Structure-aware Video Style Transfer with Map Art

THI-NGOC-HANH LE, YA-HSUAN CHEN, and TONG-YEE LEE, National Cheng-Kung University, Taiwan, Republic of China

Changing the style of an image/video while preserving its content is a crucial criterion to access a new neural style transfer algorithm. However, it is very challenging to transfer a new map art style to a certain video in which "content" comprises a map background and animation objects. In this article, we present a novel comprehensive system that solves the problems in transferring map art style in such video. Our system takes as input an arbitrary video, a map image, and an off-the-shelf map art image. It then generates an artistic video without damaging the functionality of the map and the consistency in details. To solve this challenge, we propose a novel network, *Map Art Video Network* (MAViNet), the tailored objective functions, and a rich training set with rich animation contents and different map structures. We have evaluated our method on various challenging cases and many comparisons with those of the related works. Our method substantially outperforms state-of-the-art methods in terms of visual quality and meets the mentioned criteria in this research domain.

CCS Concepts: • **Computing methodologies** → **Image manipulation;**

Additional Key Words and Phrases: Style transfer video, coherence, map art, CNN, MAViNet

ACM Reference format:
Thi-Ngoc-Hanh Le, Ya-Hsuan Chen, and Tong-Yee Lee. 2023. Structure-aware Video Style Transfer with Map Art. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 3s, Article 131 (February 2023), 25 pages.
https://doi.org/10.1145/3572030

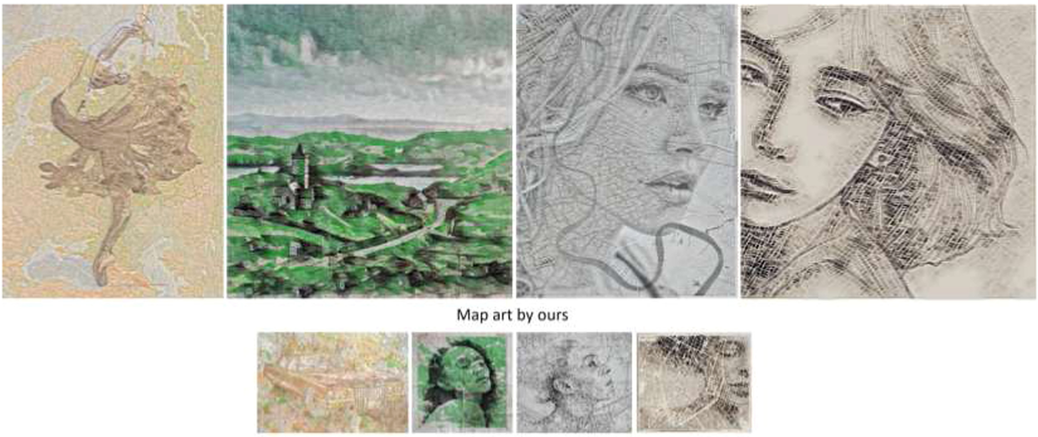
1 INTRODUCTION

Map art (MArt) is a masterpiece in which the artist integrates human portrait and topography to make it appear as though the two have always belonged together. This modern artwork was created by artist Ed Fairburn [10]. The artist uses paper maps as canvases for incredibly detailed portraits, rendering human features as topographical landscapes on top of street maps, star charts, railroad blueprints, and other maps. Fairburn described the feeling when he was creating: "I am to preserve the functionality of each map by feeding the composition instead of fighting it" [11]. To create such a MArt, an artist's experience plays a big part, and the artist can take a couple of days or even a couple of months, depending on size and complexity. Automatic computational strategies for MArt can reduce the training and production burdens and make MArt readily available to the general public. Especially, creating a **Map Art Video (MAVi)** should be an ill-posed handcraft process. Thus, automatically producing a MAVi could be an interesting research field and

This work was supported in part by the National Science and Technology Council (under Nos. 111-2221-E-006-112-MY3 and 110-2221-E-006-135-MY3), Taiwan, Republic of China.

Authors' address: T.-N.-H. Le, Y.-H. Chen, and T.-Y. Lee, National Cheng-Kung University, Tainan, Taiwan, Republic of China; emails: ngochanh.le1987@gmail.com, arielmiyabi@gmail.com, tonylee@mail.ncku.edu.tw.

1551-6857/2023/02-ART131 $15.00
https://doi.org/10.1145/3572030



Map art by ours

Map art by artists

Fig. 1. Our proposed model can transfer the map art styles from artists on various contents.

a potential industry product. Moreover, style transfer has been applied in several ways in the era of AI creation, such as photo and video editors, commercial art, gaming, and virtual reality. However, the existing style transfer methods are limited in oil paintings. Our current work shapes the applications of style transfer by extending to MArt style, which is mentioned as special modern artwork. This could be potential to enable people to obtain more predictable results and increase the diversity of user experience. As we exhibit in Figure 1, our proposed model can transfer the map art styles from artists on various contents.

Inspired by the above motivation, we develop a system to transfer MArt styles to videos. We focus on stylizing the input videos which encompass of the map in the background. We aim to generate MAVi with three criteria: (1) the style should resemble that of the style in the given MArt, (2) the map functionality/structure (e.g., text, tiny road, line structure) should be preserved as well as the portrait (i.e., animating object) is not damaged, and (3) be coherent in both temporary and map structure.

An early study in MArt generation is a multi-stage framework introduced by Shih et al. [43]. First, their scheme bases on feature maps to extract the portrait from the given MArt to obtain a "clean-MArt" (i.e., a MArt without portrait). An input portrait and the clean-MArt are optimized by Gram matrix [14] to generate a coarse-target MArt. Then, in the second stage, the coarse-target MArt and an input map are fed to a second-round optimization to enhance the texture and color in the fine-target MArt. Although their system is successful in some images, our preliminary experiments showed that it is nontrivial to extend it to MAVi due to the following reasons:

- A portrait extraction manner is conducted in advance to analyze the information of the input MArt, which is selected from the feature maps through a similarity measurement. However, this technique is straightforward and not efficient in the cases that the input MArt is complicated.
- Their system needs to further use some functions to harmonize the stylized results, such as: a map, which is the former background of the input content image, is used in the refinement stage as an extra input or the generated MArt needs to be color-converted to match with the color of the input style.
- Although Shih et al. [43] show a few smooth video results despite being trained on only still imagery, we found that without additional temporal regularization, there is noticeable flicker.

Existing learning-based methods in style transfer [12, 21, 24] may also be adopted, but the results are usually unsatisfactory due to the lack of consideration of the fundamental preservation of map functionality. This is particularly true for the map with prominent fundamental characteristics or sensitive to structure distortion, and the results are not entirely satisfactory.

In this article, we present a comprehensive system to generate **MAVi** automatically. Our system takes as input an arbitrary video, a map image, and an off-the-shelf MArt image. Output is a smooth, temporally consistent, and artistic video. We demonstrate that by properly designing the network structure, adequate objective functions, and providing sufficient training data for training the CNN, we can successfully transfer MArt style to the video and efficiently preserve the functionality of the map and structural consistency in MAVi results.

To achieve that, we propose a novel network called ***Map Art Video Network*** (**MAViNet**) that is especially useful for preserving the functionality and rich detail of the map in the results. To obtain good results, we also design tailored objective functions. Besides, we develop a rich training set containing many video frames, rich animation content, and different map structures. To validate the effectiveness of our method, we test our method with a wide variety of challenge cases. Realistic and appealing results are obtained. We also compare our results with related methods to demonstrate our capability in handling style transfer on video. In addition, we further show how to use the benefits from our scheme for sketch stylization and ancient movie production. In summary, our technical contributions are as follows:

- We propose an efficient model for MAVi generation, *MAViNet*, without damaging the rich detail of the map and preserve the consistency in map structure.
- We design objective functions that are tailored to this research domain.
- The proposed framework can shape MAVi generation of arbitrary input videos.

2 RELATED WORK

The advances in artificial intelligence technologies give new opportunities to make masterpieces of art available to digitize [2, 38]. There are plenty of studies and techniques exploring how to automatically turn images into synthetic artworks. Among these studies, the advances in **non-photorealistic rendering (NPR)** [16, 39, 46] are inspiring, and nowadays, it is a firmly established field in the community of computer graphics. However, most of these NPR stylization algorithms are designed for particular artistic styles [15, 39] and cannot be easily extended to other styles. Although these algorithms are capable of faithfully depicting certain prescribed styles, they typically have the limitations in flexibility, style diversity, and effective image structure extractions. Therefore, there is a demand for novel algorithms to address these limitations, which gives birth to the field of **neural style transfer (NST)**.

Style transfer is a technique that aims at rendering images with desired artistic styles without annotated data. The first NST algorithm, proposed by Gatys et al. [15], addresses the limitations of previous NPR algorithms without CNNs. This seminal work showed that **deep neural networks (DNNs)** encode not only the content but also the style information of an image. Gatys' method is successful in changing the style of an image while preserving its content. Although this former method suffers some limitations, such as a slow scheme due to an optimization process or generally fails for photorealistic synthesis, it inspired many later image style transfer models [3, 5, 9, 24, 28, 32, 42, 47]. Mentioned as the most successful heritors of Gatys et al. [15], Johnson et al. [24] proposed a feed-forward network to control style conversion. Essentially, they only differ in the network architecture, which is designed by following the network in Radford et al. [37]. The objective function is similar to the algorithms of Gatys et al. [15]. Therefore, it makes them suffer from the same aforementioned issues as Gatys' algorithm.

With the aims to transfer arbitrary artistic styles with one single trainable model, Huang and Belongie [21] introduced an approach that combines the flexibility of the optimization-based framework [15] and the speed similar to the feedforward approaches [24, 47]. Their method is based on Dumoulin et al. [9], an early work that was built on the basic of instance normalization layer in per-style-per-model algorithm [47]. However, these algorithms are data-driven and limited on generalizing on unseen styles. Also, it is hard to synthesize complicated style patterns with rich details and local structures [23]. Some recent works [1, 7, 30] attempt to propose methods that can adaptively perform image and video stylization on arbitrary artistic styles. They all present promising models with expressive results.

Video stylization gives the additional challenge in such research domain. Different from still image style transfer, the design of video style transfer algorithm needs to consider the smooth transition between adjacent video frames. When directly applying the image style transfer techniques [15, 24, 50] to videos, the generated stylized video will inevitably be affected with severe flicking artifacts. As such, to alleviate the flicking artifacts, a number of video style transfer approaches [4, 13, 20, 27, 40] are proposed by additionally utilizing temporal constraints to ensure the temporal consistency across frames. Recently, some novel approaches are introduced to improve the robustness of video style transfer [29, 48]. For more related work, readers can see a survey in Jing et al. [23].

