(\mathbb{T}(p)) - c_p\|_1 < \tau\} \cap \mathcal{P}_{pot}$ where τ is a small constant. Fig. 6 (d) visualizes an example of $\mathcal{P}_{no}$.

The next stage of the refinement step updates the mask $\mathbb{M}$ by minimizing the loss function defined as:

$$\mathcal{L}_{refine} = \mathcal{L}_{recon} + \mathcal{L}_{no}, \tag{9}$$

where $\mathcal{L}_{no}$ is a loss for the non-occluded pixels. $\mathcal{L}_{no}$ is defined as:

$$\mathcal{L}_{no} = \lambda^{(1)}_{no} \sum_{p \in \mathcal{P}_{no}} |1 - \mathbb{M}(p)|^2 + \lambda^{(2)}_{no} \sum_{(p, p_a) \in \mathcal{P}_a} |\mathbb{M}(p) - \mathbb{M}(p_a)|\,, \tag{10}$$

where $\lambda^{(1)}_{no}$ and $\lambda^{(2)}_{no}$ are balancing weights. The first term on the right hand side promotes $\mathbb{M}$ to be close to 1 for the non-occluded pixels inside the ROI while the second term propagates the refined mask values to other frames. We update only $\mathbb{M}$ using $\mathcal{L}_{refine}$ in the mask refinement step.

3.4 Video Reconstruction using a Soft Neural Atlas Model

Although our mask refinement step can effectively enhance the accuracy of the mask $\mathbb{M}$, the mask after the refinement tends to be close to a hard mask whose values are either 0 or 1. On the other hand, pixels along the boundaries between occluding objects and the ROI usually have mixtures of the colors from different regions. Thus, using a hard mask leads to unnatural reconstruction results after atlas editing as indicated by the orange arrow in Fig. 4 (d).

For more visually pleasing blending, we thus estimate a soft mask that correctly reflects the blending of occluding objects and the ROI using an off-the-shelf matting network, VitMatte [18]. Specifically, for each video frame, we render its mask from $\mathbb{M}$, and apply morphological erosion and dilation to obtain a trimap. Then, we estimate a soft mask $\hat{\mathbb{M}}$ by feeding the frame and its trimap to the matting network.

For video reconstruction after atlas editing, we may naïvely replace $\mathbb{A}$ and $\mathbb{M}$ in Eq. (1) with $\mathbb{A}_{edit}$ and $\hat{\mathbb{M}}$, respectively. However, this approach does not produce natural-looking results on the boundary pixels around occluding objects

Fig. 7: Qualitative comparisons with previous methods. RNA achieves natural editing by handling complex occlusions without artifacts, such as ghosting and omission, significantly outperforming previous methods.

since Eq. (1) is designed in the consideration of using a hard mask. To address this, we additionally propose a soft neural atlas model, which is defined as:

$$c_p = \hat{\mathbb{M}}(p)\mathbb{L}(\mathbb{T}(p), t)\mathbb{A}(\mathbb{T}(p)) + (1 - \hat{\mathbb{M}}(p))c_p^{oc}, \tag{11}$$

where c_p^{oc} is the color of the occluding object at p. From Eq. (11), we can derive c_p^{oc} as:

$$c_p^{oc} = \frac{c_p - \hat{\mathbb{M}}(p)\mathbb{L}(\mathbb{T}(p), t)\mathbb{A}(\mathbb{T}(p))}{1 - \hat{\mathbb{M}}(p)}. \tag{12}$$

