Computational Visual Media
https://doi.org/10.26599/CVM.2026.9450481 Vol. 12, No. 5, October 2026, 1441–1457

Research Article

User-oriented interactive style transfer

Zheng Lin[1], Zhao Zhang[2], Kang-Rui Zhang[2], Bo Ren[2] (✉), and Ming-Ming Cheng[2]

© The Author(s) 2026.

Abstract Neural style transfer (NST) can create impressive artworks by transferring a reference style to a content image. Current image-to-image NST methods lack the fine-grained control often demanded for artistic editing. To mitigate this limitation, we propose a user-oriented interactive style transfer (IST) method, using which a harmonious image like drawing can be interactively created. Our IST method can serve as a brush, dipping style from anywhere, and then painting to any region of the target content image. To control the action scope, we formulate a fluid simulation algorithm, which takes styles as pigments around the position of brush interaction, and uses diffusion in style or content images according to similarity maps. By dipping and painting, even employing a single style image can produce thousands of eye-catching works. Our method expands the creative capabilities of NST.

Keywords user interaction; style transfer; image manipulation; content creation

1 Introduction

Neural style transfer (NST) is a widely considered artistic creation tool, which aims to transfer the artistic style from a reference image to a content image. Many impressive NST methods have been proposed, emphasizing content retention [1, 2], robustness [3, 4], the ceiling of style types [5, 6], geometric changes [7], brushstroke simulation [8], etc. Recently, several works have focused on user intent, considering manual adjustment of the degree of stylization [9], semantic guidance [10, 11], and masked style splicing [12, 13]. Although these methods can create impressive artworks, through global adjustment, they cannot provide fine-grained control over artistic editing, and do not follow natural drawing logic.

In this paper, we propose a user-oriented interactive style transfer (IST) method, in which users can dip a style brush into any region in the style images and then paint the style on any region of the target content image, in a similar way to drawing. (For simplicity, we will use "dipping a style" in the following to refer to the process of selecting a style.) The scopes of dipping and painting are supported by our proposed fluid simulation algorithm, making styles flow and blend like pigments in the content image, according to neural similarity maps. Our IST method expands the creative capability of traditional NST methods and allows greater artistic imagination in creating artworks with harmonious style. Some examples are shown in Fig. 1. In Fig. 1(a) the user dips the whole style of the style image and paints on the content image with different interactions. The stylized scope in the content image first covers the clouds, and then fills the whole sky. Compared to previous image-to-image NST, our IST allows users to freely control the scope of stylization. Therefore, users can transfer different styles to different regions of the content image. In Fig. 1(b), different areas show separate style tendencies, such as stroke direction and color concentration within a style image. Our IST can freely dip the style of a specific region, then paint on the content image. In Fig. 1(c), in ordinary drawing, a brush can be dipped in many pigments. We can also dip multiple styles from one or more style images, to be blended in the target content region. This interactive fusion can create thousands of styles by combination.

1 BNRist, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: frazer.linzheng@gmail.com.

2 College of Computer Science, Nankai University, Tianjin, China. E-mail: Z. Zhang, zzhang@mail.nankai.edu.cn; K.-R. Zhang, kang_rui@mail.nankai.edu.cn; B. Ren, rb@nankai.edu.cn (✉); M.-M. Cheng, cmm@nankai.edu.cn.

Manuscript received: 2024-07-29; accepted: 2025-02-22

清華大學出版社 Tsinghua University Press Available on IEEE Xplore®

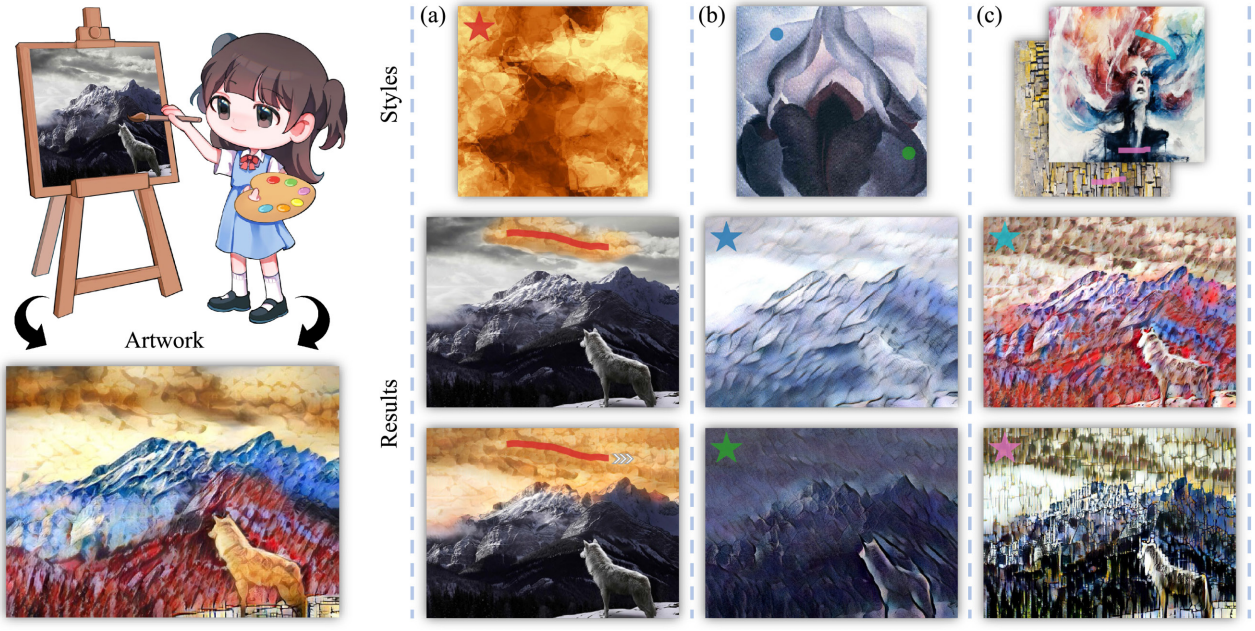


Fig. 1 Our proposed interactive style transfer (IST) method allows users to transfer style using a drawing-like process. Users can dip style brushes at any position in multiple style images and create attractive art on the content image. Some interesting cases using interaction are shown above. (a) Same style image, different scopes of painting, (b) same style image, different dipping regions to create varied styles, (c) dipping and merging various styles in the brush on either one or more images. The dot "•" and scribble "≀" indicate user interactions, while the star "★" indicates dipping or painting operations on the whole image.

Our carefully designed interaction modes and algorithms allow users to create artworks according to their wishes. We show a complete artwork produced by our method in Fig. 2. More creative scenes are shown in following sections.



Fig. 2 An artwork created by our IST method. We omit the interaction and simply show the style sources, content image, and end result. Using operations of dipping and painting, we decorate the sky with stars, the mountains with ink, and the river with light.

2 Related work

2.1 Neural style transfer

Neural style transfer (NST) [5, 14–16] aims to create novel artworks by transferring the artistic style from a reference image to a content image. Making the transfer process more flexible and efficient has always been the focus of researchers. *Single style* methods [17, 18] learn a style-specific network, relying on perceptual loss [17], a GAN [19], normalization [20], etc. These earlier single style methods must be retrained for new styles. Alongside these developments, the rapid evolution of deep generative models, including GANs and diffusion models, has significantly advanced the broader field of image synthesis and generation [21, 22]. *Multiple style* methods [14, 23–25] can transfer from multiple given styles. Such methods often save specific parameters for each style. However, in real-world applications, it is difficult to learn all styles in advance. *Arbitrary style* methods [2, 4, 5, 15, 16, 26–35] can use any given reference style image. Online optimization methods [36–38] decouple content and style information in input images using a pre-trained model. With iterative optimization, the new artistic image imitates style image in the feature Gram matrix, while maintaining similarity with the content image in feature representation. Online optimization is time consuming, requiring intensive calculation. Other methods have tried to transfer arbitrary styles using a single forward pass. A popular practice [2, 4, 5, 15] directly aligns the statistics of the new artwork and the given style image in the instance feature space. Some works [6, 39, 40] learn to predict parameters

for any specified style. Recent approaches leveraging diffusion models have demonstrated remarkable results [41–44]. Concurrently, controllable multi-domain semantic artwork synthesis [45], style-aware diffusion adaptation [46], and content-controllable painting recreation [47] have expanded the boundaries of digital art. Other specific stylization branches, such as anime-style colorization [48], face cartoonization [49], and audio-guided local stylization [50], further enrich the diversity of stylized creations. Although such methods improve the flexibility, processing is still applied to the full image, which cannot meet the needs of all users. Some methods like Ref. [34] allow for the application of different styles to distinct regions of an image, but this is done algorithmically, with predefined criteria determining how the styles are assigned. This automated approach offers efficiency in applying styles but provides limited flexibility in terms of user customization. In contrast, our method introduces user interaction into the process, allowing for more customizable and tailored style transfer. By enabling users to manually select and adjust which styles are applied to specific regions, our approach offers greater control over the final artistic result. This interactive characteristic makes the style transfer process more flexible and personal, empowering users to achieve more precise creative results.

