Digital Art Image and Animation Compositing Technology Based on Computer-aided Drawing Algorithms

JianHui Yang[1], Yuan Shen[2]
[1]School of Art and Design, Jiangsu Ocean University, Lianyungang 222000, Jiangsu, China
[2]Department of Internal Medicine, Muwei Hospital, Lianyungang 222213, Jiangsu, China
Corresponding author: JianHui Yang (e-mail: jouyangjh1987@163.com)

ABSTRACT Addressing the lack of temporal coherence of brushstroke particles, brushstroke misalignment in moving regions, and degradation of stylistic details in digital-art image animation compositing, this paper proposes a digital-art animation compositing method based on optical-flow-guided brushstroke registration and dual-domain adaptive constraints. First, geometric contours and density-transparency attribute fields of brushstrokes are jointly extracted, and rigid temporal constraints are introduced for static regions to suppress texture jitter caused by random sampling. A style-preserving weight parameter is used to adaptively adjust the penalty intensity according to local texture entropy, distinguishing random high-frequency noise from intentional artistic vibration. Second, a multi-scale semantic style encoder is constructed, and global color and texture style consistency are maintained through Gram-matrix statistical constraints. Finally, an optical-flow-guided brushstroke flow field and motion-heatmap attenuation strategy are designed to achieve intelligent following and natural redrawing of brushstrokes under complex non-rigid motion. Experiments on the WikiArt dataset and DAVIS 2023 benchmark show that the method improves temporal coherence, reduces perceptual flicker, lowers endpoint error and FID, and shortens inference time, achieving smooth motion transitions while preserving artistic style.

INDEX TERMS computer graphics, non-photorealistic rendering, frame interpolation, optical flow, visual signal processing

I. INTRODUCTION

Generating dynamic representations of digital artwork through animated sequences with narrative tension is an exciting new area of research between digital media art and computer graphics. The ability to convert stills of illustrations, concept art, and stylized media into animated sequences will add significant visual impact and narrative expressiveness when distributing this type of content [1-2]. One relatively new way to create animation from still images is to create intermediate transition frames between single keyframes. This method is being utilised widely because it requires minimal amount of data and is applicable for many uses [3-4]. Unfortunately, existing methods to synthesise animation mainly focus on natural video frames as their input and are optimised for pixel-level accuracy in reconstructing images, and smoothing out the motion vectors between them [5-6]. Therefore, they do not consider the unique "brushstroke drawing units" of digital paintings (e.g., hand-drawn brushstrokes may be drawn in random directions) [7], or the way in which watercolours will diffuse or blur at their edges [8], or the way that oil paint might stack in 3D [9]. Animation generated using these current methods will encounter three primary challenges when applied to artistic images:

Temporal coherence within individual brushstrokes is inadequate. Current frame-based interpolation or rendering algorithms do not impose any restrictions/dependencies on either the distribution, orientation and/or opacity of brushstrokes between two adjacent frame's intermediate frames. In the absence of these restrictions, high-frequency flicker, edge crawl, and jitter can occur in brushstroke textures possessing stable appearances in static works of art during animation playback [10-11]. The flicker manifested by these textures arises from an internal random sample of brushstrokes utilized in the brushstroke texture generation process and the independent optimization at the frame level rather than from image noise at the pixel level.

The use of an inter-frame smoothing filter or a pixel-based blending operation of two adjacent frames is commonly employed in order to address the flicker that is present when two adjacent frame's brushstroke textures. By utilizing inter-frame smoothing filters and pixel-based blending operations, the noise is removed; however the sharpness of the brushstroke edge or the layering of the brushstrokes is also reduced. Due to the inherent contradiction between suppressing the temporal flicker and retaining the original

artistic quality of the individual brushstrokes, there will always be an impact on the ability to maintain the quality of the brushstroke texture during animation playback [12-13].

When there is non-linear motion present in an image, current techniques cannot accurately determine how brushstrokes will follow the expected motion over time. Even though optical flow estimation generates pixel level estimates of motion vector displacement, with respect to brushstrokes being macroscopic forms that exhibit directional properties and occupy area, it can produce significantly more complex overall deformation patterns than that of individual pixel displacements. As a result, attempting to simply "paste" or transfer brushstrokes from a previous frame to the next frame with reference to the optical flow field will often cause visual issues, such as tearing off sections of texture, distortion of brushstrokes, or duplicate mapping being created at the boundaries of frames. This issue is even more evident in scenes which contain more complex motions [14-15].