To challenge with map art style transfer, a multi-stage framework is introduced by Shih et al. [43], which is mentioned as a variation of Gatys et al. [15]. The method can simulate good MArt results. However, this method has some shortcomings as we discussed in previous section. Therefore, there is a demand for a novel algorithm to address these limitations.

In summary, existing works in style transfer, including still images and video algorithms, have the benefits of transferring new style to a certain image/video. However, they still suffer the common limitations of NST algorithms. Therefore, such NST algorithms are tailored to transfer new style in natural image/video. The image/video with map background is a special content since it comprises of rich details (e.g., line structure, text, pattern) and local structures. Transferring map art style in map content is a challenge that needs to be investigated.

3 OUR APPROACH

3.1 System Overview

Our system is a comprehensive process, as shown in Figure 2. The system gets as input an arbitrary video, a digital map image, and an off-the-shelf MArt image. We first adopt a segmentation method [6] to extract animating objects from the input video. This segmented video is then composited with the input map to assemble the **map-video (*m*-Vi)**, i.e., video with map background. After that, *m*-Vi and the input MArt are fed to our proposed model to transfer the style of MArt to *m*-Vi. The stylized *m*-Vi in our system, which is called MAVi, is the sequence video frames in which "the content" mirrors those from *m*-Vi, and the "style" resembles that of the given MArt.

For the purpose of transferring the style of a given MArt to *m*-Vi, we propose a network called ***Map Art Video Network*** **(MAViNet)**, which gets as input an *m*-Vi and a MArt and generates an artistic and temporally-consistent video. The overall framework of MAViNet is outlined in Figure 3. Section 3.2 describes the MAViNet architecture in detail. The model is trained by minimizing four types of loss functions described in Section 3.3.

3.2 Map Art Video Network

In designing an appropriate CNN model, we have two requirements. First, the portrait and the functionality of the map should be preserved. Second, the generated video should be temporally

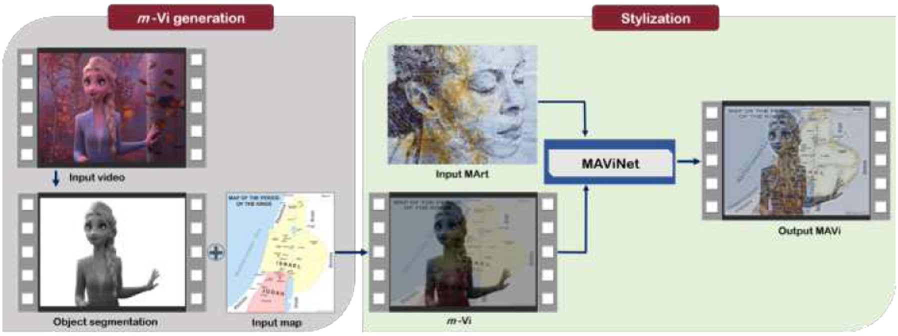


Fig. 2. The pipeline of our comprehensive system. The system takes as input an arbitrary video, a map image, and a MArt. In the *m*-Vi generation, the input video is first segmented to extract the animating objects. Segmented video and the input map are composited to assemble *m*-Vi. In the stylization manner, the *m*-Vi and MArt are fed to our MAViNet to transfer the style in the MArt to the *m*-Vi.

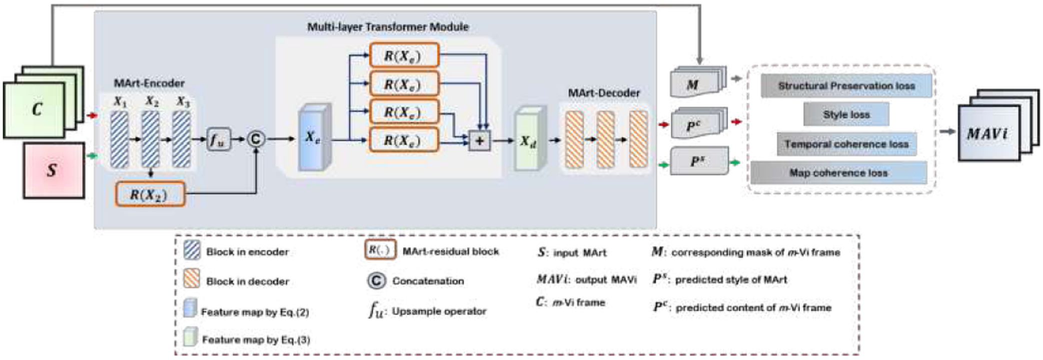
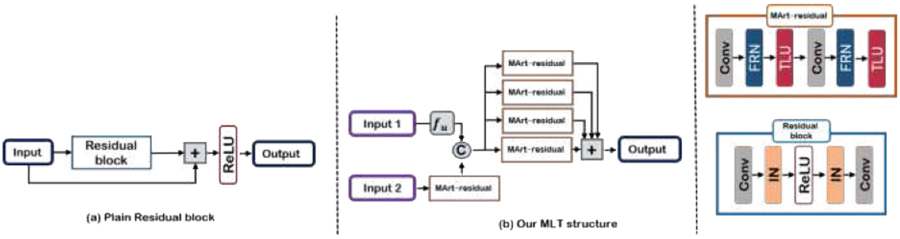


Fig. 3. The architecture of our MAViNet.

consistent and map consistent. To achieve these, we adopt a pixel-wise CNN model to build our model (MAViNet). A typical pixel-wise CNN model is composed of two parts: an encoding network (*Encoder*) and a decoding network (*Decoder*). The encoding network compresses the input into feature vectors. Meanwhile, the decoding network is built after the encoding network to reconstruct the desired output from the encoded feature vectors. However, the basic structure *Encoder-Decoder* is too "plain" and may lead to a degrading problem due to its streamlined nature, as gradient is hard to propagate from higher levels to lower levels. It may be unable to produce high-quality results when the input has complex content, such as the map patterns in our MAVi application, even with deeper levels of convolutions. To increase the depth of the network while avoiding this degrading problem, we construct our model with three parts: *MArt-Encoder, Multi-layer Transformer Module,* and *MArt-Decoder*. Our full model jointly learns to transfer style of a MArt to *m*-Vi and maintain temporal consistency. The network is designed to be fully convolutional, which can handle arbitrary size input. The overview of MAViNet is illustrated in Figure 3.

3.2.1 MArt-Encoder (MArt-E). We design the MArt-E to encode the *m*-Vi frames and MArt image to feature space. The basic encoder block used in a typical CNN model is a block consisting of a pyramid of **convolutional layers (*Conv*)** followed by a **batch normalization layer (*BN*)** [22] and a **rectified linear unit layer (*ReLU*)** [33] **(*Conv-BN-ReLU*)**. Differently, in our CNN, the convolutional layers are followed by a **Filter Response Normalization (FRN)** [45] instead of



(a) Plain Residual block

(b) Our MLT structure

Fig. 4. Plain residual block structure vs. Our **MLT** structure.

BN and a **Thresholded Linear Unit layer (TLU)** instead of ReLU (***Conv-FRN-TLU***). The reason is that FRN performs normalization operation at channel level and does not subtract the average value of the channel. Thus, the correlation between channels can be maintained. In terms of TLU, the threshold in this activation function is also a parameter that can be learned. Therefore, it can effectively balance the instability in network training to reduce the water droplet-like artifacts [25].

Given an m-Vi frame C and a MArt image $\mathcal{S}$, we can extract the corresponding set of feature maps after each MArt-E block. We denote them as $\Phi^c = \{\mathcal{X}_i^c\}$, $\Phi^s = \{\mathcal{X}_i^s\}$, respectively ($i = 1 \ldots 3$, i.e., three blocks in MArt-E). In the following parts, for short, we use notation $\mathcal{X}_l^j$ to stand for the feature maps of a certain layer in MArt-E ($j \in \{\Phi^c, \Phi^s\}$).

3.2.2 *Multi-layer Transformer Module.*

Preserving functionality in the input m-Vi frames is a challenge in this research domain, since such frames comprise either specific attributes of the map (e.g., line structure, text, pattern) or the portrait of animating objects. A naïve style transfer model succeeds in stylizing an ordinary content image but it may fail in such m-Vi content. The reason is that the extracted features are not sufficient to reconstruct the intricate details of the map. Taking this issue into account, we propose a module, called ***Multi-layer Transformer*** (**MLT**), to deepen our network and enhance the extracted feature maps to overcome the aforementioned challenge.

Conventionally, a pure architecture simply adds convolution layers to deepen the network to obtain deep features from different resolutions. However, this strategy makes the gradient disappear, and the output images will become gradually blurry. He et al. [18] showed that residual networks are easier to optimize and can gain accuracy from considerably increased depth. Inspired by this, we design our MLT module consisting of five blocks called *MArt-residual*, which takes as input feature maps from two distinct layers in our MArt-E (Figure 4(b)) rather than one single layer [18, 24] (Figure 4(a)). This can help in understanding the semantic-level, which is required to preserve in the stylized results. Each MArt-residual consists of two convolution layers with FRN [45] and TLU activation function. Hereafter, we describe the procedure to obtain features via our MLT module.

Once the m-Vi frames and MArt image are encoded by our MArt-E, feature maps of the later blocks ($\mathcal{X}_2^j$ and $\mathcal{X}_3^j$) are fed to MLT module. Note that we only use two later blocks in Φ^c and Φ^s in this manner, since they capture deeper features of input frames than the first layer. Thus, their features are adequately meaningful to learn. First, we feed $\mathcal{X}_2^j$ to a MArt-residual and generate new feature maps ($\mathcal{X}_u^j$):

$$\mathcal{X}_u^j = \mathcal{X}_2^j + \mathcal{R}\left(\mathcal{X}_2^j\right), \tag{1}$$

where $\mathcal{R}(.)$ is MArt-residual. Meanwhile, $\mathcal{X}_3^j$ is operated by a upsample operator. Subsequently, they are concatenated with $\mathcal{X}_u^j$ to yield the feature maps, denoted by $\mathcal{X}_e^j$, as the followed formula:

$$\mathcal{X}_e^j = f_c\left(\mathcal{X}_u^j, f_u(\mathcal{X}_3^j, k)\right), \tag{2}$$

where $f_c(.)$ is the concatenation, $f_u(.)$ denotes the up-sample operator with filter $k = 2$. Thereafter, we obtain the enhanced feature maps of MLT module by formulating as:

$$\mathcal{X}_d^j = \mathcal{X}_e^j + \sum_{\eta=1}^{N_r} \mathcal{R}_\eta\left(\mathcal{X}_e^j\right), \tag{3}$$

where $N_r = 4$ is the number of MArt-residual used this manner, $\mathcal{R}_\eta$ is referred to a MArt-residual. The efficiency of the enhanced feature maps obtained from our MLT module is visualized in Figure 10. Obviously, by adapting MLT module, the stylized result preserves both map attributes and portrait details.