2.2 Style transfer with interactions

The interaction discussed here is drawing interaction, rather, e.g., textual interaction [51–53]. Some works have introduced the latter kind of interaction as a way to adjust the style. The pursuit of intuitive user control is highly active across various image manipulation domains. For instance, researchers have explored user-guided portrait editing with conditional GANs [54], language-guided local editing [55], and semantic-based facial manipulation [56]. Ref. [57] allow users to control the ratio of different sub-styles extracted from a style image. Ref. [58] provides a fast soft color segmentation approach, by which which users can extract and change the color distribution. Other works focus on restricting transfer to local areas. This is often done by utilizing a mask, e.g., a semantic mask [10] or an instance mask [13]. Many methods [9, 59] adopt binary masks generated by users, such as brush areas [60] and object masks [12, 61] determined by an interactive segmentation method [62]. In addition, other operations apply, e.g., local-retouching for style refinement [63], changing the semantic mask to generate an image with a new texture [11], and transferring the vector graphic [64]. However, these works based on adjusting style parameters of the entire image fail to provide fine-grained control. The results from mask-based methods may look contradictory, and producing masks may place extra burdens. In this paper, we propose a method like a real drawing process with dipping and painting operations, unlike previous works.

2.3 Fluid simulation

In computer graphics, fluid simulation is an active field, studying how to simulate fluids using computers. For a comprehensive review of physics-based fluid simulation and its recent research trends in computer graphics, we refer readers to Ref. [65]. Controllable fluid simulation is essential for animators to achieve special effects and animations in 2D [66] and 3D [67] scenes. Many methods have been proposed to calculate control forces for the fluid, based on blurred density gradients [68], proportional errors of fluid shape between key frames with velocity feedback [69], and optimal controllers [70]. For image processing, simplified forms of fluid motion equations, such as an anisotropic diffusion equation [71], are used in denoising and edge detection. In our work, we introduce fluid simulation for fine-grained control of IST.

3 Method

3.1 Framework overview

Our framework for IST is inspired by traditional oil painting. We treat the content image $\boldsymbol{C}$ as a painting canvas, and the set of style images $\{\boldsymbol{S}^1, \ldots, \boldsymbol{S}^n\}$ as the painting palette. The user interactively dips the style from the style images, using clicks and scribbles, and then interactively paints on the content image to transfer the style to specific regions. Figure 3 outlines the interactive process using a single dipping and painting operation. In the implementation of the algorithm, we input the content and style images into the backbone neural network to obtain their deep features for each layer, denoted $\boldsymbol{F}^i$, calculate the similarity with the interaction location to obtain a global similarity map, denoted $\boldsymbol{G}$ (see Section 3.2), and then use our similarity-based fluid

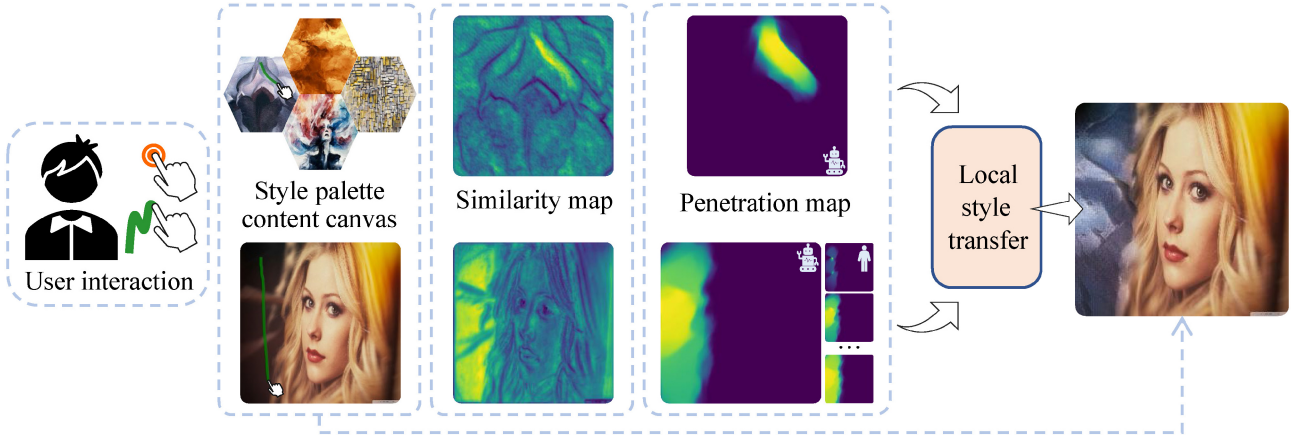


Fig. 3 Framework of our interactive process. The user interacts with style and content images. A similarity map is used in the fluid simulation algorithm, which generates a penetration map. (This is done automatically for style images and uses an additional manual process for content images.) A local style transfer method then generates the result.

simulation algorithm (see Section 3.3) to generate the penetration map $\boldsymbol{P}$ of interactions on the two images. We modify the classic AdaIN [5] method for use as the style transfer algorithm and use local style transfer (see Section 3.4) to apply the local style to the selected region of the content image. Instead of simply splicing the stylized image through a binary mask, our IST method generates images with smooth transitions by considering soft style diffusion and blending. Whether considering style dipping or painting, the degree of stylization for each pixel is a numerical weight. If different styles are applied to the same position after multiple interactions, the styles are merged according to their fluid concentration ratio. We also adopt the characteristics of oil painting, making subsequent styles dominant (see Section 3.5).

3.2 Revisiting neural feature similarity

As we know, neural network features show different levels of image characteristics [72], from the low-level information, e.g., color and texture, to the high-level information, e.g., semantic. For our framework, users take clicks or scribbles on the content and style images. We also adopt these neural features to calculate the interaction radiation area. In terms of concrete implementation, we use the widely used VGG [73] network as the backbone and utilize the first four layers. For each layer, we will calculate the corresponding similarity map by averaging the feature similarity between the user interaction and all pixels. These user interactions $\boldsymbol{I}$ may contain only one pixel (click) or more (scribble). Then we average the similarity maps in four layers into one $\hat{\boldsymbol{G}}$ as the basis of our fluid simulation algorithm. We introduce a variable parameter s that can be adjusted by the user, with a range from 0 to 1. For the final weighted $\hat{\boldsymbol{G}}$, the weights of $\boldsymbol{G}^1$ and $\boldsymbol{G}^2$ are $\frac{1-s}{2}$, and the weights of $\boldsymbol{G}^3$ and $\boldsymbol{G}^4$ are $\frac{s}{2}$. When $s = 0.5$, we take the average of the four similarity maps ($\overline{\boldsymbol{G}}$), which is the default setting for our method. We utilize the cosine similarity as the metric (shown in Eq. (1)). The calculation process of similarity maps can be formulated as

$$\boldsymbol{G}_{m,n} = \frac{1}{\sum \boldsymbol{I}} \sum_{i,j} \frac{\boldsymbol{F}_{m,n} \cdot \boldsymbol{F}_{i,j} \cdot \boldsymbol{I}_{i,j}}{\|\boldsymbol{F}_{m,n}\| \|\boldsymbol{F}_{i,j}\|}, \quad \boldsymbol{I}_{i,j} \in \{0,1\} \quad (1)$$

where (i, j) means the coordinates of all pixels and $\|\boldsymbol{F}_{i,j}\|$ means the 2-norm value in position (i, j). In Fig. 4, we illustrate the similarity maps of four layers according to the click. From $\boldsymbol{G}^1$ to $\boldsymbol{G}^4$,We can find that similar regions transit from similar colors to

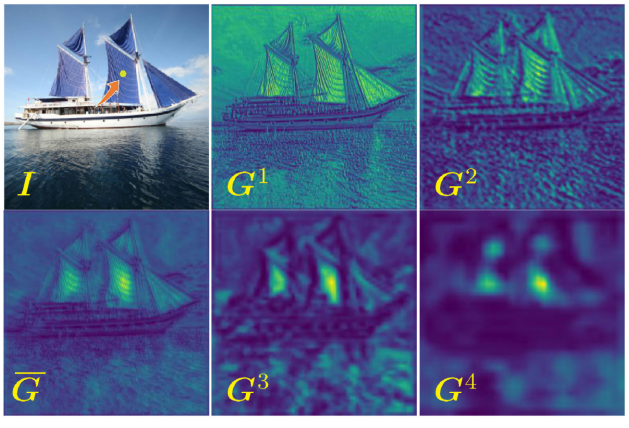


Fig. 4 Neural similarity maps based on location of interaction. $\boldsymbol{G}^i$ shows similarity maps of features in four layers based on the click in $\boldsymbol{I}$. The average is shown as $\overline{\boldsymbol{G}}$.

similar semantics. On the whole, the deeper the features are, the information concerned becomes more complex and the similar area is smaller. It is worth mentioning that the similarity maps are based on the original image, so no matter how we interact later, the similarity map will not be updated accordingly.

3.3 Similarity-based fluid simulation

3.3.1 Approach

The global similarity maps measure the correlation between the interaction position and the pixels of the whole picture. However, in the interactive process, users often hope to obtain the style from or transfer the style to the area around the interaction. More importantly, users often need to control the scope of painting, as in brush hardness or concentration controls in painting. Therefore, we introduce an algorithm of similarity-based fluid simulation to generate the influence area of user interactions. Since diffusion and absorption of ink can be regarded as opposite processes, we use the same algorithm for dipping and painting style. Figure 5 shows the concept of our algorithm. Generally speaking, similarity controls resistance to fluid diffusion. The higher the similarity, the smaller the resistance. The fluid concentration gradually diffuses outwards from the position of user interaction.