Then, by replacing $\mathbb{A}$ in Eq. (11) and substituting Eq. (12) into Eq. (11), we obtain our final reconstruction equation, which is defined as:

$$c_p^{edit} = \begin{cases} \hat{\mathbb{M}}(p)\mathbb{L}(\mathbb{T}(p), t)(\mathbb{A}_{edit}(\mathbb{T}(p)) - \mathbb{A}(\mathbb{T}(p))) + c_p & \hat{\mathbb{M}}(p) < 1 \\ \mathbb{L}(\mathbb{T}(p), t)\mathbb{A}_{edit}(\mathbb{T}(p)) & \hat{\mathbb{M}}(p) = 1, \end{cases} \tag{13}$$

where c^{edit} is a video reconstructed from $\mathbb{A}_{edit}$. The equation for $\hat{\mathbb{M}}(p) < 1$ in Eq. (13) should reduce to $\mathbb{L}(\mathbb{T}(p), t)\mathbb{A}_{edit}(\mathbb{T}(p))$ when $\hat{\mathbb{M}}(p) = 1$ if $\mathbb{A}$ and $\mathbb{L}$ are perfectly estimated, i.e., $\mathbb{L}(\mathbb{T}(p), t)\mathbb{A}(\mathbb{T}(p)) = c_p$. However, $\mathbb{A}$ and $\mathbb{L}$ may have a small amount of error in practice, so we use $\mathbb{L}(\mathbb{T}(p), t)\mathbb{A}_{edit}(\mathbb{T}(p))$ for $\hat{\mathbb{M}}(p) = 1$.

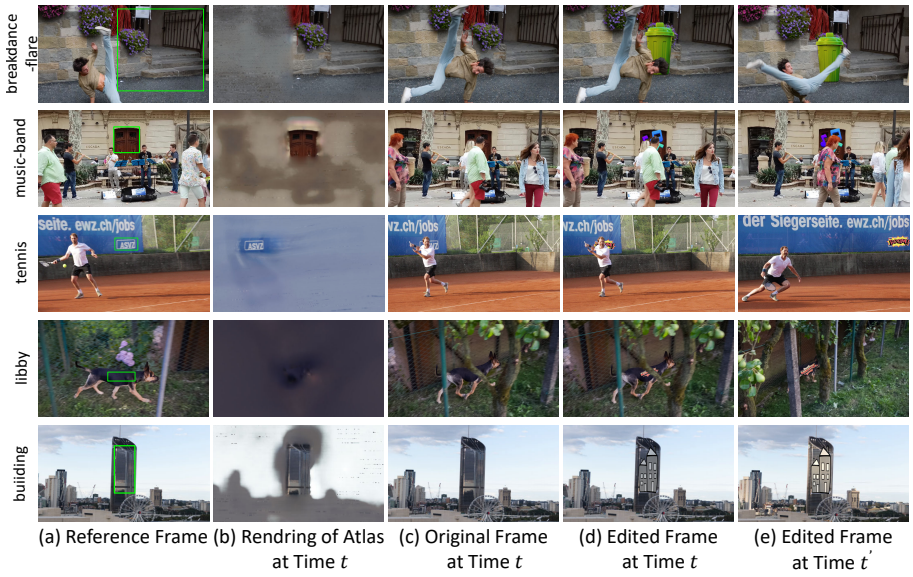

Fig. 8: Editing results of RNA applied to various video scenarios, including complex foreground objects (breakdance-flare, music-band), large camera movements (tennis), foreground object with occlusion (libby) and time-varying light change (building).

Table 1: Quantitative comparisons of memory usage, training time, and PSNR for reconstructed frame. The training is conducted using a GeForce RTX 3090 GPU with 24 GB. Deformable Sprites [19] is tested at a lower resolution due to memory constraints.

Method	Resolution	lucia [1 object, 70 frames]			dogs-jump [3 objects, 65 frames]			dog-gooses [5 objects, 70 frames]		
		GPU Memory	Training Time	PSNR	GPU Memory	Training Time	PSNR	GPU Memory	Training Time	PSNR
Deformable Sprites [19]	427 × 240	9.4 GB	25 min.	25.6	15.1 GB	35 min.	30.5	21.2 GB	55 min.	25.3
CoDeF [13]	768 × 432	3.8 GB	10 min.	26.2	3.8 GB	10 min.	33.8	3.8 GB	10 min.	25.1
LNA [9]	768 × 432	3.1 GB	6 hours	31.4	4.7 GB	10 hours	33.8	6.3 GB	14 hours	27.6
Hashing NVD [4]	768 × 432	2.9 GB	1.2 hours	31.0	3.7 GB	2.2 hours	33.2	4.4 GB	3.2 hours	27.3
Ours	768 × 432	2.4 GB	40 min.	31.5	2.4 GB	40 min.	33.5	2.4 GB	40 min.	27.7