3.2.3 MArt-Decoder (MArt-D). The goal of MArt-D is to decode feature maps to stylized frames. Because using deconvolution layers to decode the feature maps eventually produces "checkerboard" effects [34] by the backward pass of convolution layers, we design the MArt-D with three upsample layers (nearest neighbor interpolation) followed with a convolutional layer for the upsampling. As in the discussion above, we do not use Instance Normalization and Relu activation function as designed in such a classic deconvolution layer, but instead adopting FRN and TLU after each layer in our design. Our MArt-D has three blocks, and each of them performs the ***UpSample-Conv-FRN-TLU*** structure. This design offers a crucial benefit to preserve map functionality and rich detail of *m*-Vi frames.

Given an input set of feature maps $\mathcal{X} \in \mathbb{R}^{C_{in} \times H \times W}$ where H, W, and C_{in} is the height, width, and the number of input feature maps. The output feature maps yielded by a certain upsampling layer is defined as:

$$\mathbf{Z} = H_u(\mathcal{X}; \Psi), \tag{4}$$

where Ψ is the bias and $H_u(.)$ is the upsampling function. The output feature maps of decoder $\mathbf{Z} \in \mathbb{R}^{C_{out} \times H \times W}$, where C_{out} is the number of output feature maps, is then fed to final block, consisting of a *Conv* and *Tanh* activation function, to generate the final output.

3.3 Loss Function

We train the our MAViNet model f by solving the following objective function:

$$\min_f \left(\lambda_{SP}\mathcal{L}_{SP}(f) + \lambda_S\mathcal{L}_S(f) + \lambda_{CH}\mathcal{L}_{CH}(f) + \lambda_{TV}\mathcal{L}_{TV}\right), \tag{5}$$

where $\mathcal{L}_{SP}$ is a **structural preservation loss**, $\mathcal{L}_S$ is a **MArt style loss**, $\mathcal{L}_{CH}$ is a **coherence loss**, and $\mathcal{L}_{TV}$ is the total variance loss. $\lambda_{SP}, \lambda_S, \lambda_{CH}$, and λ_{TV} are the training weights. We present the pseudocode of our training procedure in Algorithm 1. The detail of each component function is presented below. In Section 4, we further give out the ablations to verify the necessity and the effectiveness of loss function in our model training.

3.3.1 Structural Preservation Loss. To measure high-level perceptual and semantic differences between images, the straightforward way is by calculating perceptual loss functions [24]. This concept is used in some well-known style transfer networks [12, 15, 24, 51]. However, our early experiments show that the generated results are blurry and fail in preserving features of the content, including the portrait and map patterns, when training the network with the standard perceptual loss [24]. Therefore, in this study, we design a tailored loss for MAVi application, called **structural preservation loss**, to measure the different representations. In particular, the differences between the standard content loss and our structural preservation loss are probably in three points as follows: First, our structural preservation loss jointly computes the difference on both background and foreground of input frames. Second, instead of calculating Euclidean distance between the target features and the features of the output image over multiple layers of the pre-trained VGG-19

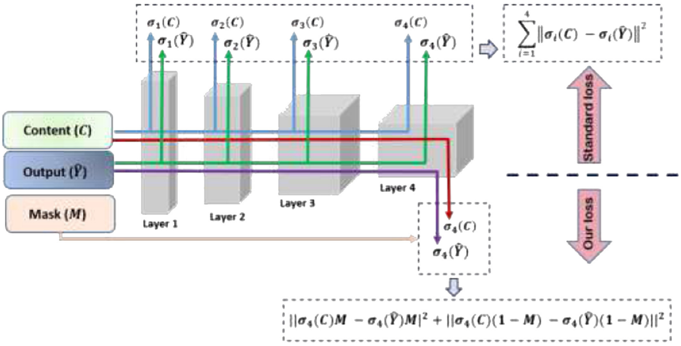


Fig. 5. Standard content loss vs. Our structural preservation loss.

[44], we only use one layer of VGG-19 [44] (as shown in Figure 5). And third, we further use the ground-truth mask of input frames in our structural preservation loss. It can help transfer the style to the foreground as strongly as possible, while the background information is better preserved.

ALGORITHM 1: Procedure of training MAViNet

```
Input: A map art style image ($\mathcal{S}$), a pair of two consecutive input frames ($C_{t-1}, C_t$), the corresponding backward flow $F_{t\Rightarrow t-1}$, and ground-truth occlusion map $\mathcal{M}_{t\Rightarrow t-1}$, mask of input frame $\mathcal{M}$.
Output: MAViNet model.
for each iteration do
    $\hat{\mathcal{Y}}_{t-1} \leftarrow$ MAViNet($C_{t-1}, \mathcal{S}$)
    $\hat{\mathcal{Y}}_t \leftarrow$ MAViNet($C_t$), $\mathcal{S}$)
    Compute $\Phi(C), \Phi(\hat{\mathcal{Y}})$        /*$\Phi(.)$ is the conv4 of VGG; $C$ stand for $C_t$ or $C_{t-1}$; $\hat{\mathcal{Y}}$ stand for $\hat{\mathcal{Y}}_{t-1}$ or $\hat{\mathcal{Y}}_t$ */
    Warp $\varphi(C_{t-1}, F_{t\Rightarrow t-1})$
    Warp $\varphi(\hat{\mathcal{Y}}_{t-1}, F_{t\Rightarrow t-1})$
    Compute gradient by losses:
    $\mathcal{L}_{SP} \leftarrow \mathcal{L}(\Phi(C, \Phi(\hat{\mathcal{Y}}), \mathcal{M})$
    $\mathcal{L}_S \leftarrow \mathcal{L}(\mathcal{G}(\Phi_i(C)), \mathcal{G}(\Phi_i(\hat{\mathcal{Y}})))$ /*$i \in [1 \ldots 5]$ */
    $\mathcal{L}_m \leftarrow \mathcal{L}(\mathcal{M}_{t\Rightarrow t-1}, F_{t\Rightarrow t-1}, C_{t-1}, C_t, \hat{\mathcal{Y}}_{t-1}, \hat{\mathcal{Y}}_t)$
    $\mathcal{L}_t \leftarrow \mathcal{L}(\mathcal{M}_{t\Rightarrow t-1}, F_{t\Rightarrow t-1}, \hat{\mathcal{Y}}_{t-1}, \hat{\mathcal{Y}}_t)$
    Update gradient.
end for
Return MAViNet model.
```

Our structural preservation loss $\mathcal{L}_{SP}$ includes two terms $\mathcal{L}_b$ and $\mathcal{L}_f$, which corresponds to the background differences and foreground differences between the input m-Vi frame C and the predicted MAVi frame $\hat{\mathcal{Y}}$. We use layer 4 in the pre-trained VGG-19 [44] as Φ to define these two terms. The reason is that features at lower layers are free from the influence of colors. But if we use very deep layers, then important features are lost and difficult to be reconstructed (see Figure 6). Especially, both of two terms are defined as the (squared, normalized) Euclidean distance of activations in layer 4 of Φ. Assume Φ has K distinct filters, and the size of response feature maps is $H \times W$. The value of the activation of k filter at position (i, j) in layer 4 can be represented by a

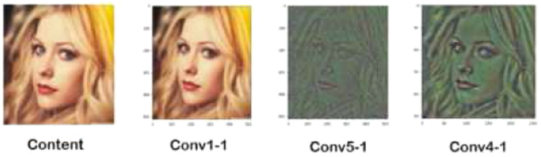


Fig. 6. Visualization of features obtained in different layers of VGG-19.

matrix:

$$\Phi_{(i,j),k} \in \mathbb{R}^{(H\times W)\times K}. \tag{6}$$

Then, two terms in the structural preservation loss are defined as

$$\mathcal{L}_b = ||\Phi(C)(1-\mathcal{M}) - \Phi(\hat{\mathcal{Y}})(1-\mathcal{M})||^2, \tag{7}$$

$$\mathcal{L}_f = ||\Phi(C)\mathcal{M} - \Phi(\hat{\mathcal{Y}})\mathcal{M}||^2, \tag{8}$$

$$\mathcal{L}_{SP} = \mathcal{L}_b + \mathcal{L}_f, \tag{9}$$

where $\mathcal{M}$ is the ground-truth mask of the input m-Vi frame C.

3.3.2 MArt Style Loss. The MArt style loss $\mathcal{L}_S$ is to minimize the style differences between generated MAVi frames and input MArt image. The standard style loss function in References [15, 43] use all the layers in pretrained VGG-19 network. If we naively apply this to train our MAViNet network, then noise-like patterns appear in the results. To tackle this issue, feature maps of five layers (L) in VGG-19 (L = {*conv1, conv2, conv3, conv4, conv5*}) are detached for different functions in our overall style loss. Gatys et al. [15] showed that the first few layers of VGG are sensitive to style of the image and later layers are sensitive to the content. Guided by this work, we use layers *conv4* and *conv5* to retain the texture of style image; meanwhile, the first three layers are computed by cross-layer Gram matrix [49] to reduce their impact on the results. Accordingly, the **MArt style loss** in our training is formulated as:

$$\mathcal{L}_S = \sum_{i=0}^{L} ||\mathcal{G}^i(\Phi_i(C)) - \mathcal{G}^i(\Phi_i(\mathcal{S}))||^2, \tag{10}$$

where $\mathcal{G}(.)$ is the Gram matrix:

$$\mathcal{G}^i(\Phi_i(.)) = \begin{cases} [\Phi_i(.)][\Phi_i(.)]^\top, \, i = \textit{conv4, conv5} \\ [\Phi_i(.)][\Phi_{i+1}(.)]^\top, \, \textit{if otherwise.} \end{cases} \tag{11}$$

3.3.3 Coherence Loss. The MAVi results in the current application must be consistent not only in temporary but also in map structure. Existing researches [4, 13, 17, 41] have found that image style transfer models are not temporally consistent. In other words, the style loss and temporal loss might not be well minimized at the same time. Therefore, image stylization models that target minimizing style loss cannot avoid flickering artifacts. Moreover, the experiments in the prior networks [12, 21], which are trained for video style transfer, indicate that if the stylization model is forced to maintain temporal consistency, then the stylization performance will degrade. Although these models are temporally smoother, the distribution of color is simpler, and the details of strokes and textures are lost.

Motivated by above reasons, we design the **coherence loss** $\mathcal{L}_{CH}$ in this study as the weighted combination of two parts:

$$\mathcal{L}_{CH} = \beta\mathcal{L}_m + \lambda\mathcal{L}_t, \tag{12}$$

where $\mathcal{L}_m$ is the map coherence loss, $\mathcal{L}_t$ is the temporal coherence loss, and β, λ is the weight of each loss, respectively.