For multiple styles, in contrast to generating hard areas consisting of Boolean values 0 and 1, our goal is to generate a smooth transition at the border. Pixels at positions where the user directly interacts have a higher concentration of fluid, which diffuses to surrounding pixels with a lower concentration over time. We therefore use a fluid-simulation like algorithm to achieve our goal. Specifically, a penetration function $\boldsymbol{P}(x,y,t)$ represents a distribution of fluid concentration. The similarity-based diffusion coefficient function $\boldsymbol{D}(x,y)$ represents a parameter controlling the velocity of fluid diffusion. The 2D fluid diffusion equation is

$$\frac{\partial \boldsymbol{P}(x,y,t)}{\partial t} = \nabla \cdot (\boldsymbol{D}(x,y)\nabla \boldsymbol{P}(x,y,t)) \tag{2}$$

where $(x,y) \in [0,1] \times [0,1]$, with initial condition $\boldsymbol{P}(x,y,0) = \boldsymbol{I}(x,y)$ where $\boldsymbol{I}(x,y)$ represents the interactive position in continuous space. We use the Neumann boundary condition, $\partial \boldsymbol{P}(x,y,t)/\partial n = 0$ on the boundary, where n represents the normal direction to the boundary, to restrict the fluid to move only along the image boundary and not diffuse outside the image. $\nabla\cdot$ and ∇ represent divergence and gradient operators respectively. $\boldsymbol{P}(x,y,t)$ is a continuous function of position (x,y) and time t. $\boldsymbol{D}(x,y)$ and $\boldsymbol{I}(x,y)$ are continuous functions of position (x,y). It needs to be emphasized that a similarity-based $\boldsymbol{D}(x,y)$ in Eq. (2) makes the fluid concentration diffuse more in areas with higher similarity. In contrast, a standard isotropic diffusion model diffuses concentration uniformly in every direction, allowing less controllability. For the convenience of discretization of Eq. (2), we write it in the form of components as Eq. (3):

$$\begin{aligned}\frac{\partial \boldsymbol{P}(x,y,t)}{\partial t} = &\frac{\partial}{\partial x}\left(\boldsymbol{D}(x,y)\frac{\partial}{\partial x}\boldsymbol{P}(x,y,t)\right) + \\ &\frac{\partial}{\partial y}\left(\boldsymbol{D}(x,y)\frac{\partial}{\partial y}\boldsymbol{P}(x,y,t)\right)\end{aligned} \tag{3}$$

3.3.2 Discretization scheme

In order to make the calculation domain of the discretized diffusion equation to be the image we care about, we follow the discrete domain of the image, consisting of $N_i \times N_j$ pixels. We write $\boldsymbol{P}^n_{i,j}$ for $\boldsymbol{P}(i\Delta h_i, j\Delta h_j, n\Delta t)$, $\boldsymbol{D}_{i,j}$ for $\boldsymbol{D}(i\Delta h_i, j\Delta h_j)$, where $\Delta h_i = 1/N_i$, $\Delta h_j = 1/N_j$, and Δt is a given time step. We can now define the penetration map $\boldsymbol{P}^n$ and diffusion coefficient map $\boldsymbol{D}$ as two matrices consisting of $\boldsymbol{P}^n_{i,j}$ and $\boldsymbol{D}_{i,j}$, respectively. An implicit backward difference in time method [74] is used to solve Eq. (3) stably and quickly. The discretization scheme for Eq. (3) is

$$\begin{aligned}\frac{\boldsymbol{P}^{n+1}_{i,j} - \boldsymbol{P}^n_{i,j}}{\Delta t} = &\Big(\boldsymbol{D}_{i+1/2,j}\left(\boldsymbol{P}^{n+1}_{i+1,j} - \boldsymbol{P}^{n+1}_{i,j}\right) - \\ &\boldsymbol{D}_{i-1/2,j}\left(\boldsymbol{P}^{n+1}_{i,j} - \boldsymbol{P}^{n+1}_{i-1,j}\right)\Big)/\Delta x^2 + \\ &\Big(\boldsymbol{D}_{i,j+1/2}\left(\boldsymbol{P}^{n+1}_{i,j+1} - \boldsymbol{P}^{n+1}_{i,j}\right) - \\ &\boldsymbol{D}_{i,j-1/2}\left(\boldsymbol{P}^{n+1}_{i,j} - \boldsymbol{P}^{n+1}_{i,j-1}\right)\Big)/\Delta y^2\end{aligned} \tag{4}$$

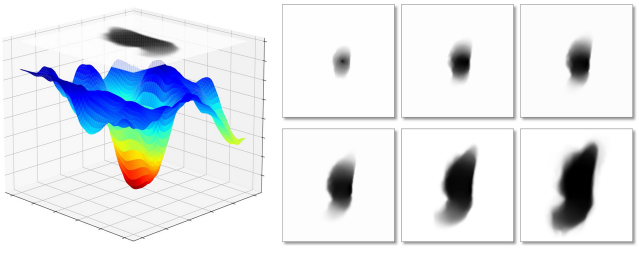


Fig. 5 Similarity-based fluid simulation. Left: fluid diffusion occurs according to the similarity map, shown as a curved surface. Right: penetration maps in the diffusion process.

清華大學出版社 Tsinghua University Press Available on IEEE Xplore®

$\boldsymbol{P}^0_{i,j}$ is given by the initial conditions. The calculation of $\boldsymbol{D}$ and the diffusion coefficient values on a staggered grid, such as $\boldsymbol{D}_{i+1/2,j}$, will be described in detail later. Eq. (4), with corresponding discretized initial conditions and boundary conditions, traversing all i and j, is a large sparse linear system whose coefficient matrix is symmetric and positive definite. We enforce the Neumann boundary condition by setting the diffusion coefficient $\boldsymbol{D}_{i,j}$ at the image boundary to zero. We use a standard conjugate gradient method [75] to solve this linear system quickly.

3.3.3 Diffusion coefficient

Calculation of the diffusion coefficient is formulated as Eq. (5):

$$\boldsymbol{D}_{i,j} = v\mathrm{e}^{-r(1-\boldsymbol{G}_{i,j})} \tag{5}$$

which has a nonlinear dependence on similarity. The main benefit of using this non-linear function is to diffuse nearly no concentration where similarity is relatively low to make a visible difference with isotropic diffusion. v is the diffusion velocity, controlling global diffusion effects. r is the resistance coefficient, controlling local diffusion effects. With higher r, the distribution of $\mathrm{e}^{-r(1-\boldsymbol{G}_{i,j})}$ becomes sharper, and the diffusion area shrinks; conversely as the distribution gets flatter, the diffusion area expands. An interpolation method is used to calculate the coefficient on the staggered grid:

$$\boldsymbol{D}_{i+1/2,j} = \min\left(\boldsymbol{D}_{i,j}, \boldsymbol{D}_{i+1,j}\right) \tag{6}$$

Other methods, such as linear interpolation and a maximum function, would achieve similar effects except for slight differences in diffusion at the border. All commutative functions conserve concentration.

3.3.4 Automatic termination

These diffusion processes continue all the time. Our IST method provides a manual way for users to stop diffusion at a specific moment when they are satisfied. However, when the users want to interact more naturally and efficiently, we also provide an automatic mode. When the rate of variation in concentration is less than a certain value ϵ, formulated as

$$\frac{\sum_{i,j}\left|\boldsymbol{P}^{n+1}_{i,j} - \boldsymbol{P}^{n}_{i,j}\right|}{\sum_{i,j}\boldsymbol{P}^{n}_{i,j}} \leqslant \epsilon \tag{7}$$

the fluid diffusion is stopped.

3.4 Local style transfer

3.4.1 Background

At present, most NST methods extract the style from a whole picture and transfer it to another picture. Partial style extraction is less studied. Since the classic method of NST, AdaIN [5], focuses on the mean and variance of neural features from content and style images, and these can be calculated locally, we modify the method to adapt it to local transfer.

3.4.2 Revisiting AdaIN

The AdaIN method is based on the premise that the style of the image is determined by the mean and variance of the deep features. This method first puts the corresponding content $\boldsymbol{C}$ and style $\boldsymbol{S}$ into the backbone network to get the deep features, $\boldsymbol{F}^c$ and $\boldsymbol{F}^s$. Then the content features are normalized according to their mean and variance, and finally restored using the mean and variance of the style features, so that the generated features $\boldsymbol{A}$ have the content of the content image and the style of the style image. The calculation is performed as Eq. (8):

$$\boldsymbol{A} = \sigma\left(\boldsymbol{F}^s\right)\left(\frac{\boldsymbol{F}^c - \mu\left(\boldsymbol{F}^c\right)}{\sigma\left(\boldsymbol{F}^c\right)}\right) + \mu\left(\boldsymbol{F}^s\right) \tag{8}$$

where μ and σ represent functions to calculate channel-wise mean and standard deviation, respectively. The generated features are fed into a decoder network and restored to the transferred image $\boldsymbol{T} = \mathrm{Decoder}(\boldsymbol{A})$.