4 Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of RNA. We use video examples from the DAVIS dataset [2] for the evaluations. We refer the readers to the Supplemental Document for implementation details including the network architecture of the mappings, the balancing weights of the loss function, and the details of the editing process.

4.1 Video Editing Quality

We first compare the quality of video editing results of different methods. Fig. 7 shows a qualitative comparison of the editing results of Hashing NVD [4], LNA [9] and CoDeF [13]. Hashing NVD [4] and LNA [9] exhibit ghosting artifacts in all

editing results due to the inaccurate atlas estimation of the foreground objects. CoDeF [13] exhibits several issues across the editing examples. In the 'schoolgirls' example, the head of the girl in violet disappears in the edited frames, because CoDeF relies on instance segmentation to extract foreground objects, which is unfortunately unreliable for objects with complex motions. In the 'dog-gooses' example, the edited contents show jittering artifacts as the camera moves due to its relatively naïve motion estimation. In the 'dogs-jump' example, boundary artifacts can be observed between the dog and the background due to its inaccurate segmentation and hard mask-based approach. In contrast, RNA achieves natural video editing results in these challenging scenarios.

4.2 Reconstruction Quality and Efficiency

For plausible video editing, accurate atlas estimation is crucial. To evaluate the quality of atlas estimation of RNA, we compare video reconstruction results of different methods on video examples with various numbers of moving objects in Tab. 1. Additionally, we also compare the computation times to evaluate the efficiency of the proposed method. In the table, we compare the GPU memory usage and the training times needed for atlas and mask estimation with those of previous methods. For a fair comparison, we compare the PSNR value within a specified ROI region between the reference and reconstructed frames. The details of these ROI regions are included in the Supplementary Document. As described in the table, Deformable Sprites [19], LNA [9], and Hashing NVD [4] require memory and training times that proportionally increase with the number of foreground objects, as they model each foreground object using an individual network. Meanwhile, RNA achieves comparable PSNR values to these methods with generally smaller and constant computational overload, regardless of the number of moving objects. CoDeF [13] exceptionally requires constant and small computational overload, similar to our approach, since it models only a single atlas for the background region. However, it exhibits poor PSNR values when there are camera motions, as shown in the 'lucia' and 'dog-gooses' examples. This poor reconstruction results in jittering artifacts after editing, as shown in the results of 'dog-gooses' and 'dogs-jump' in Fig. 7 (e).

4.3 Additional Qualitative Examples

Fig. 8 shows rendering results of estimated atlases and video editing results applied to various scenarios. As shown in (b) and (c) in the figure, the rendering results of atlases exhibit the contents without occluding foreground objects. We effectively utilize this property in our mask refinement and soft neural atlas model, achieving high-quality editing results. (d) and (e) in the figure demonstrate high-quality video editing results achieved by RNA. Especially, despite the high complexity with many people walking around in the 'music-band' example, RNA effectively handles such challenging occlusions, achieving natural video editing results. In the 'tennis' example, RNA succeeds in editing despite large camera movements. The 'libby' example shows that RNA successfully edits the

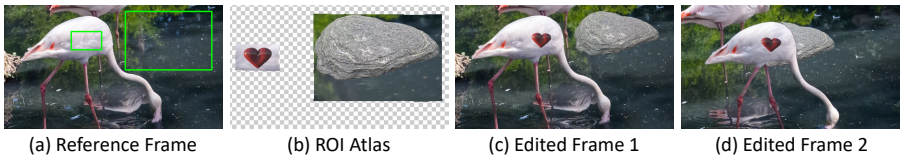

Fig. 9: (a) shows multiple ROIs designated for a moving foreground object and a static background, and (b) shows an estimated atlas corresponding to those ROIs. (c) and (d) show that RNA can address the editing in both the foreground object and the background using a single atlas.