Map coherence loss. The structural preservation loss $\mathcal{L}_{SP}$ we proposed in the above section is sufficient to preserve the map functionality on a MArt result (i.e., a single image). When generating video result, $\mathcal{L}_{SP}$ by itself does not guarantee such a clear structure of the map as well as the structure consistency between consecutive generated frames. Therefore, we define a map coherence loss $\mathcal{L}_m$ to preserve the integrity of map after transferring to a new style. This loss is defined as

$$\mathcal{L}_m = \frac{1}{H \times W} \sum_{t=2}^{T} \mathcal{M}_{t \Rightarrow t-1} (\xi_c - \xi_s)^2, \tag{13}$$

where H, W is the height and width of input/output video, T denotes the total frames, ξ_c and ξ_s is the temporal warping error of two consecutive input frames (C_{t-1}, C_t) and generated frames $(\hat{\mathcal{Y}}_{t-1}, \hat{\mathcal{Y}}_t)$, respectively. We calculate the warping error by using the estimated backward flow to warp the previous frame to the next frame. Accordingly, they are defined as follows:

$$\xi_c = C_t - \varphi(C_{t-1}, F_{t \Rightarrow t-1}), \tag{14}$$

$$\xi_s = \hat{\mathcal{Y}}_t - \varphi(\hat{\mathcal{Y}}_{t-1}, F_{t \Rightarrow t-1}), \tag{15}$$

where $F_{t \Rightarrow t-1}$ is the estimated backward flow between input frames, φ is the warping function, and $\mathcal{M}_{t \Rightarrow t-1}$ is the occlusion map:

$$\mathcal{M}_{t \Rightarrow t-1} = \begin{cases} 0, & \textit{if occlusion and motion boundary = 0} \\ 1, & \textit{otherwise.} \end{cases} \tag{16}$$

Our formulation of $\mathcal{L}_m$ encourages the structure of the generated frame at timestep t to be consistent with those in the corresponding input m-Vi frame. For instance, if there is occlusion between two consecutive frames, then using forward warping that is only based on optical flow properly fails to define the corresponding pixel in the next frame. Therefore, we use the ground-truth occlusion map to set the weight of residual of two consecutive frames to 0. This reduces the error in the calculation of the loss function. Beside, this not only helps the output to be consistent along the motion trajectories but also avoids ghosting artifacts at occlusions or motion discontinuities.

Temporal coherence loss. The temporal coherence loss is used to preserve the temporal consistency of the generated MAVi by minimizing the error between the tth and $(t-1)$th generated frame $(\hat{\mathcal{Y}}_t, \hat{\mathcal{Y}}_{t-1})$. We follow the protocol in Reference [40] that uses occlusion map $\mathcal{M}_{t \Rightarrow t-1}$ and optical flow $F_{t \Rightarrow t-1}$ to design our temporal loss function as

$$\mathcal{L}_t = \frac{1}{H \times W} \sum_{i=1}^{H \times W} \mathcal{M}_{t \Rightarrow t-1} \, ||\hat{\mathcal{Y}}_t - F_{t \Rightarrow t-1} \hat{\mathcal{Y}}_{t-1}||^2 \, . \tag{17}$$

3.3.4 Total Variation Loss. To make the predicted results smoother, we add the total variation loss function. This function calculates the square difference between adjacent pixels to achieve a smoothing affect. The formula is as follows:

$$\mathcal{L}_{TV} = \sum_{i,j} \sqrt{|y_{i+1,j} - y_{i,j}|^2 + |y_{i,j+1} - y_{i,j}|^2}. \tag{18}$$

We further add this function in our full loss as an auxiliary loss to reduce the noise on the generated frames.

4 EXPERIMENTAL RESULTS

4.1 Implementation Details

We trained our MAViNet network on our prepared training data, which consists of 450 sets of video. The detail of training data preparation is presented in the Appendix, Section A . Adam solver [26] is used with a batch size of 2 for 60,000 iterations and an initial learning rate of 3×10^{-3}. During the training of our MAViNet model, we empirically set parameters as $\lambda_{SP} = 5 \times 10^5, \lambda_S = 10^5, \lambda_{CH} = 10^{-3}$, and $\lambda_{TV} = 5 \times 10^5$ to balance the impact among the loss functions.

4.2 Our Results and Discussion

To evaluate our method, we test it on several video contents and MArt styles. The results are exhibited in Figure 7. In the content frames of these experiments, the map encompasses of tiny attributes (e.g., stroke, circle line, text) and the moving object is with challenge-to-preserve details (e.g., flying hair, the eyes, or the eyebrow). We can see that most of these attributes and details are well preserved after stylizing session. These results reveal that our proposed model is tolerant to the complex content structure when styling with different map art styles. To highlight the capability of our method, we further examine it on different kinds moving objects. For example, the character in (A) and blooming flower in (B) are sensitive to be distorted in such a style transfer process. Despite these challenges, the stylized frames appear as the style in the given art map and the content is well preserved. In the example (E), the input frame consists of multiple moving objects, and the background is the map with painting texture. By observation, our method can do such a good style transfer in this challenging case. We further verify the effectiveness of our proposed model with the example in Figure 8. In these cases, the input frames encompass rich content (e.g., multiple moving objects and significant background). We hypothesize that the background is the map and we aim to preserve it when stylizing. The results reveal that our method is also successful in this challenge.

To further highlight the effectiveness of our proposed model on general content, we test it on different content images, which are non-map-background contents. The stylized results are demonstrated in Figure 9. Here, we compare our results with a recent style transfer method MCCNet [7]. MCCNet is proposed for generating arbitrary style and on such content that we used in this experiment. The code of MCCNet is released by the authors, therefore, it is fair for this comparison. By observing the intensive results in this figure, generally, MCCNet has significant shortcomings when examining on MArt styles. The color in the style images can be transferred well, but the content is damaged. Meanwhile, our method transfers successfully MArt style on such data. More specifically, the eyes of the women (in the results of (A)), line attributes in the results of (B) and (C), and the tiny detail of the landscape photo (in the results of (D) are almost preserved well). This success makes the stylized results able to appear in such an artistic form but still keep the soul of the original image.

4.3 Ablation Study

4.3.1 The Impact of MAViNet Network. One of our core design choices is to use an Encoder-Decoder stream to construct a MArt-Encoder/Decoder in conjunction with a proposed MLT module. To validate this design, we compare our model (*MArt-E → MLT → MArt-D*) performance with those of the three naive baselines. They are (1) a plain *Encoder-Decoder*, (2) an *Image Transform Network*, and (3) *Encoder-Residual block-Decoder*. We construct all the models with a comparable number of parameters. As shown from experiments 1 to 8 in Table 1, the performance of each experiment is slightly different and at a low acceptance level. In experiment 9, we train our model with the standard perceptual loss; the score of the evaluation metric is not significantly improved. In experiment 10, the similarity metric is quite high when we construct our model with our loss

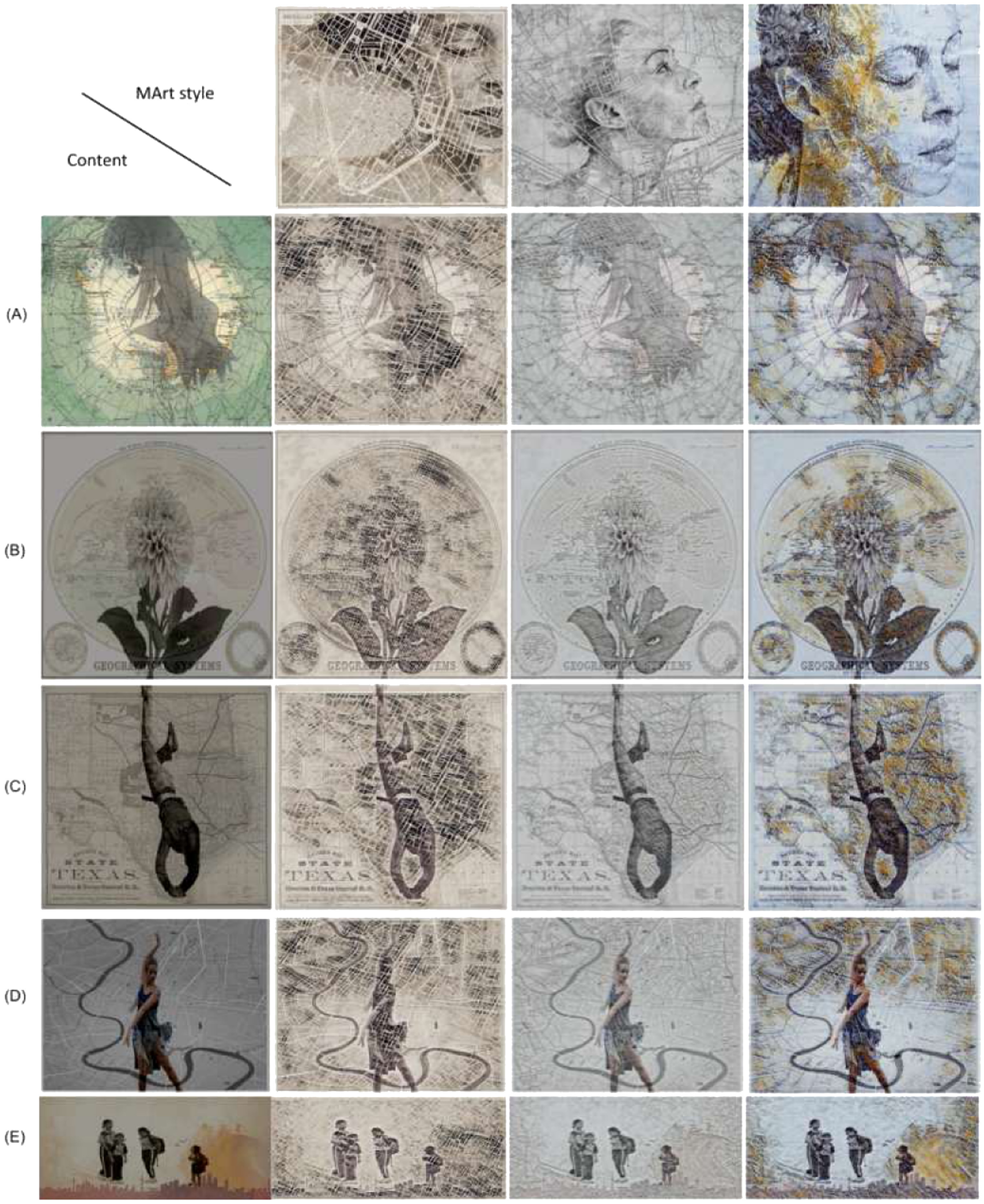


Fig. 7. Visualization of our stylized frames on different styles with different frame contents.