3.4.3 Modified local AdaIN

We modify the original AdaIN to carry out local style transfer. For style extraction, we use the normalized penetration map $\boldsymbol{P}$ to calculate the weighted mean and variance of style features. This can be formulated as

$$\boldsymbol{A} = \tilde{\sigma}\left(\boldsymbol{F}^s, \boldsymbol{P}^s\right)\left(\frac{\boldsymbol{F}^c - \mu\left(\boldsymbol{F}^c\right)}{\sigma\left(\boldsymbol{F}^c\right)}\right) + \tilde{\mu}\left(\boldsymbol{F}^s, \boldsymbol{P}^s\right) \tag{9}$$

where $\tilde{\mu}$ and $\tilde{\sigma}$ represent functions to calculate the weighted mean and standard deviation as Eq. (10):

$$\tilde{\mu}\left(\boldsymbol{F}, \boldsymbol{P}\right) = \frac{\sum_{i,j}\boldsymbol{P}_{i,j}\cdot\boldsymbol{F}_{i,j}}{\sum_{i,j}\boldsymbol{P}_{i,j}} \tag{10}$$

$$\tilde{\sigma}\left(\boldsymbol{F}, \boldsymbol{P}\right) = \sqrt{\frac{\sum_{i,j}\boldsymbol{P}_{i,j}\cdot\left(\boldsymbol{F}_{i,j} - \tilde{\mu}\left(\boldsymbol{F}, \boldsymbol{P}\right)\right)^2}{\sum_{i,j}\boldsymbol{P}_{i,j}}} \tag{11}$$

Then we merge the transferred features and the original content features to get $\boldsymbol{M}$ and feed it into the decoder network to get the transferred image $\boldsymbol{T}$:

$$\boldsymbol{T} = \mathrm{Decoder}(\boldsymbol{M}), \quad \boldsymbol{M} = (1-\boldsymbol{P}^c)\cdot\boldsymbol{F}^c + \boldsymbol{P}^c\cdot\boldsymbol{A} \tag{12}$$

The neural network causes partial distortion of the image, even when the values in $\boldsymbol{P}$ are all equal to 0. Thus, when we generate the final output $\boldsymbol{O}$, we mix the original image and the transferred image $\boldsymbol{T}$ for the low-penetration area, as Eq. (13):

$$\boldsymbol{O} = \boldsymbol{C} \cdot \phi(\boldsymbol{R}) + \boldsymbol{T} \cdot (1 - \phi(\boldsymbol{R})) \tag{13}$$

$$\phi(\boldsymbol{R}) = \max\left(\frac{\boldsymbol{R} - \alpha}{1 - \alpha}, 0\right) \tag{14}$$

where $\boldsymbol{R}$ is the content retention map and is equal to $1 - \boldsymbol{P}^c$ when these is only one interaction, and ϕ is a function to calculate the specific proportion of the original content image with factor α.

The above equations show the case where there is only one style and content image, such as the cyan scribble "ɾ" in Fig. 1(c). Suppose we want to fuse some areas of multiple style images to create a new style, such as the purple scribble "ɾ" in Fig. 1(c). As long as these pixels and corresponding concentrations are regarded as a whole, the corresponding weighted mean and standard deviation can be obtained to represent the new style, as used in our IST method.

3.5 Mixing multiple styles

IST often applies multiple styles to different areas of the picture. These styles may come from one picture or multiple pictures, and the same style may also correspond to many regions. It is worth considering how to mix styles when the same pixel position is affected by multiple styles. We adopt a treatment method inspired by oil painting. When a new style and interaction occur, any subsequent style is mixed with previous ones, and the subsequent one plays a dominant role. From the perspective of the fluid, the previous concentration based on the fluid simulation algorithm is retained to a certain extent at these locations, while the subsequent diffusion results dominate. In terms of the specific implementation, we adopt an iterative calculation. The current mixed feature $\boldsymbol{M}^{[n]}$ is generated based on the transferred features $\boldsymbol{A}^{[n]}$ using the current style and the previously saved mixed features $\boldsymbol{M}^{[n-1]}$ according to the penetration map. This is formulated as

$$\boldsymbol{M}^{[n]} = (1 - \boldsymbol{P}^{[n]}) \cdot \boldsymbol{M}^{[n-1]} + \boldsymbol{P}^{[n]} \cdot \boldsymbol{A}^{[n]}, \quad \boldsymbol{M}^{[0]} = \boldsymbol{F}^c \tag{15}$$

Here, the initial mixed feature is the same as $\boldsymbol{F}^c$, and the feature gradually integrates more and more features of different styles. $\boldsymbol{P}$ in Eq. (15) means the penetration map of the content image and the index $[n]$ is used to indicate the interaction sequence. With increasing interactions on the content image, the content retention map is also gradually affected, as

$$\boldsymbol{R}^{[n]} = \boldsymbol{R}^{[n-1]} \cdot \left(1 - \boldsymbol{P}^{[n]}\right), \quad \boldsymbol{R}^{[0]} = \boldsymbol{1} \tag{16}$$

3.6 Interaction and user interface

3.6.1 Operations

For IST, there are two basic operations: dipping the style and painting on the content. We introduce these two operations using the user interface (UI) prototype shown in Fig. 6. The UI has two main parts: the left is the content canvas and the right is the style palette. The style palette may contain multiple style images, four are shown and used in the prototype. Users can click "•" and scribble "ɾ" for dipping and painting. We also provide a convenient operation to dip or paint on the whole image, shown as "★" in the upper left corner. This is implemented by a special action (clicking the middle mouse button).

3.6.2 Dipping the style

Different pictures have different styles. Even in the same picture, the styles in different positions may vary. For example, in a sketch, the stroke style and the hardness vary in different locations. The user clicks or scribbles on the specific area on one of the style images. The system calculates a penetration map based on the automatic fluid simulation algorithm based on the similarity maps. If the user wants to select two areas in two style images and merge them into a new style, they can take a specific action to combine the two interactions by holding the control key and performing interactions on multiple style images (Fig. 6④).

3.6.3 Painting on the content

For the extracted local style, the user can choose to add one or more interactions to the content image, which again can be clicks or scribbles. Unlike dipping on style, the user can choose manual or

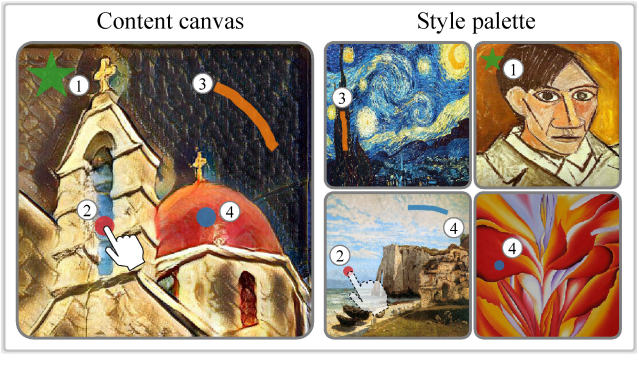


Fig. 6 User interface prototype. The interaction mode can be global ①, clicking ②, or scribbling ③. The operation ④ mixes styles in different locations and creates a new style. The interactions of the same color represent matching relationships between dipping and painting.

清華大學出版社 Tsinghua University Press Available on IEEE Xplore®

automatic generation of the penetration map. The automatic method works similarly to style dipping. After interaction, the fluid simulation algorithm calculates a stable state as the penetration map. For the manual approach, the user takes an additional action before the interaction, pressing the shift key. After the interaction, our method generates a corresponding transferred result according to the process of fluid diffusion, which looks like a style transfer that continuously diffuses outward from the interaction. When the user is satisfied, the shift key is released to stop diffusion. For the click interaction, users can directly control the style diffusion process by pressing and releasing the right mouse button.

3.7 Extension for general style transfer

Our method is based on changes to AdaIN [5], but with the continuing development of techniques, various stylization methods have emerged. We aim to apply interactive style transfer to methods in general, so we have designed some extended approaches. The general method selects the style based on the entire image. We make certain modifications to the dipping operation: after the user obtains the penetration map for dipping, we perform a binary segmentation on the map and select the smallest inscribed square. Then, we use the corresponding style image portion of this square as the style to apply to the entire content image. During the painting operation, we use the penetration map obtained through user interaction and blend it using the same blending strategy as our method for local stylization results. This extended method can be applied to any stylization technique, significantly enhancing its applicability, including transformer-based [76] and diffusion-based methods [41].

4 Experiments

4.1 Settings

For IST, α is set to 0.7. The user-adjustable parameter ϵ that controls the stopping point of the changes in the penetration map in automatic mode is set by default to 0.01. The default values for v and r are 1.0 and 10. In our specific implementation, AdaIN is trained as in Ref. [5], using MS-COCO [77] and WikiArt [78] as datasets. We adopt VGG-19 [73] as our backbone neural network, adopting the PyTorch [79] framework. The fluid simulation algorithm is based on compute unified device architecture (CUDA). All experiments were run on a single NVIDIA Titan XP GPU. Our fluid simulation method can produce real-time animation, achieving 174 frames/s on 512×512 images. After adding the style transfer method, it can still achieve 53 frames/s.