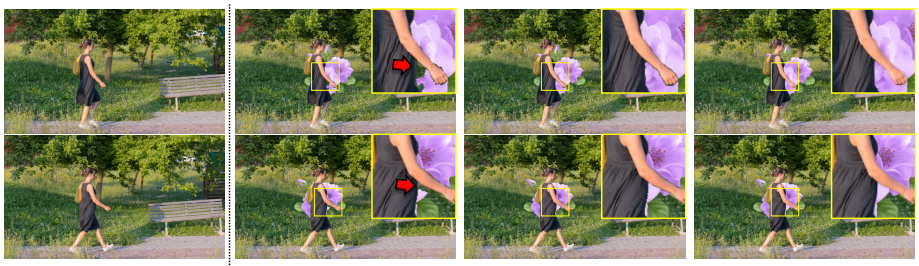

Fig. 10: Ablation study for our altas estimation, mask refinement, and soft neural atlas model. (b) Atlas estimation produces artifacts for pixels around the boundaries of occluding objects (red arrows). (c) Mask refinement successfully addresses those artifacts. (d) Soft neural atlas model achieves smooth compositions between the editing area and the foreground.

moving foreground object despite its non-rigid motion, and handles occlusions despite the motion of the foreground object. The 'building' example demonstrates that RNA can also effectively handle time-varying illumination changes.

Multiple ROIs RNA also allows for a user to simultaneously edit both a moving foreground object and the background as long as they are not overlapped in the reference frame. For editing multiple ROIs, a user specifies multiple ROIs, as shown in Fig. 9. Then, RNA estimates a single atlas for multiple ROIs as shown in Fig. 9 (b) so that the user can edit the ROIs together.

4.4 Ablation Study

Impact of each phase We conduct an ablation study by sequentially applying our mask refinement and soft neural atlas model after atlas estimation (Fig. 10). Without the mask refinement phase, an imprecise mask is estimated, leading to artifacts for pixels around the boundaries of occluding objects (red arrows in Fig. 10 (b)). Our mask refinement effectively suppresses these artifacts. Finally, our soft neural atlas model smoothly blends the edited content and moving

Fig. 11: Comparison of (c) a naïve reconstruction result, which uses Eq. (1) by replacing $\mathbb{M}$ and $\mathbb{A}$ with $\hat{\mathbb{M}}$ and $\mathbb{A}_{edit}$, and (d) a result using our soft neural atlas model. In the boundary areas between the edited content and the occluding foreground object, our proposed model shows a more natural-looking transition between them.

foreground object, as shown in Fig. 10 (d). More ablation examples are displayed to the Supplemental Document.

Soft neural atlas model We conduct another ablation study to verify the effect of our soft neural atlas model. A naïve approach to video reconstruction is to use the hard mask-based neural atlas model presented in Eq. (1). Specifically, we may simply replace $\mathbb{A}$ and $\mathbb{M}$ with $\mathbb{A}_{edit}$ and $\hat{\mathbb{M}}$, respectively, to achieve video reconstruction using an edited atlas. In this ablation study, we compare our soft neural atlas model-based video reconstruction against the naïve approach to verify the effect of the soft neural atlas model in Fig. 11. Fig. 11 (c) shows a result of the naïve reconstruction approach. As shown in the magnified view in Fig. 11 (c), the boundary between the leg and green bin includes unnatural dark colors, which originate from the original input frame. In contrast, our soft neural atlas model successfully avoids such artifacts and achieves highly-natural reconstruction results, as shown in Fig. 11 (d).