($\mathcal{L}_{SP}$ and $\mathcal{L}_m$). Figure 10 visualizes a sample of this experiment. As shown in this figure, a plain encoder can generate results with the color and patterns that are similar to those of the input style. However, the portrait is damaged (e.g., eyes and eyebrows of the woman) and the map information is almost lost. In contrast, the results with our MLT module have clear advantages in style and better content preservation ability. The results in Table 1 imply that making use of the off-the-shelf network structure does not always work in stylizing *m*-Vi, rather, the careful architecture is required.

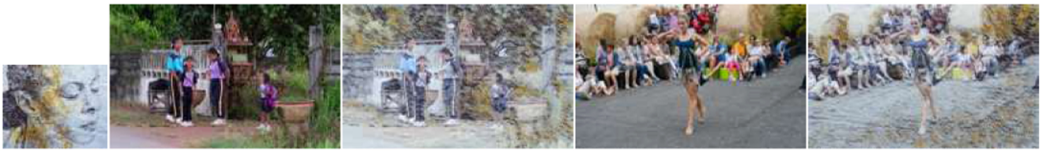


Fig. 8. In these examples, the backgrounds of the input videos are significant. We hypothesize the background as map and preserve them. Better visualization can be seen on our project website.

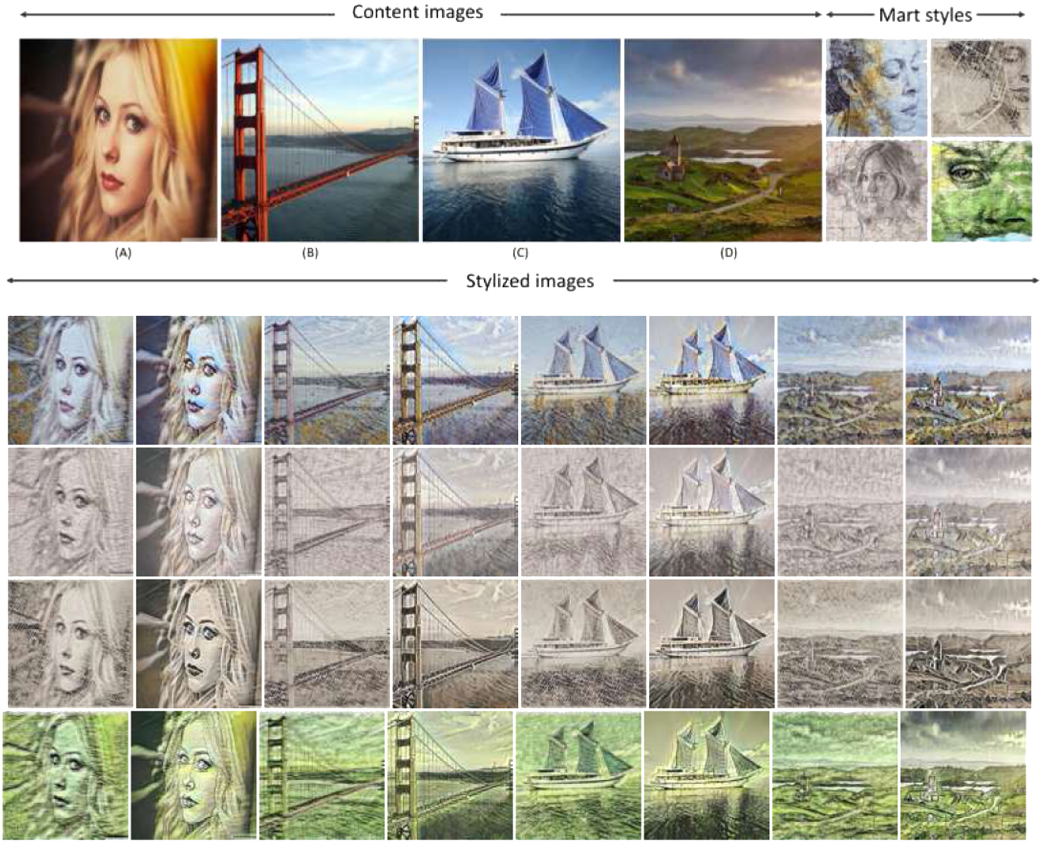


Fig. 9. Visualization of our performance on "non-map content" images. In each pair of stylized images: **left**: by our method, and **right:** by MCCNet [7].

4.3.2 The Impact of Loss Function. We verify the effectiveness of our loss functions by removing each loss term gradually from our full loss function.

(A) Structural preservation loss. We compare the generated results with our structural preservation loss and with the standard content loss to verify the effect of structural preservation loss. As shown in Figure 11, using structural preservation loss can generate results with preserving more specific attributes of the content (e.g., text of the map) compared with the stylized results with the standard content loss.

(B) MArt style loss. The MArt style loss is proposed to eliminate the noise-like patterns on the transferred results. We remove the MArt style loss in the training stage to train with the standard

Table 1. Ablation Study on Architecture Design and Loss Function

	Architecture				**Losses**			**Evaluation metric**
Experiment	*E-D*	*ITN*	*1 P-Re*	Our model	Perceptual loss	$\mathcal{L}_{SP}$	$\mathcal{L}_m$	SSIM
1	✓				✓			0.41
2	✓					✓		0.47
3	✓					✓	✓	0.53
4		✓			✓			0.32
5		✓				✓		0.46
6		✓				✓	✓	0.51
7	✓		✓		✓			0.49
8	✓		✓			✓	✓	0.37
9				✓	✓			0.54
10				✓		✓	✓	**0.76**

In these experiments, we conduct on a plain encoder-decoder (*E-D*), Image Transform Network (*ITN*) [21, 24], a single plain residual block (*1 P-Re*), and Our model. In the conjunction with loss function: Perceptual loss [24], our structural preservation loss ($\mathcal{L}_{SP}$), our map coherence loss ($\mathcal{L}_m$).

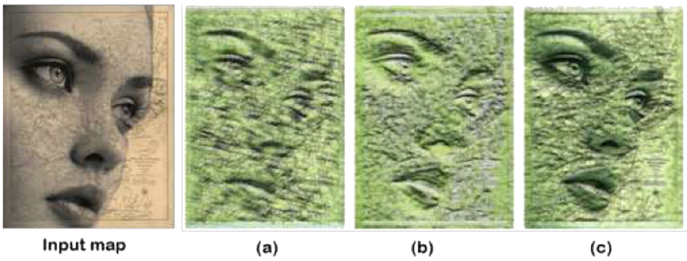


Fig. 10. (a) without the combination of different resolutions; (b) without MLT module; (c) with MLT module.

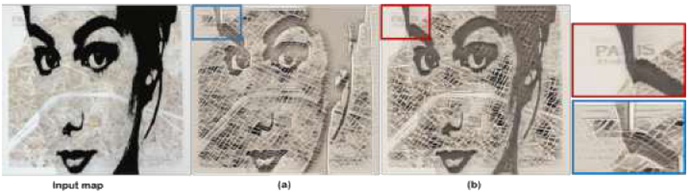


Fig. 11. (a) result trained with standard loss function [15]; (b) result trained with our structural preservation loss. The details of map, such as text, specific map patterns, are well preserved in our result.

style loss and compare the results with ours in Figure 12. Without MArt style loss, several black-square noises appear on the results, while the style patterns are well transferred by utilizing our MArt style loss.

(C) Map coherence loss. We verify the effectiveness of the map coherence loss by comparing the generated results with and without map coherence loss. The comparison is visualized in Figure 13. As observed, both results have the capability of preserving the portrait. However, there is a lack of of structure consistency in the input and the generated result when training without map coherence loss, while the results are exhibited with clean and sufficient information by map coherence loss.

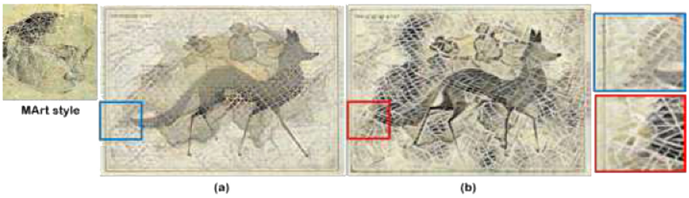


Fig. 12. (a) result trained with standard style loss [15]; (b) result trained with our MArt style loss. We highlight the difference by rectangles. With our MArt style loss, the result is quite improved by eliminating the black-square noise.

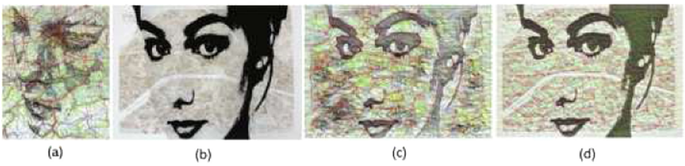


Fig. 13. (a) MArt style; (b) input content; (c) stylized result without $\mathcal{L}_m$; and (d) stylized result with $\mathcal{L}_m$. The color of (c) is closer to (a) than (d) in this style, but the map consistency with (b) is not good as (d) (e.g., the road, text, tiny regions of the map).

4.4 Visual Comparisons

To demonstrate that our method advances prior work in transferring MArt styles to video, we compare it with five methods. We visualize these comparisons in Figure 14. More comparisons can be found on our project website.[1] To the best of our knowledge, we are the first to apply deep learning for MAVi generation. Therefore, we visually compare our results to the state-of-the-art style transfer methods [1, 7, 21, 24] and a map art style transfer on a single image [43]. For a fair comparison, most of the compared methods are trained with our training dataset and the source code released by authors. Note that in this section, we compare on a single frame in terms of the ability to preserve the map functionality and stylization effects. The comparisons on the criteria of video are presented in later sections.

We first compare our method to a perceptual loss for the real-time style transfer method proposed by Johnson et al. [24] in Figure 14(a). In general, Johnson's method can preserve most of the major information of the content. Nevertheless, their results can only retain the foreground but might fail to evenly distribute the style to the entire image as our method does. In particular, some specific attributes of the map (e.g., text) are distorted and not recognized well. Besides, the transferred results are blurry, and the style is not evenly distributed. These drawbacks prevent Johnson's method from generating plausible MArt results. In contrast, our method successfully preserves attributes of map and portrait in the image and distributes color and patterns of the input style evenly. The reason is that rather than using a pure image transformation network, we propose a network to extract deep features in both content image and MArt image. The other reason is that it is not trained with our structural preservation loss and MArt style loss.

In Figure 14(b), we compare our method with that of AdaIN model [21], which is proposed to transfer arbitrary new styles in real-time video. Since AdaIN is tailored for natural images, as observed, AdaIN method generally fails to preserve the attributes of content-map images, especially when the map composes of line structures, text, and so on. This evidences the importance of

[1] Project website: http://graphics.csie.ncku.edu.tw/MArtVi.