4.2 Qualitative results

4.2.1 What can dipping create?

In our IST method, we first dip the style by clicking or scribbling on style images. Compared to transferring the style of the whole reference image, our dipping process can create diverse works that match the user's intent. We show some examples in Fig. 7. In the first row, the content picture is the flower with a butterfly, and the style picture is a sketched portrait. Employing AdaIN [5], users cannot control

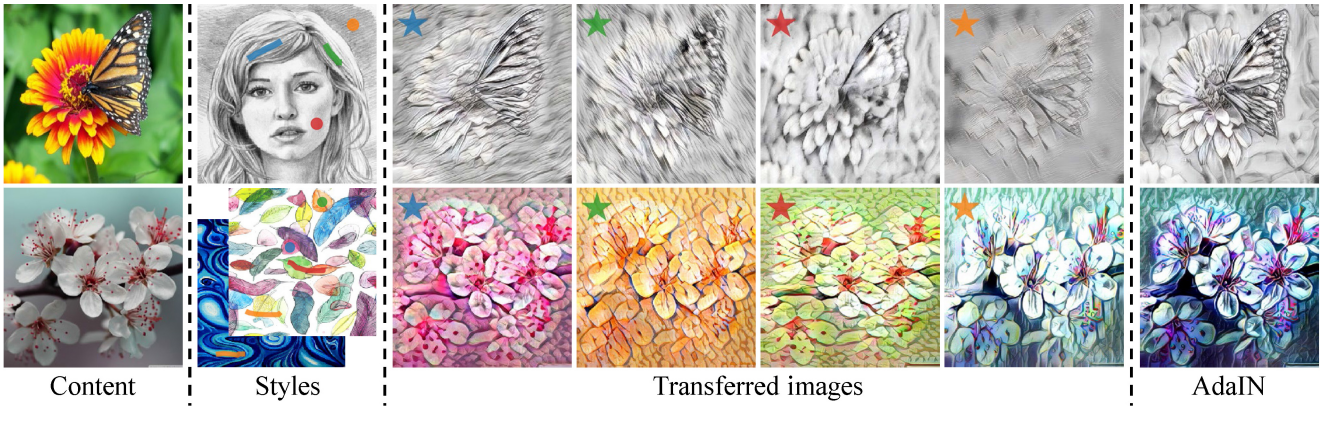


Fig. 7 Dipping from different regions of style images. In comparison to AdaIN, using our IST method, users can create diverse stylized results using limited style images. Above: examples of dipping in a single style. Users can create artworks with different texture directions and color concentrations according to their intent. Below: dipping operations merging styles from one or more images. The stylized results can generate various color co-occurences according to user intent.

清華大學出版社 Tsinghua University Press Available on IEEE Xplore®

the detailed style of the output; while, using IST, the output artworks can present different stroke directions and color concentrations. The second row shows more complex scenes. Users dip and merge styles from one or more images to create dazzling artworks with limited reference images.

4.2.2 What can painting create?

After dipping the style, we can paint by clicking or scribbling on content images. We illustrate some cases in Fig. 8. The first row contains a content image of buildings and a style image. We can dip a style and paint it onto different regions of the building. The second row shows the style flow from the interaction position with increasing holding time, gradually turning the grassland into a lake. The third row shows an interesting scene. We dip four different styles from two style images and paint them onto three flowers and the background. We exchange styles between them to produce a variety of artworks.

4.2.3 Artworks freely created by users

We invited various people to try out our IST method. Some of the resulting artworks created by their interactions and integrating a variety of styles are shown in Fig. 9.

4.3 Ablation and parameter study

4.3.1 Similarity-based fluid simulation

Final results of our method for a particular example are displayed in the second row of Fig. 8. In Fig. 10, we conduct ablation experiments on our similarity-based fluid simulation algorithm. When we do not use the similarity map to control the resistance in the

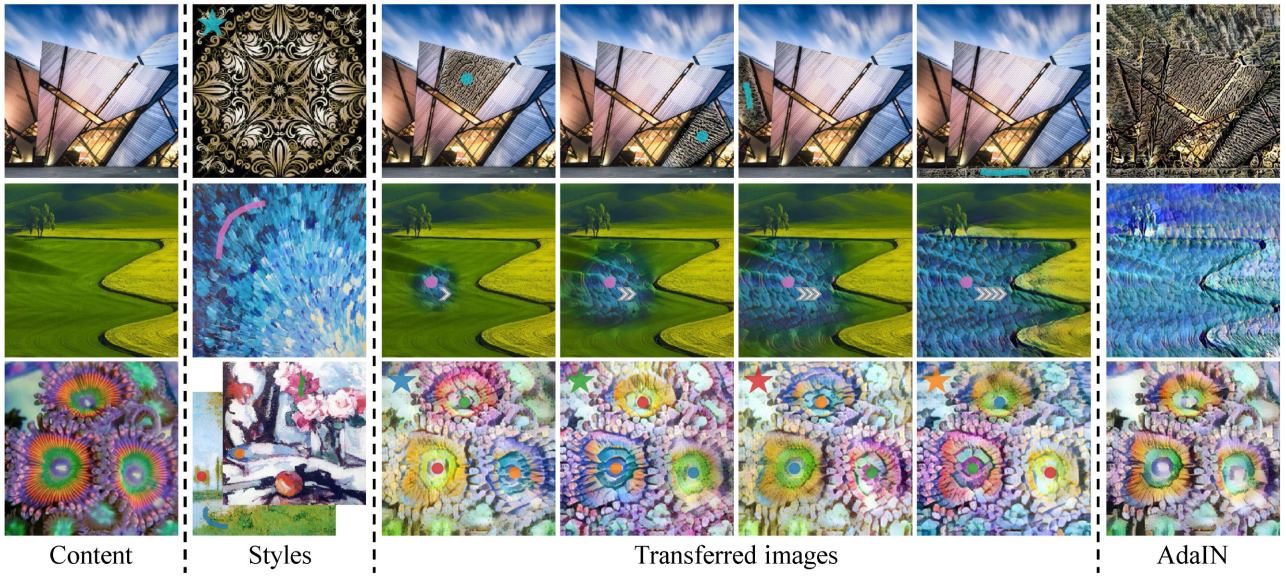


Fig. 8 Painting into different regions of content images. Above: transfer to different areas of the building. Middle: style flow from the interaction position with increasing holding time. Below: we dip four styles from style images and paint them on three flowers and the background to generate various results.



Fig. 9 Various artworks created by our IST method.

清華大學出版社 Tsinghua University Press Available on IEEE Xplore®

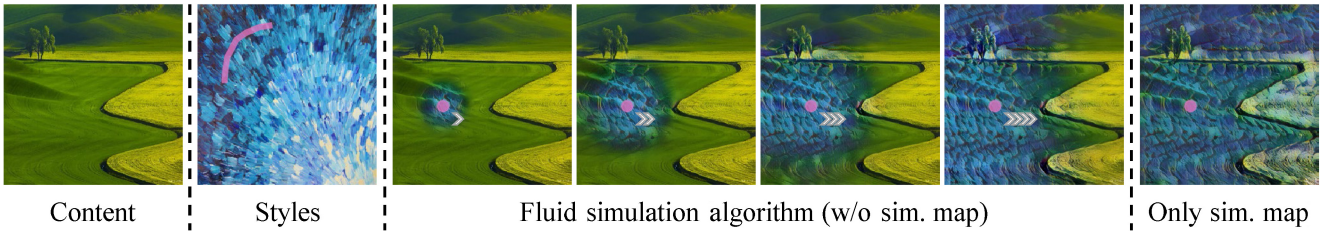


Fig. 10 Ablation study for the similarity-based fluid simulation.

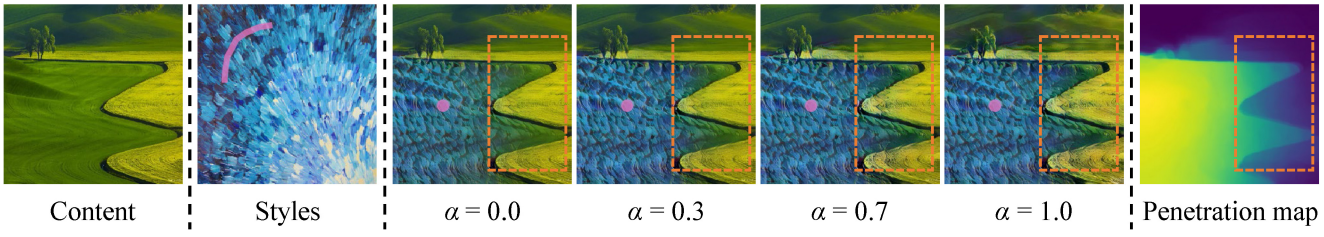


Fig. 11 Mixing results for different values of the mixing factor α.

fluid simulation algorithm, the fluid diffusion spreads circularly from the interaction points. Only when it encounters the external boundary of the image does it gradually change shape. Such a brush makes it difficult to efficiently stylize the target area. When we use the similarity map solely as the penetration map for stylization, the stylization applies to the entire image, making it difficult to control the target area with fine detail through stepwise fluid animation.

4.3.2 Mixing factor α

In Fig. 11, we provide the mixed results under different values for the mixing factor α. We can observe that the parameter α determines the degree of transition of the style image at the boundaries. For values in the range of 0 to $(1-\alpha)$ in the penetration map, interpolation with the original content image will occur. When α is small, most of the values undergo interpolation, while when α is large, only the areas with small values in the penetration map, primarily the outline regions, undergo interpolation. As shown in Fig. 11, for the areas marked by rectangles, as α increases, the intensity of stylization in these regions becomes stronger, and the transition zone becomes smaller.

4.3.3 Similarity map averaging strategy

In Fig. 12, we present the average similarity map of three different mixing ratios with s. When $s = 0.0$, we primarily use low-level features. When $s = 0.5$, we use the average features. When $s = 1.0$, we primarily use high-level features. In the case of similarity maps, the low-level features of the image are more focused on attributes like color, while the high-level features emphasize semantics. As shown in Fig. 12, if we use a low-level similarity map, it may quickly diffuse to areas with similar colors, such as spreading from the sail to the sky. On the other hand, a semantic-based similarity map limits the diffusion to a smaller region. Therefore, considering their average can help alleviate the issues of both to some extent. However, since each user's needs may vary, there is no one-fit-all similarity map.