The three challenges described above all refer back to one center issue and uncertainty: the contradiction of having to achieve three properties simultaneously, namely; inter-frame physical correspondence of brushstrokes across frames; high fidelity containment of style details of a brushstroke within a single frame; and intelligently following the flow of brushstrokes over time in areas with motion. Most previous studies have focused on using improved optical flow networks to better define boundaries in motion [16-17], but still do not provide an explicit temporal constraint to the internal structure of brushstroke(s) which could eliminate the style loss (issue number two) and have drawn on the concept of using image style transfer as a means to introduce the necessary Gram matrix statistics that would create a degree of texture distribution consistency [18-19]. However, these statistical global constraints can, in some instances, provide an inter-frame correspondence and a following for brushstrokes, acting as separate entities, with respect to the flow of brushstrokes; other studies have tried to combine vector graphics representation with parametric interpolation methods but do have inherent limits with respect to the representation of complex pigment textures or non-rigid deformations, and also often yield unintentional smoothness effects.

To address the aforementioned challenges, existing research either focuses on pixel-level motion smoothing while neglecting the physical properties of brushstrokes as macroscopic drawing units, or relies on global style statistics, failing to achieve individual brushstroke tracking and adaptive deformation. There is a lack of a complete technical framework that organically unifies brushstroke temporal consistency, motion following, and style fidelity. To address this, this paper proposes a digital art image animation synthesis technique based on optical flow-guided brushstroke registration and dual-domain adaptive constraints. This technique addresses the aforementioned contradictions from three dimensions: brushstroke attribute modeling, style feature encoding, and motion deformation guidance. Firstly, a joint extraction framework for brushstroke geometric contours and density-transparency attribute fields is constructed, which explicitly models the alpha channel transparency and particle density characteristics of digital brushstrokes. Optical flow alignment establishes the inter-frame physical correspondence of brushstroke particles, and rigid temporal constraints in static areas are introduced to fundamentally sever the high-frequency flickering link caused by random sampling. The penalty intensity of the temporal constraint is dynamically modulated by the local texture entropy of the brushstroke region. Regions with high texture entropy corresponding to intentional stylistic texture fluctuations receive reduced constraint weights to preserve artistic vitality, while regions with low texture entropy dominated by sampling noise receive full constraint weights to eliminate flickering artifacts. Secondly, a multi-scale semantic style encoder based on a pre-trained VGG-19 (Visual Geometry Group 19-layer network) is employed to simultaneously extract brushstroke micro-texture and global color composition features at different receptive field levels. Gram matrix statistical constraints ensure high-fidelity restoration of the artistic style in a single frame. Thirdly, an optical flow-guided brushstroke flow field and motion heatmap mask attenuation strategy are designed. In areas of weak motion, brushstrokes are forced to undergo affine deformation alignment along their trajectory. In areas of strong motion, the consistency penalty is dynamically relaxed based on a continuous attenuation function of optical flow amplitude to trigger natural brushstroke redrawing, solving the problems of brushstroke misalignment and texture tearing under nonlinear deformation. Testing is done to check the viability of the new technique that is developed against the WikiArt digital art dataset and the DAVIS 2023 benchmark, by using multiple iterations of testing on both datasets. The tests confirm that the new method did indeed produce superior results when evaluating brushstroke temporal consistency, naturalness of motion following, and accurate artistic representation over all current mainstream algorithms. The results validate the potential for a new technical pathway for the dynamic display of high quality digital art works.

II. A METHOD FOR ASSISTED ANIMATION COMPOSITING BASED ON JOINT STYLE AND MOTION CONSTRAINTS

Figure 1 illustrates the complete process of the proposed technique. There are four main phases to this process: (1) feature extraction: using Canny Edge Detection and Bézier curve vectorization methods [20-21] for obtaining geometric contours as well as multi-scale style features and Gram matrix calculations from the VGG-19 encoder [22-23]; (2) target setting: calculating average style features by interpolating between skeleton contours; (3) constraint synthesis: using a hybrid loss optimization to drive the UNET generator in generating initial intermediate frames; and (4) visual optimization: obtaining the final synthesis re-

sult through adaptive anti-aliasing and local color correction methods. The specific implementation details of each stage are shown in Figure 1.