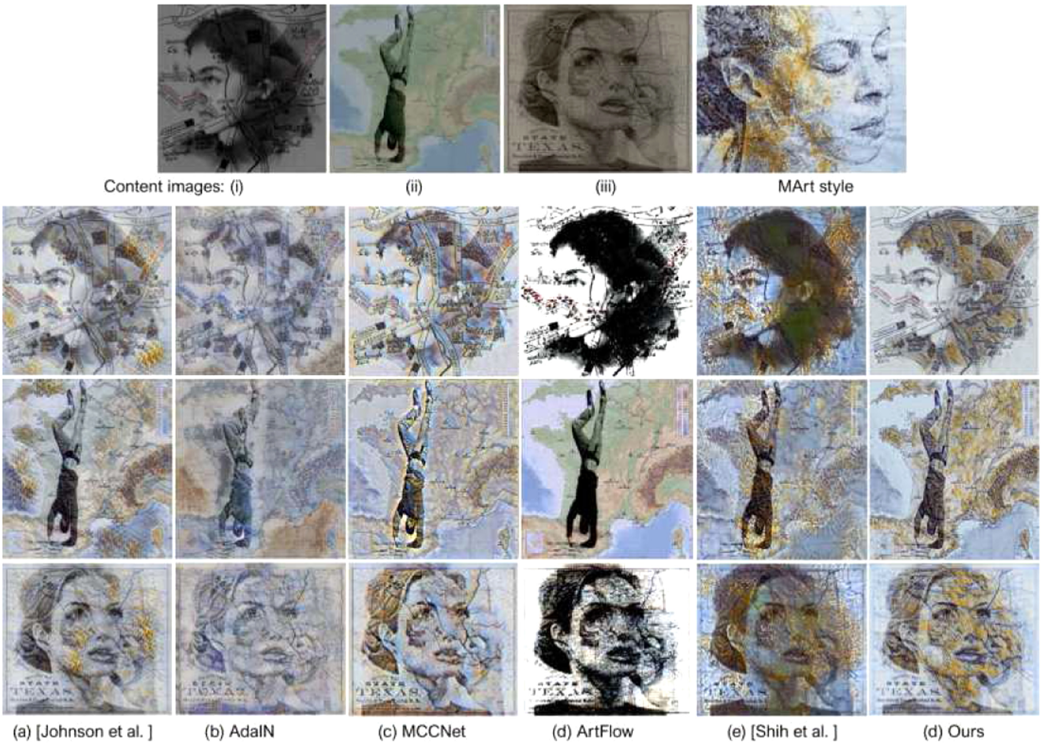


Fig. 14. Comparisons of map art style transfer on different maps among five competitors and our method.

our MAViNet design and structural preservation loss. Our method outperforms all competitors in terms of the abilities in transferring MArt style to *m*-Vi and preserving specific attributes, although the attributes in the input of this comparison are complicated.

In Figures 14(c) and (d), we visually compare our results with those in two recent works, MCCNet [7] and ArtFlow [1]. Both MCCNet and ArtFlow are designed to tackle problems in universal style transfer. We can see that MCCNet [7] can preserve some of the major attributes of the input content (e.g., line, text), but it suffers from transferring the given style to the content, i.e., color, pattern from the given MArt style. Meanwhile, the results by ArtFlow [1] are damaged significantly in the first and third rows, and deems not to be effective in the second row.

In Figure 14(e), we compare our method to a multi-stage framework in map art style transfer [43]. This compared work is mentioned as a variation of Gatys et al. [15], which is early proposed to challenge MArt generation. From this figure, we can see that the color of style in Shih et al. [43]'s results are more comparable to the input map art style than ours. However, some regions are blurry, and the content, including the portrait and map, is damaged significantly. In particular, the text and the eyes/eyebrow of the woman are damaged significantly. This may be due to the failure of the loss function of Shih's method. The other reason is that the stylized image in their system is generated by optimizing the Gram matrix between a portrait image and a style image, which is obtained by extracting its early portrait. In sharp contrast, our method fully preserves map attributes and portrait information while successfully delivering the plausible stylized result.

In summary, the compared methods share similar difficulties in preserving the portrait and the specific patterns in the content frames. Conversely, our method outperforms all competitors in the ability to transfer MArt style and preserve map functionality. This indicates that our method can challenge transferring map art style to various types of input map.

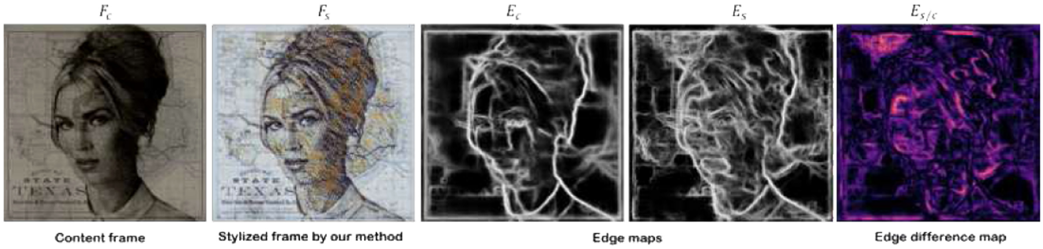


Fig. 15. Visualization of the objects used in our evaluation metrics. Here, F_c, F_s, E_c, E_s, respectively, denote the content frame, stylized frame, edge map of content, and edge map of stylized one. $E_{s/c}$ is used to visualize the difference between E_s and E_c. The stylized frame generated by our method is used in this demonstration. The visualizations of the compared methods are presented in the Appendix, Figure 20.

Table 2. Analysis on the Perfomance of Content Structure Preservation

		[24]	AdaIN [21]	MCCNet [7]	ArtFlow [1]	[43]	Ours
SSIM	on source	0.643	0.524	0.482	0.561	0.605	**0.763**
	on edge map	0.595	0.621	0.597	0.518	0.526	**0.784**
	RCR	0.589	0.285	0.462	0.293	0.318	**0.615**

4.5 Evaluation Metrics

To quantitatively evaluate our method's performance, we measure our results on two aspects, i.e., performance on preserving content structure on each single stylized frame and temporal coherency on the entire generated video. In this quantitative evaluation, we use 72 videos that are generated by five competitors, i.e., Johnson et al. [24], AdaIN [21], MCCNet [7], ArtFlow [1], Shih et al. [43], and ours. Each method consists of a set of 12 stylized videos, i.e., four input videos are stylized with four MArt styles.

4.5.1 Content Structure Preservation. To measure the effectiveness of our model in terms of preserving content structure, we use two metrics: **Structural Similarity Index (SSIM)** [19] and **region coverage rate (RCR)**. We note here that "*content structure*" encompasses of moving objects and the map background. SSIM is an index measuring the structure similarity between two images. When two images are nearly identical, their SSIM is close to 1. In our evaluation, we use SSIM to measure the similarity between the stylized frames and their corresponding edge map against that of the source frames. These are illustrated intuitively in Figure 15. We can see that our results keep plausible edge maps with consistent spatial and structure senses. By comparing the edge maps [31] computed from the content and stylized frames, we find that our results can better recover the maps than the compared methods. In terms of RCR, we define as $RCR = (F_s \wedge F_c)/(F_s \vee F_c)$, where F_s is the stylized frame and F_c is the corresponding content frame. The analysis results on two metrics are outlined in Table 2. For both measurement metrics, the higher the values are, the better preservation the results are. In terms of SSIM, the score on edge map of our method is more close to that of the source frame and outperforms the compared methods. This result reveals that our method is more tailored to preserve the content structure of the input than others. For the RCR score, our method achieves the highest value, and Johnson et al. [24] has a relatively comparable score with ours. This implies that Johnson et al. [24] is tolerant for the current application. However, by visually inspecting the results, Johnson's results are blurry and the style is not distributed as well as ours.

Table 3. Comparisons on Temporal Coherency

	Source video	[24]	AdaIN [21]	MCCNet [7]	ArtFlow [1]	[43]	Ours
$Mean_\delta$	0.017	0.183	0.157	0.072	0.083	0.272	**0.052**
$LPIPS(.)$	0.162	0.318	0.247	0.215	0.284	0.352	**0.197**

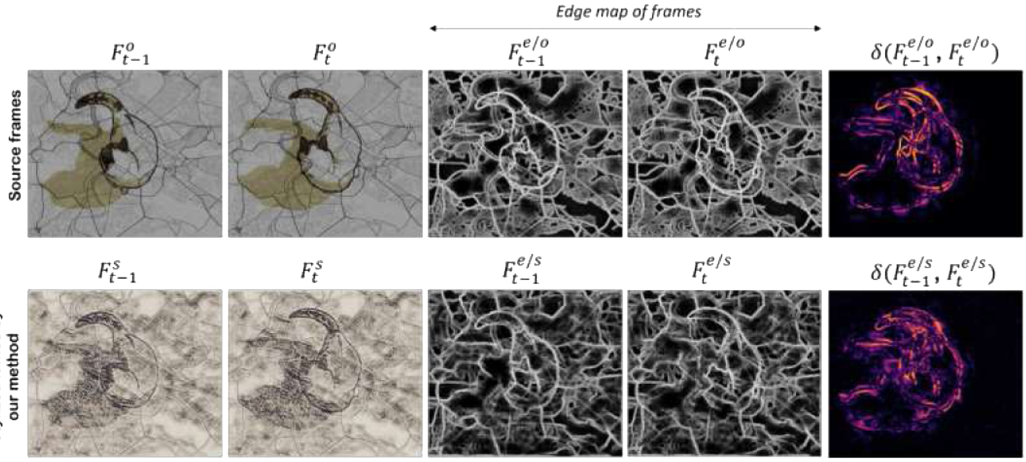
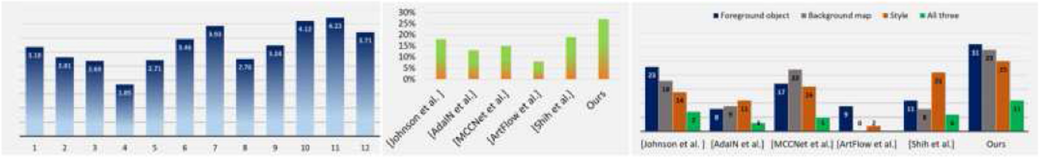


Fig. 16. Visualization of the objects used in evaluating the temporal consistency. Two consecutive frames of source video (F^s_{t-1}, F^s_t) and corresponding stylized frames (F^o_{t-1}, F^o_t) by each method are extracted to edge maps denoted by ($F^{e/o}_{t-1}, F^{e/o}_t$) and ($F^{e/s}_{t-1}, F^{e/s}_t$), respectively. Differences of edge map $\delta(.)$ are visualized in the last column. The stylized frames generated by our method are used in this demonstration. The results of the compared methods are presented in the Appendix, Figure 19. The metric of this comparison is presented in Table 3.