4.3.4 Fluid simulation parameters

We vary the parameters v and r in our similarity-based fluid simulation algorithm, showing the animation sequence in Fig. 13. On the left, as the diffusion velocity increases, it can be seen that the

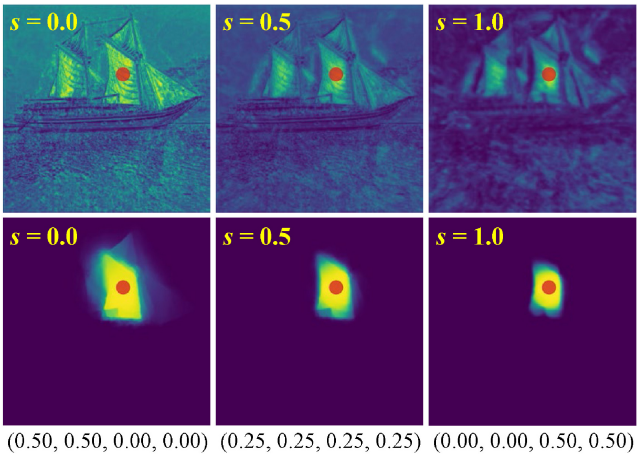


Fig. 12 Results with different similarity map averaging strategies. Above: weighted average similarity map. Below: corresponding penetration maps when for 100 animation steps.

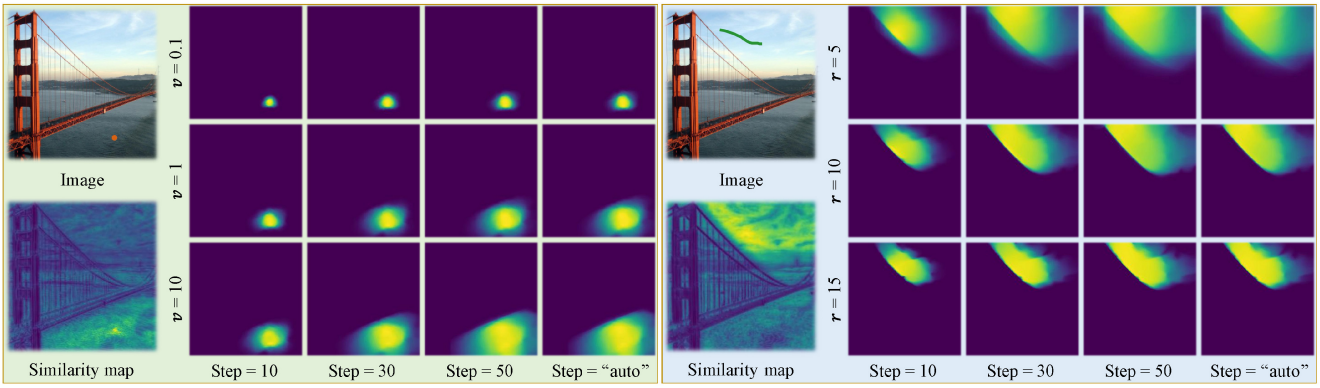


Fig. 13 Parameters v and r in the similarity-based fluid simulation algorithm are the diffusion velocity and resistance coefficient, respectively. We show animations for 10, 30, and 50 steps, and automatic steps.

diffusion speed of the fluid increases significantly. In addition, due to the increase in diffusion velocity, the difference between steps becomes larger, affecting the automatic results. On the right, we change the resistance coefficient, whose effect is negatively related to the diffusion velocity to a certain extent. However, as the resistance coefficient increases, the more difficult it is to diffuse on the border, and the more obvious the border becomes.

4.4 Method comparisons

4.4.1 Comparison to hard segmentation

In Fig. 14, we present an experiment comparing our method to one using hard segmentation. We apply an interactive segmentation method (FCANet [80]) to segment a specified region of the sky. When the segmentation mask is obtained directly by clicking (as in this method), the following drawbacks arise: (i) it is difficult for the user to only stylize a specific part of the sky, as shown in this figure, and (ii) stylization at the boundaries of the hard mask lacks a smooth transition. In contrast, our IST allows users to gradually diffuse the sky through animation, similar to how dye spreads in real painting, providing more intuitive interaction and finer control.

4.4.2 Extension for various stylization methods

In Section 3.7, we presented ways to extend our method for use with other general stylization methods. In Fig. 15, we show results of using the extended IST version with various stylization methods, including ArtFlow [29], IEContraAST [30], AdaAttN [81], StyleFormer [76], and InST [41]. The IST results are in the second row. We can observe that users can select local styles from the style image, such as its upper part, to stylize the target image. Ultimately, the quality of the final stylization depends on the algorithm itself. The results in the first row are provided by the original versions of these methods with global style transfer. By comparing the results of the two rows, it is evident that global style transfer achieves distinct light and dark features. In contrast, local style transfer provided by IST using only to

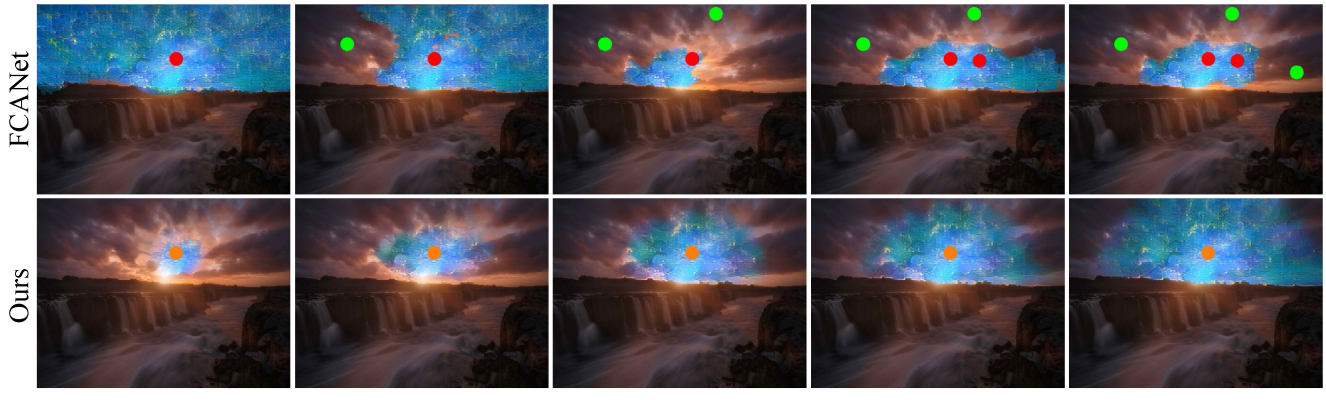


Fig. 14 Comparison of our method to use of hard segmentation. Above: result using interactive hard segmentation. Below: results from our method.

清華大學出版社 Tsinghua University Press Available on IEEE Xplore®



Fig. 15 Using the extended IST version with various stylization methods.

the upper portion of the style image produces results with a noticeably lighter tone, more closely aligned with the user's interactively selected intent.

4.4.3 Quantitative comparison

Since our extended method provides an interactive version of other stylization techniques, the style quality of the generated image depends on the characteristics of these underlying methods, and these related style metrics are specifically compared in their own papers. In Table 1, we employ subjective rating for a quantitative comparison. We collected 10 pairs of images stylized by the original version and the IST version. We conducted a user study involving 30 participants, who were asked to rate the images based on four criteria across two dimensions, with scores ranging from 1 to 5. The first dimension was style similarity, in terms of color and texture resemblance of the stylized image to the style around the user interaction. The second dimension considered aesthetics, assessing the diversity of the style and the overall appeal of the results to the users. From the results, it is evident that our method, compared to the original version, achieves closer similarity in both color and texture to the user's interactive input. Additionally, users found our approach to offer greater style diversity. As for user preference scores, both methods performed similarly, in agreement with expectations. Some users preferred the uniform style of the original version, while others favored more diverse styles. This balance reflects the varying tastes and preferences of users.

Table 1 User study: subjective rating of style similarity and aesthetics

Version	Style similarity		Aesthetics	
	Color	Texture	Diversity	Preference
Original	2.37	3.13	2.23	4.17
IST	4.73	4.30	3.93	4.23

4.5 Usage study

4.5.1 Aims

In order to better analyze our IST method from a user perspective, we invited 20 users to try out our system and record their opinions. These data are presented in Table 2, mainly focusing on the perceived benefit and frequency of usage of the three interaction operations (global, click, and scribble). In Table 2, these three types of operations are divided in detail into 12 sub operations, with benefit and usage frequency assigned to three levels.

4.5.2 Global operations

Global operations allow users to dip and paint styles from the entire style or content images. They can be also used to mix the overall styles of multiple style images. Table 2 shows that most users believe that these operations can bring benefits, but they are more inclined to choose to directly dip from a style image and directly paint into a content image.

4.5.3 Click operations

Click operations allow users to dip and paint styles from or to local areas on style or content images. They can be also used to mix many styles from multiple click areas. From Table 2, it can be seen that users tend to select one local style rather than mixed styles. Most users believe that the manual is more helpful, but in terms of actual usage frequency, the two are similar.

4.5.4 Scribble operations

Scribble operations are similar to click operations, with the additional feature that users can quickly mix styles from multiple regions through a long scribble.