5 Conclusions

In this paper, we propose RNA, a novel ROI-based video editing framework. RNA enables video editing by allowing users to specify an ROI that they want to edit. Our ROI-based approach enables computationally-efficient and robust atlas and mask estimation, while removing the burden of users to manually specify all moving foreground objects. For high-quality video editing, we also present a novel mask refinement method, and a novel soft neural atlas model. Consequently, RNA offers a more practical and efficient solution to video editing that is applicable to a wider range of videos. Our method is not free from limitations. we assume that the ROI a user wants to edit must be clearly visible without any occluding objects in a reference frame. However, such frames may not be available in some videos.

Acknowledgements This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No.2019-0-01906, Artificial Intelligence Graduate School Program(POSTECH), RS-2024-00395401, Development of VFX creation and combination using generative AI).

References

1. Bar-Tal, O., Ofri-Amar, D., Fridman, R., Kasten, Y., Dekel, T.: Text2live: Text-driven layered image and video editing. In: European conference on computer vision. pp. 707–723. Springer (2022)
2. Caelles, S., Pont-Tuset, J., Perazzi, F., Montes, A., Maninis, K.K., Van Gool, L.: The 2019 davis challenge on vos: Unsupervised multi-object segmentation. arXiv:1905.00737 (2019)
3. Chai, W., Guo, X., Wang, G., Lu, Y.: Stablevideo: Text-driven consistency-aware diffusion video editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23040–23050 (2023)
4. Chan, C.H., Yuan, C.Y., Sun, C., Chen, H.T.: Hashing neural video decomposition with multiplicative residuals in space-time. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7743–7753 (2023)
5. Cheng, Y., Li, L., Xu, Y., Li, X., Yang, Z., Wang, W., Yang, Y.: Segment and track anything. arXiv preprint arXiv:2305.06558 (2023)
6. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
7. Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial intelligence **17**(1-3), 185–203 (1981)
8. Jampani, V., Gadde, R., Gehler, P.V.: Video propagation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 451–461 (2017)
9. Kasten, Y., Ofri, D., Wang, O., Dekel, T.: Layered neural atlases for consistent video editing. ACM Transactions on Graphics (TOG) **40**(6), 1–12 (2021)
10. Meyer, S., Sorkine-Hornung, A., Gross, M.: Phase-based modification transfer for video. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. pp. 633–648. Springer (2016)
11. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM transactions on graphics (TOG) **41**(4), 1–15 (2022)
12. Oh, S.W., Lee, J.Y., Sunkavalli, K., Kim, S.J.: Fast video object segmentation by reference-guided mask propagation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7376–7385 (2018)
13. Ouyang, H., Wang, Q., Xiao, Y., Bai, Q., Zhang, J., Zheng, K., Zhou, X., Chen, Q., Shen, Y.: Codef: Content deformation fields for temporally consistent video processing. arXiv preprint arXiv:2308.07926 (2023)
14. Rav-Acha, A., Kohli, P., Rother, C., Fitzgibbon, A.: Unwrap mosaics: A new representation for video editing. In: SIGGRAPH '08 ACM SIGGRAPH 2008 papers. ACM (August 2008), `https://www.microsoft.com/en-us/research/publication/unwrap-mosaics-new-representation-video-editing/`
15. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 402–419. Springer (2020)
16. Texler, O., Futschik, D., Kučera, M., Jamriška, O., Sochorová, Š., Chai, M., Tulyakov, S., Sỳkora, D.: Interactive video stylization using few-shot patch-based training. ACM Transactions on Graphics (TOG) **39**(4), 73–1 (2020)
17. Wang, X., Jabri, A., Efros, A.A.: Learning correspondence from the cycle-consistency of time. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2566–2576 (2019)

18. Yao, J., Wang, X., Yang, S., Wang, B.: Vitmatte: Boosting image matting with pre-trained plain vision transformers. Information Fusion **103**, 102091 (2024)
19. Ye, V., Li, Z., Tucker, R., Kanazawa, A., Snavely, N.: Deformable sprites for unsupervised video decomposition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2657–2666 (2022)