4.5.2 Temporal Consistency. To quantitatively evaluate the temporal consistency in the rendered videos, we synthesize 15 stylized videos by Johnson et al. [24], AdaIN [21], MCCNet [7], ArtFlow [1], Shih et al. [43], and our method and measure the coherency of rendered videos by calculating two metrics. Given two consecutive frames, we first adopt an edge extraction algorithm [31] to generate the edge maps of these frames (as visualized in Figure 16). Then, we define $\delta_{F(t)} = \| F_t - F_{t-1} \|$ and calculate the mean ($mean_\delta$) of $\delta_{F(t)}$. Second, we use the **LPIPS (Learned Perceptual Image Patch Similarity)** metric proposed by Zhang et al. [52]. LPIPS computes distance in AlexNet feature space with linear weights to better match human perceptual judgments. On each pair of consecutive frames, we compare these two metrics of both the generated pairs and source pairs. During the experiment, we feed the edge maps to $\delta_{F(t)}$, while the source frames and generated frames are fed in LPIPS. We then treat the values of these metrics on the source pairs as the ground truth, since the input frames are themselves temporally consistent. The results, which have closer values to those of the ground truth, are more stable. Table 3 shows the metric error that measures the results. From the results, we can conclude that our method can yield the best video results with high consistency. For the better visualization on video, please see our video results on our website.

4.5.3 User Study. In addition to visually inspecting the results from the above comparisons, we conduct two user studies to further learn the human perception on the performance of our system and visual quality of our results. Two user studies are conducted independently on two distinct groups of participants and designed with different goals. The first one, denoted as **US**-p, is



(a)

(b-1)

(b-2)

Fig. 17. Analysis result on the user study **US**-p (a) and on the user study **US**-q (b-1 and b-2).

to measure the performance of our proposed system in terms of preserving content structure. The other one, denoted as **US**-q, is to validate the visual quality of our stylized videos.

In **US**-p, we invite 11 participants to join in this study. Seven of them have graphics-related backgrounds. Only our results are used in this user study. We randomly select 12 videos in our results in which they have diversity in map backgrounds and animation objects. Each time, a participant was first shown a source content frame and then the corresponding stylized frame side-by-side. Every participant watched 12 examples and each time was asked to answer the question: "*How do you rate the degree of content structure preservation in the stylized result (structure clear? messy lines? distorted text?)?*" The participants answer the question by voting in one of the following five levels: *1, 2, 3, 4, 5* (1 = bad, 5 = very good). Thereafter, we compute the average score from 11 participants as the performance of our system in terms of content structure preservation. A higher score means better agreement for the good performance. Figure 17(a) shows the statistic results. As indicated from this figure, most of our results are judged to well preserve content structure. There is one case in which the score is in the lowest rate among 12 results. However, it is still at acceptable rate, i.e., close to the average rate 2.5. And the result in low rate only accounts for 8.3% of the total data used in this user study. Please see our project website for the visualization of these 12 videos.

In **US**-q, a total of 19 participants are invited to join in this user study. Five of them have the interest of art. We select four animations and compose each of them with a random digital map to generate four content videos. Five style transfer methods are used in this session, Johnson et al. [24], AdaIN [21], MCCNet [7], ArtFlow [1], and Shih et al. [43]. Four content videos are stylized into four different MArt styles by six methods, i.e., five compared methods and ours. As a result, each method has 16 stylized videos. We first show the content video and the MArt style image, then a set of six stylized results is shown. The resultant videos are displayed in a random order, and the participants are not provided any video information to prevent them from inferring the method and having bias in their perceptual feeling. Then, at the end of each set, we ask the users two questions:

Q1. Which stylized result do you like the best?
Q2. In the result you selected in Q1, which factor is mainly considered: Foreground object? Background map? Style? or All three?

We ask the users to select only one result in Q1 is a single choice question and Q2 is a multiple choice one. Figure 17(b-1) shows the percentage of each method as the best, and Figure 17(b-2) presents the number of votes on each factor of each method in Q2. The analysis result of Q1 reveals that our results receive majority vote as the best one. Three methods, Johnson et al. [24], MCCNet [7], and Shih et al. [43], are also judged to be potential. Among our competitors in this user study, ArtFlow [1] is rarely selected, because it cannot handle the rich content in this current application, as we can see the visual comparisons in the previous session. Inferring the analysis in Figure 17(b-2), our results also receive the highest vote for the quality of all three raised factors. In terms of the style, Shih et al. [43] receive the higher vote than ours, but the difference is not

Table 4. The Average Inferring Time (in Seconds) with Different Resolutions of Input Videos

Resolution/Method	[24]	AdaIN [21]	MCCNet [7]	ArtFlow [1]	[43]	Ours
256 × 256	67	1.57	4.62	3.14	139.5	**0.86**
512 × 512	71	2.4	6.18	5.47	124.4	**0.95**
1,024 × 1,024	76.5	3.375	7.36	8.92	173	**4.27**

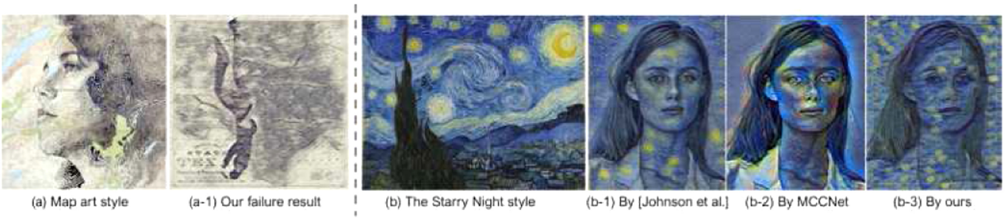

(a) Map art style (a-1) Our failure result (b) The Starry Night style (b-1) By [Johnson et al.] (b-2) By MCCNet (b-3) By ours

Fig. 18. Examples in which our method may not perform well. (a) is the MArt style with disentanglement pattern; (b) is the generic artist style, i.e., by Vincent van Gogh.

significant. And the scores on other aspects (e.g., foreground object, background map) of Shih's method are relatively lower than ours.

4.5.4 Speed Analysis. All experiments are conducted on a PC with Intel Core i7 2.5 GHz, 16 GB RAM. Our current system is implemented with the PyTorch framework [35] accelerated by a single GPU NVIDIA Geforce RTX 306. The offline training process takes approximately nine hours and converges after two epochs. After training, the stylizing process consumes only 0.72 second, on average, to process one frame. We compare the average running time of our method with related works in Table 4. The experiments are conducted on the videos of the same length but different resolutions. All timing statistics are recorded on a PC with GPU NVIDIA GeForce RTX 306. Obviously, our method performs faster than the compared methods when inferring the same video resolutions.

4.6 Limitations

Our framework still suffers from some limitations. The current trained model does not have good performance in two cases. The first case is the map art styles composing of disentanglement pattern, as visualized in Figure 18(a). This is mainly due to the lack of such data in our training set. Our model may accidentally regard some non-screened regions as stylized regions and try to stylize them. This may cause the map attributes to be distorted. For example, in Figure 18(a-1), the disentanglement patterns of the map art style are distributed in the entire images and thus yield distortion to map information. The other case is that transferring the generic artist styles, for example, "The Starry Night" style by Vincent van Gogh, shown in Figure 18(b). These styles are created by a different medium, oil painting. Our method may fail to preserve the structure of content when transferring these styles. It is worth pointing that such kind of styles is not our focus in this work. However, as shown in this figure, our method also can deliver the similar results compared to those are originally designed for this style.

5 CONCLUSIONS AND FUTURE WORK

In this article, we propose a learning-based framework for ***Map Art Video*** **(MAVi)** generation. To the best of our knowledge, we are the first to apply deep learning to MAVi application. The core contribution is the awareness of (1) preserving the map functionality and the consistency in map structure and (2) supporting MAVi generation of arbitrary input videos. By proposing MAViNet, suitable objective functions, and rich map art training data, we are able to generate realistic videos. Our results show that the proposed system substantially outperforms related methods in terms of visual quality and meets the mentioned criteria in this research domain. We also demonstrate multiple potential applications utilizing the benefits from our method. In future works, we plan to explore more advanced network architectures to investigate the complicated semantic in MArt. In addition, extending this work to other artist styles could be a possibility in our near future. We hope this will help bridge the gap between digital animation and traditional hand-painted artist work.

APPENDICES

A TRAINING DATA PREPARATION

This section introduces our training data preparation for MArtVi style transfer investigation. The dataset should be diverse in moving objects and the cartography background. Such dataset is not easily obtainable. The existing videos are deficient in the diversity of background. Most of them do not compose of map as the background, and it is difficult to separate the foreground from the background. To make it adequate for reliable learning of MArt transferring, we prepare a dataset for our current application by the following steps. First, we collect two kinds of data: (1) a large set of ordinary videos with the main objects manually labeled and (2) a collection of digital maps in the real world. Second, we extract the main objects, which are labeled in the videos, and merge with a random map in the collection. This strategy yields a new set of videos, in which the backgrounds are maps. We call such videos in the term *m*-Vi. We then use *FlowNet2* [8] to calculate the optical flow of *m*-Vis. Finally, we follow the protocol in Ruder et al. [40] that formulates former video and predicted optical flow to define the motion boundary and occlusion map. In other words, this value presents whether objects in the consecutive frames are occluded. And, we use it in coherence loss in our later training process.

In our data preparation, we collect the entire videos in the DAVIS-2017 dataset [36], which is provided by DAVIS Challenge on video object. This dataset, which is originally designed for video segmentation, contains 90 sets of videos. Each segment of video is attached with a manually cut label. Simultaneously, a set of digital maps is prepared. We collect 1,000 map images from Pinterest by providing seven keywords acquired from this website and cluster them in two collections. One is called *content-map-collection*, which contains the images used to merge with content in videos to produce *m*-Vi. The other, called *style-map-collection*, refers to the map images treated as the style in the transferring process. Thereafter, each video is combined with a random image in the content-map-collection. As a result, we get a dataset of 90 *m*-Vis.