清華大學出版社 Tsinghua University Press Available on IEEE Xplore®

Table 2 User study of various interaction operations: perceived benefit and usage frequency

	Operation	Benefit			Frequency		
		Low	Medium	High	Low	Medium	High
★	Dip from one style image	5%	25%	70%	10%	20%	70%
	Dip and mix from multiple style images	25%	35%	40%	40%	40%	20%
	Paint into the content image	5%	5%	90%	10%	15%	75%
•	Dip from one click area	0%	25%	75%	0%	25%	75%
	Dip and mix from multiple click areas	15%	55%	30%	20%	50%	30%
	Paint into a local click area (automatic)	5%	25%	70%	5%	45%	50%
	Paint into a local click area (manual)	5%	0%	95%	5%	45%	50%
≀	Dip from one scribble area	0%	45%	55%	0%	55%	45%
	Dip and mix from a cross style area with a long scribble	15%	60%	25%	30%	45%	25%
	Dip and mix from multiple scribble areas	20%	40%	40%	30%	50%	20%
	Paint into a local scribble area (automatic)	0%	30%	70%	0%	40%	60%
	Paint into a local scribble area (manual)	5%	25%	70%	10%	25%	65%

From Table 2, it can be seen that the proportion of users who believe that scribbles can bring great benefits is lower than for clicks, and their usage frequency is also lower than for clicks. This may be because the interaction mode of scribbling is not as fast as clicking. Most users believe that automatic and manual methods are both important. However, in practical use, some users are not very fond of using the manual method for scribbles because such interactive operations may be complex and cannot be achieved by long pressing the mouse as for a click.

5 Conclusions

In this paper, we have brought a new creative dimension to neural style transfer. It breaks the whole image-to-image transfer limitation and formulates NST as a drawing process. Users can dip a style from one or multiple style images and then paint it onto any region of the content image. We call this creation process interactive style transfer (IST). In our IST method, we employ a fluid simulation algorithm to simulate the style dipping and painting process using a brush. Our paper has also demonstrated many interesting artworks using our IST method.

Author contributions

Zheng Lin was responsible for the main coding and writing. Zhao Zhang was responsible for polishing the writing. Kang-Rui Zhang was responsible for the coding of fluid simulation. Bo Ren and Ming-Ming Cheng were responsible for project discussions. All the authors have approved the final manuscript.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Project No. 62495060), the Fundamental Research Funds for the Central Universities (Nankai University, Grant No. 63253233), the Research Grant of Tsinghua–Tencent Joint Laboratory for Internet Innovation Technology, the China Postdoctoral Science Foundation (Grant Nos. 2025T180422, 2024M761682, GZB20240357), and Shui Mu Tsinghua Scholar (Grant No. 2024SM079).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article. The author Ming-Ming Cheng is the Area Executive Editor of this journal.

References

[1] Kotovenko, D.; Sanakoyeu, A.; Ma, P.; Lang, S.; Ommer, B. A content transformation block for image style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10024–10033, 2019.

[2] Cheng, J.; Jaiswal, A.; Wu, Y.; Natarajan, P.; Natarajan, P. Style-aware normalized loss for improving arbitrary style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 134–143, 2021.

[3] Chiu, T. Y. Understanding generalized whitening and coloring transform for universal style transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 4451–4459, 2019.

[4] Wang, P.; Li, Y.; Vasconcelos, N. Rethinking and improving the robustness of image style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 124–133, 2021.

[5] Huang, X.; Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision, 1510–1519, 2017.

[6] Chandran, P.; Zoss, G.; Gotardo, P.; Gross, M.; Bradley, D. Adaptive convolutions for structure-aware style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7968–7977, 2021.

[7] Liu, X. C.; Yang, Y. L.; Hall, P. Learning to warp for style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3701–3710, 2021.

[8] Kotovenko, D.; Wright, M.; Heimbrecht, A.; Ommer, B. Rethinking style transfer: From pixels to parameterized brushstrokes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12191–12200, 2021.

[9] Gatys, L. A.; Ecker, A. S.; Bethge, M.; Hertzmann, A.; Shechtman, E. Controlling perceptual factors in neural style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3730–3738, 2017.

[10] Kurzman, L.; Vazquez, D.; Laradji, I. Class-based styling: Real-time localized style transfer with semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop, 3189–3192, 2019.

[11] Men, Y.; Lian, Z.; Tang, Y.; Xiao, J. A common framework for interactive texture transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6353–6362, 2018.

[12] Virtusio, J. J.; Talavera, A.; Tan, D. S.; Hua, K. L.; Azcarraga, A. Interactive style transfer: Towards styling user-specified object. In: Proceedings of the IEEE Visual Communications and Image Processing, 1–4, 2018.

[13] Stahl, F.; Meyer, M.; Schwanecke, U. IST-style transfer with instance segmentation. In: Proceedings of the 11th International Symposium on Image and Signal Processing and Analysis, 277–281, 2019.

[14] Chen, D.; Yuan, L.; Liao, J.; Yu, N.; Hua, G. StyleBank: An explicit representation for neural image style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2770–2779, 2017.

[15] Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; Yang, M. H. Advances in neural information processing systems. In: Proceedings of the Advances in Neural Information Processing Systems, 385–395, 2017.

[16] Lin, T.; Ma, Z.; Li, F.; He, D.; Li, X.; Ding, E.; Wang, N.; Li, J.; Gao, X. Drafting and revision: Laplacian pyramid network for fast high-quality artistic style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5137–5146, 2021.

[17] Johnson, J.; Alahi, A.; Li, F. F. Perceptual losses for real-time style transfer and super-resolution. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9906.* Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 694–711, 2016.

[18] Ulyanov, D.; Lebedev, V.; Vedaldi, A.; Lempitsky, V. S. Texture networks: Feed-forward synthesis of textures and stylized images. In: Proceedings of the 33rd International Conference on Machine Learning, 1349–1357, 2016.

[19] Li, C.; Wand, M. Precomputed real-time texture synthesis with Markovian generative adversarial networks. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9907.* Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 702–716, 2016.

[20] Ulyanov, D.; Vedaldi, A.; Lempitsky, V. 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, 4105–4113, 2017.

[21] Wei, Y.; Zheng, Y.; Zhang, Y.; Liu, M.; Ji, Z.; Zhang, L.; Zuo, W. Personalized image generation with deep generative models: A decade survey. *Computational Visual Media* Vol. 11, No. 6, 1141–1194, 2025.

[22] Liu, F.; Chang, X. IIDM: Image-to-image diffusion model for semantic image synthesis. *Computational Visual Media* Vol. 11, No. 2, 423–429, 2025.

[23] Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; Yang, M. H. Diversified texture synthesis with feed-forward networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 266–274, 2017.

[24] Wang, X.; Oxholm, G.; Zhang, D.; Wang, Y. F. Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7178–7186, 2017.

[25] Zhang, H.; Dana, K. Multi-style generative network for real-time transfer. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11132.*

Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 349–365, 2019.

[26] Sheng, L.; Lin, Z.; Shao, J.; Wang, X. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8242–8250, 2018.

[27] Park, D. Y.; Lee, K. H. Arbitrary style transfer with style-attentional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5873–5881, 2019.

[28] Svoboda, J.; Anoosheh, A.; Osendorfer, C.; Masci, J. Two-stage peer-regularized feature recombination for arbitrary image style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 13813–13822, 2020.

[29] An, J.; Huang, S.; Song, Y.; Dou, D.; Liu, W.; Luo, J. ArtFlow: Unbiased image style transfer via reversible neural flows. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 862–871, 2021.

[30] Chen, H.; Zhao, L.; Wang, Z.; Zhang, H.; Zuo, Z.; Li, A.; Xing, W.; Lu, D. Artistic style transfer with internal-external learning and contrastive learning. In: Proceedings of the 35th International Conference on Neural Information Processing Systems, 26561–26573, 2021.

[31] Deng, Y.; Tang, F.; Dong, W.; Ma, C.; Pan, X.; Wang, L.; Xu, C. StyTr2: Image style transfer with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11316–11326, 2022.

[32] Zhang, Y.; Tang, F.; Dong, W.; Huang, H.; Ma, C.; Lee, T. Y.; Xu, C. Domain enhanced arbitrary image style transfer via contrastive learning. In: Proceedings of the SIGGRAPH Conference, 1–8, 2022.

[33] Cheng, B.; Liu, Z.; Peng, Y.; Lin, Y. General image-to-image translation with one-shot image guidance. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 22679–22689, 2023.

[34] Wang, Q.; Li, S.; Wang, Z.; Zhang, X.; Feng, G. Multi-source style transfer via style disentanglement network. *IEEE Transactions on Multimedia* Vol. 26, 1373–1383, 2024.

[35] Deng, Y.; He, X.; Tang, F.; Dong, W. Z*: Zero-shot style transfer via attention reweighting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6934–6944, 2024.

[36] Gatys, L. A.; Ecker, A. S.; Bethge, M. Image style transfer using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2414–2423, 2016.

[37] Li, C.; Wand, M. Combining Markov random fields and convolutional neural networks for image synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2479–2486, 2016.

[38] Li, S.; Xu, X.; Nie, L.; Chua, T. S. Laplacian-steered neural style transfer. In: Proceedings of the 25th ACM International Conference on Multimedia, 1716–1724, 2017.

[39] Ghiasi, G.; Lee, H.; Kudlur, M.; Dumoulin, V.; Shlens, J. Exploring the structure of a real-time, arbitrary neural artistic stylization network. In: Proceedings of the British Machine Vision Conference, Article No. 114, 2017.

[40] Shen, F.; Yan, S.; Zeng, G. Neural style transfer via meta networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8061–8069, 2018.

[41] Zhang, Y.; Huang, N.; Tang, F.; Huang, H.; Ma, C.; Dong, W.; Xu, C. Inversion-based style transfer with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10146–10156, 2023.

[42] Yang, S.; Hwang, H.; Ye, J. C. Zero-shot contrastive loss for text-guided diffusion image style transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 22816–22825, 2023.

[43] Wang, Z.; Zhao, L.; Xing, W. StyleDiffusion: Controllable disentangled style transfer via diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 7643–7655, 2023.