The goal of our proposed framework not only is to transfer MArt style to a *m*-Vi but also obtain the consistency in stylized video. To enforce stronger consistency between adjacent frames, motivated by Reference [40], we need to detect disoccluded regions and motion boundaries. Basically, the CartVi dataset generated in the first phase does not provide relevant optical flow information. To achieve this, we adopt *FlowNet2* [8] to calculate the optical flows of *m*-Vi. Thereafter, the optical flows are used in a forward-backward flow warping in two adjacent frames to detect disocclusions [40]. Based on the defined flow in both forward and backward directions, motion boundaries are detected [40]. The reason we employ the disoccluded regions and motion boundaries detection in our dataset preparation is that if the object is occluded, then the error before and after warped calculation is larger and thus it increases the calculation error of the training model. Recall that after the disocclusion detection manner, two flow maps are obtained. One is the flow map in forward direction and the other represents the flow map in the backward direction. Last, these two maps are synthesized to a map, called “*weight map*,” for later use in the temporal consistency loss computation. Later, we will experimentally show these two maps can significantly improve both temporal coherence and cartography consistency in stylized video results.

B MORE VISUALIZATIONS

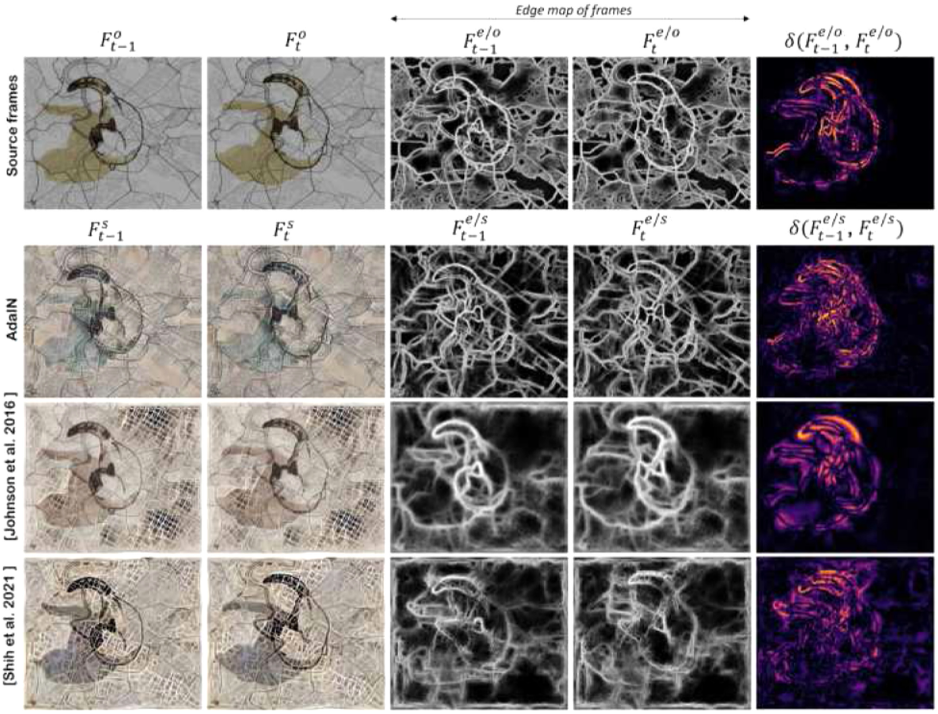


Fig. 19. Visualization of content structure preservation of the compared methods.

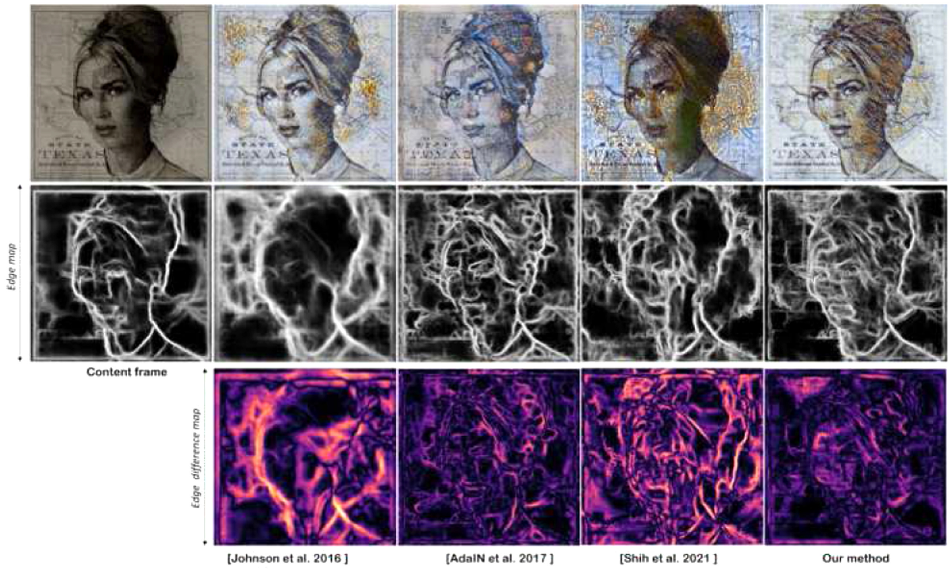


Fig. 20. Visualization of differences in adjacent frames of the compared methods.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for the many constructive comments that greatly help improve our article.

REFERENCES

[1] Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. 2021. ArtFlow: Unbiased image style transfer via reversible neural flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 862–871.

[2] Eva Cetinic and James She. 2022. Understanding and creating art with AI: Review and outlook. *ACM Trans. Multim. Comput. Commun. Applic.* 18, 2 (2022), 1–22.

[3] Alex J. Champandard. 2016. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv preprint arXiv:1603.01768* (2016).

[4] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2017. Coherent online video style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*. 1105–1114.

[5] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. 2017. StyleBank: An explicit representation for neural image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1897–1906.

[6] Yi-Wen Chen, Yi-Hsuan Tsai, Chu-Ya Yang, Yen-Yu Lin, and Ming-Hsuan Yang. 2018. Unseen object segmentation in videos via transferable representations. In *Proceedings of the Asian Conference on Computer Vision*. Springer, 615–631.

[7] Yingying Deng, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, and Changsheng Xu. 2021. Arbitrary video style transfer via multi-channel correlation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 1210–1217.

[8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 2758–2766.

[9] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. 2016. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629* (2016).

[10] Ed Fairburn 2021. *Ed-Fairburn, Original Artwork and Illustration*. Retrieved from https://edfairburn.com/.

[11] Chang Gao, Derun Gu, Fangjun Zhang, and Yizhou Yu. 2018. Reconet: Real-time coherent video style transfer network. *Asian Conference on Computer Vision*, Springer, 637–653.

[12] C. Gao, D. Gu, F. Zhang, and Y. Yu Reconet. 2018. Real-time coherent video style transfer network. In *Proceedings of the Asian Conference on Computer Vision*.

[13] Chang Gao, Derun Gu, Fangjun Zhang, and Yizhou Yu. 2018. ReCoNet: Real-time coherent video style transfer network. In *Proceedings of the Asian Conference on Computer Vision*. Springer, 637–653.

[14] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* (2015).

[15] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2414–2423.

[16] Amy Gooch. 2001. *Non-photorealistic Rendering*. DOI:https://doi.org/10.1201/9781439864173

[17] Agrim Gupta, Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2017. Characterizing and improving stability in neural style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*. 4067–4076.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[19] Alain Hore and Djemel Ziou. 2010. Image quality metrics: PSNR vs. SSIM. In *Proceedings of the 20th International Conference on Pattern Recognition*. IEEE, 2366–2369.

[20] Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. 2017. Real-time neural style transfer for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 783–791.

[21] Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*. 1501–1510.

[22] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*. PMLR, 448–456.

[23] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. 2019. Neural style transfer: A review. *IEEE Trans. Visualiz. Comput. Graph.* 26, 11 (2019), 3365–3385.

[24] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision*. Springer, 694–711.

[25] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 8110–8119.

[26] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[27] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. 2019. Learning linear transformations for fast image and video style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 3809–3817.

[28] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. 2017. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088* (2017).

[29] Honglin Lin, Mengmeng Wang, Yong Liu, and Jiaxin Kou. 2022. Correlation-based and content-enhanced network for video style transfer. *Pattern Anal. Applic.* (2022), 1–13.

[30] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. 2021. AdaAttN: Revisit attention mechanism in arbitrary neural style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 6649–6658.

[31] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. 2017. Richer convolutional features for edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3000–3009.

[32] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. 2017. Deep photo style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 4990–4998.

[33] Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning (ICML).*

[34] Augustus Odena, Vincent Dumoulin, and Chris Olah. 2016. Deconvolution and checkerboard artifacts. *Distill* 1, 10 (2016), e3.

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga et al. 2019. PyTorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703* (2019).

[36] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. 2017. The 2017 DAVIS challenge on video object segmentation. *arXiv:1704.00675* (2017).

[37] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[38] Ana Daniela Peres Rebelo, Guedes De Oliveira Inês, and D. E. Verboom Damion. 2022. The impact of artificial intelligence on the creativity of videos. *ACM Trans. Multim. Comput. Commun. Applic.* 18, 1 (2022), 1–27.

[39] Paul L. Rosin and J. Collomosse. 2013. Image and video-based artistic stylisation. In *Computational Imaging and Vision.* Springer.

[40] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. 2016. Artistic style transfer for videos. In *Proceedings of the German Conference on Pattern Recognition.* Springer, 26–36.

[41] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. 2018. Artistic style transfer for videos and spherical images. *Int. J. Comput. Vis.* 126, 11 (2018), 1199–1219.

[42] Ahmed Selim, Mohamed Elgharib, and Linda Doyle. 2016. Painting style transfer for head portraits using convolutional neural networks. *ACM Trans. Graph.* 35, 4 (2016), 1–18.

[43] Chiao-Yin Shih, Ya-Hsuan Chen, and Tong-Yee Lee. 2021. Map art style transfer with multi-stage framework. *Multim. Tools Applic.* 80, 3 (2021), 4279–4293.

[44] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[45] Saurabh Singh and Shankar Krishnan. 2020. Filter response normalization layer: Eliminating batch dependence in the training of deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 11237–11246.

[46] Thomas Strothotte and Stefan Schlechtweg. 2002. *Non-photorealistic Computer Graphics: Modeling, Rendering, and Animation.* Morgan Kaufmann.

[47] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2017. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 6924–6932.

[48] Kai Xu, Longyin Wen, Guorong Li, Honggang Qi, Liefeng Bo, and Qingming Huang. 2021. Learning self-supervised space-time CNN for fast video style transfer. *IEEE Trans. Image Process.* 30 (2021), 2501–2512.

[49] Mao-Chuang Yeh and Shuai Tang. 2018. Improved style transfer by respecting inter-layer correlations. *arXiv preprint arXiv:1801.01933* (2018).

[50] Hang Zhang and Kristin Dana. 2018. Multi-style generative network for real-time transfer. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops.*

[51] Lingzhi Zhang, Tarmily Wen, and Jianbo Shi. 2020. Deep image blending. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 231–240.

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

Received 21 October 2021; revised 2 June 2022; accepted 11 November 2022