[44] Zhang, Z.; Zhang, Q.; Xing, W.; Li, G.; Zhao, L.; Sun, J.; Lan, Z.; Luan, J.; Huang, Y.; Lin, H. ArtBank: Artistic style transfer with pre-trained diffusion model and implicit style prompt bank. *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 38, No. 7, 7396–7404, 2024.

[45] Huang, Y.; Iizuka, S.; Simo-Serra, E.; Fukui, K. Controllable multi-domain semantic artwork synthesis. *Computational Visual Media* Vol. 10, No. 2, 355–373, 2024.

[46] Liu, Y.; Zheng, H.; Yang, S. SAD: Style-aware diffusion adaptation for few-shot style transfer image generation. *Computational Visual Media* Vol. 11, No. 4, 889–895, 2025.

[47] Zhao, Y.; Guo, D.; Lian, Z.; Gao, Y.; Han, J.; Feng, J.; Wang, G.; Zhou, B.; Li, S. Neural-polyptych: Content controllable painting recreation for diverse genres. *Computational Visual Media* Vol. 11, No. 3, 635–654, 2025.

[48] Maejima, A.; Shinagawa, S.; Kubo, H.; Funatomi, T.; Yotsukura, T.; Nakamura, S.; Mukaigawa, Y.

Continual few-shot patch-based learning for anime-style colorization. *Computational Visual Media* Vol. 10, No. 4, 705–723, 2024.

[49] Zhou, Y.; Li, S.; Huang, H. Deformable few-shot face cartoonization via local to global translation. *Computational Visual Media* Vol. 11, No. 2, 269–287, 2025.

[50] Lee, S. H.; Kim, S.; Byeon, W.; Oh, G.; In, S.; Park, H.; Yoon, S. H.; Hong, S. H.; Kim, J.; Kim, S. Audio-guided implicit neural representation for local image stylization. *Computational Visual Media* Vol. 10, No. 6, 1185–1204, 2024.

[51] Kwon, G.; Ye, J. C. CLIPstyler: Image style transfer with a single text condition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 18041–18050, 2022.

[52] Fu, T. J.; Wang, X. E.; Wang, W. Y. Language-driven artistic style transfer. In: *Computer Vision – ECCV 2022. Lecture Notes in Computer Science, Vol. 13696*. Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; Hassner, T. Eds. Springer Cham, 717–734, 2022.

[53] Chung, J. J. Y.; Adar, E. PromptPaint: Steering text-to-image generation through paint medium-like interactions. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, Article No. 6, 2023.

[54] Liu, L.; Fu, Q.; Hou, F.; He, Y. Intuitive user-guided portrait image editing with asymmetric conditional GAN. *Computational Visual Media* Vol. 11, No. 2, 361–379, 2025.

[55] Wen, H.; You, S.; Fu, Y. LucIE: Language-guided local image editing for fashion images. *Computational Visual Media* Vol. 11, No. 1, 179–194, 2025.

[56] Xu, Y.; Pinkney, J. N. M.; Yang, Y. L.; Shao, T.; Zhou, K. Emotion amplification of facial videos using a fine-tuned StyleGAN. *Computational Visual Media* Vol. 11, No. 3, 587–601, 2025.

[57] Virtusio, J. J.; Ople, J. J. M.; Tan, D. S.; Tanveer, M.; Kumar, N.; Hua, K. L. Neural style palette: A multimodal and interactive style transfer from a single style image. *IEEE Transactions on Multimedia* Vol. 23, 2245–2258, 2021.

[58] Akimoto, N.; Zhu, H.; Jin, Y.; Aoki, Y. Fast soft color segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8274–8283, 2020.

[59] Alegre, L. N.; Oliveira, M. M. SelfieArt: Interactive multi-style transfer for selfies and videos with soft transitions. In: Proceedings of the 33rd SIBGRAPI Conference on Graphics, Patterns and Images, 17–22, 2020.

[60] Dubey, M.; Otto, J.; Forbes, A. G. Data brushes: Interactive style transfer for data art. In: Proceedings of the IEEE VIS Arts Program, 1–9, 2019.

[61] Li, S.; Pan, Y. An efficient and harmonized framework for balanced cross-domain feature integration. *arXiv preprint* arXiv:2403.18461, 2024.

[62] Maninis, K. K.; Caelles, S.; Pont-Tuset, J.; Van Gool, L. Deep extreme cut: From extreme points to object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 616–625, 2018.

[63] Reimann, M.; Buchheim, B.; Semmo, A.; Dollner, J.; Trapp, M. Interactive multi-level stroke control for neural style transfer. In: Proceedings of the International Conference on Cyberworlds, 1–8, 2021.

[64] Warner, J.; Kim, K. W.; Hartmann, B. Interactive flexible style transfer for vector graphics. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, Article No. 49, 2023.

[65] Wang, X.; Xu, Y.; Liu, S.; Ren, B.; Kosinka, J.; Telea, A. C.; Wang, J.; Song, C.; Chang, J.; Li, C.; et al. Physics-based fluid simulation in computer graphics: Survey, research trends, and challenges. *Computational Visual Media* Vol. 10, No. 5, 803–858, 2024.

[66] Witting, P. Computational fluid dynamics in a traditional animation environment. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, 129–136, 1999.

[67] Stomakhin, A.; Selle, A. Fluxed animated boundary method. *ACM Transactions on Graphics* Vol. 36, No. 4, Article No. 68, 2017.

[68] Fattal, R.; Lischinski, D. Target-driven smoke animation. *ACM Transactions on Graphics* Vol. 23, No. 3, 441–448, 2004.

[69] Shi, L.; Yu, Y. Taming liquids for rapidly changing targets. In: Proceedings of the ACM SIGGRAPH/ Eurographics Symposium on Computer Animation, 229–236, 2005.

[70] Pan, Z.; Manocha, D. Efficient solver for spacetime control of smoke. *ACM Transactions on Graphics* Vol. 36, No. 4, Article No. 1, 2017.

[71] Perona, P.; Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 12, No. 7, 629–639, 1990.

[72] Bau, D.; Zhou, B.; Khosla, A.; Oliva, A.; Torralba, A. Network dissection: Quantifying interpretability of deep visual representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3319–3327, 2017.

[73] Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In: Proceedings of the International Conference on Learning Representations, 2015.

[74] Dehghan, M. Fully implicit finite differences methods for two-dimensional diffusion with a non-local boundary condition. *Journal of Computational and Applied Mathematics* Vol. 106, No. 2, 255–269, 1999.

[75] Bolz, J.; Farmer, I.; Grinspun, E.; Schröder, P. Sparse matrix solvers on the GPU: Conjugate gradients and multigrid. *ACM Transactions on Graphics* Vol. 22, No. 3, 917–924, 2003.

[76] Wu, X.; Hu, Z.; Sheng, L.; Xu, D. StyleFormer: Real-time arbitrary style transfer via parametric style composition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 14598–14607, 2021.

[77] Lin, T. Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. L. Microsoft COCO: Common objects in context. In: *Computer Vision – ECCV 2014. Lecture Notes in Computer Science, Vol. 8693*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 740–755, 2014.

[78] Nichol, K. Painter by numbers, wikiart. 2016. Available at https://www.kaggle.com/c/painter-by-numbers

[79] Steiner, B.; DeVito, Z.; Chintala, S.; Gross, S.; Paszke, A.; Massa, F.; Lerer, A.; Chanan, G.; Lin, Z.; Yang, E.; et al. PyTorch: An imperative style, high-performance deep learning library. In Proceedings of the 33rd Conference on Neural Information Processing Systems, 8024–8035, 2019.

[80] Lin, Z.; Zhang, Z.; Chen, L. Z.; Cheng, M. M.; Lu, S. P. Interactive image segmentation with first click attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 13336–13345, 2020.

[81] Liu, S.; Lin, T.; He, D.; Li, F.; Wang, M.; Li, X.; Sun, Z.; Li, Q.; Ding, E. AdaAttN: Revisit attention mechanism in arbitrary neural style transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 6629–6638, 2021.



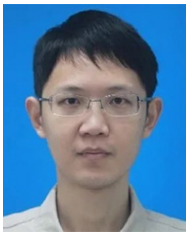
Zheng Lin is currently a postdoctoral researcher at Tsinghua University, under the supervision of Prof. Shi-Min Hu. He received a doctoral degree from Nankai University in 2023, under the supervision of Prof. Ming-Ming Cheng. His research interests are in computer vision and graphics.



Zhao Zhang is currently working as a vision-language researcher at ByteDance, with a focus on multimodal LLMs and their applications. He completed his master degree at Nankai University under the supervision of Prof. Ming-Ming Cheng.



Kang-Rui Zhang is currently a Ph.D. candidate in the College of Computer Science at Nankai University, under the supervision of Prof. Bo Ren. His research interest lies in computer graphics with fluid simulation.



Bo Ren received his Ph.D. degree from Tsinghua University in 2015. He is currently an associate professor in the College of Computer Science, Nankai University. His research interests include physically-based simulation and 3D scene reconstruction.



Ming-Ming Cheng is a professor in the College of Computer Science, Nankai University, leading the Media Computing Lab. He received his Ph.D. degree from Tsinghua University in 2012 and then worked with Prof. Philip Torr in Oxford for two years. His research interests include computer vision and computer graphics. He has published over 100 papers in leading journals and conferences, and many of his algorithms have become popular, receiving more than 40,000 citations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit `http://creativecommons.org/licenses/by/4.0/`.

To submit a manuscript, please go to `https://jcvm.org`.