A. JOINT EXTRACTION OF BRUSHSTROKE GEOMETRIC FEATURES AND ATTRIBUTE FIELD

1) Adaptive Edge Detection and Bezier Vectorization

Given the original digital art images I_0 and I_1, they are first converted to grayscale to reduce the interference of color information on edge detection, resulting in the luminance channel image I_{gray}. Then, the adaptive threshold Canny edge detection operator is used to extract brushstroke contours. The computational process of this operator consists of four consecutive steps: Gaussian filtering for noise reduction, gradient magnitude and direction calculation, non-maximum suppression, and double-threshold edge connection. Figure 2 illustrates the complete extraction process from the original digital art image to the vectorized brushstroke contours.

Let the two-dimensional Gaussian smoothing kernel be:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

σ represents the standard deviation parameter, the smoothed image is $I_{smooth} = I_{gray} * G$. At pixel position (x, y), the gradient components along the horizontal and vertical directions are calculated by the Sobel operator:

$$\begin{aligned} G_x &= \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I_{smooth}, \\ G_y &= \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I_{smooth} \end{aligned} \tag{2}$$

Therefore, the gradient magnitude $M(x, y)$ and gradient direction $\theta(x, y)$ are obtained:

$$M(x,y) = \sqrt{G_x^2 + G_y^2}, \quad \theta(x,y) = \arctan\left(\frac{G_y}{G_x}\right) \tag{3}$$

In the non-maximum suppression stage, for each pixel, the magnitude of its neighboring area is compared along its gradient direction, and only local maxima are retained, thus refining the blurred edges to a single pixel width. Finally, adaptive high and low thresholds T_{high} and T_{low}, which are related to the local statistical characteristics of the image, are used instead of fixed thresholds. T_{high} is determined by the cumulative distribution function of the image gradient magnitude histogram, $T_{high} = \mu_M + k \cdot \sigma_M$, where k is a scaling factor. Pixels with magnitudes higher than T_{high} are marked as strong edge points, pixels between T_{low} and T_{high} that are connected to strong edge points are retained, and the rest are set to zero.

After obtaining the binary edge map E_{binary}, contour tracking is performed on its connected components to obtain an ordered edge point set $P = \{p_1, p_2, \ldots, p_n\}$. To transform discrete pixel-level edges into continuous curves with analytical form, a cubic Bézier curve is introduced to fit each contour segment. As shown in Figure 2(d), a cubic Bézier curve is defined by four control points P_0, P_1, P_2, P_3, and its parametric equation is as follows:

$$\begin{aligned} B(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 \\ + 3(1-t)t^2 P_2 + t^3 P_3, \quad t \in [0,1] \end{aligned} \tag{4}$$

t is the curve parameter. For a given edge point segment, the least squares fitting method is used to solve for the optimal control point position, minimizing the sum of squared Euclidean distances between the fitted curve and the original point set. After fitting, each stroke contour is described by a set of parameterized curve segments $\{B_i(t)\}_{i=1}^m$, constituting a complete vectorized stroke contour set C.

2) Stroke Direction Field and Thickness Distribution Estimation

As a structural unit with macroscopic direction and local width, the deformation law of a stroke needs to exceed pixel-level displacement. This paper constructs the stroke spatial attribute field by jointly using structural tensor and distance transformation. Given a grayscale image I_g, the gradient structural tensor is calculated within the sliding window $\Omega_r(\mathbf{x})$:

$$\mathbf{J}(\mathbf{x}) = \sum_{\mathbf{y} \in \Omega_r(\mathbf{x})} w(\mathbf{x}-\mathbf{y}) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{5}$$

Eigenvalue decomposition is performed on $\mathbf{J}(\mathbf{x})$, and the eigenvector $\mathbf{v}_{max}(\mathbf{x})$ corresponding to the largest eigenvalue is taken as the dominant local pen-cutting direction to construct a normalized direction field $\mathbf{D}(\mathbf{x}) = [d_x, d_y]^T$. Simultaneously, an Euclidean Distance Transform (EDT) is performed on the binary pen-stroke mask to obtain the thickness field $T(\mathbf{x}) = \mathrm{EDT}(\mathbf{x})$. The direction field and thickness field together characterize the topological morphology of the pen stroke, providing a deformation guidance benchmark for the nonlinear deformation of the moving region.

3) Pen-stroke Transparency and Particle Density Estimation

The random distribution and transparency variations of pen-stroke particles are the root cause of high-frequency flickering. In the context of computer graphics, the density-transparency attribute field consists of two complementary components: the particle density field which characterizes the number of brushstroke particles per unit area, and the transparency field which corresponds to the alpha channel representing the opacity of each brushstroke layer. This paper constructs a density-transparency joint field at the local texture response level. The input image is converted to the Lab color space, and the local normalized variance and Shannon entropy are calculated on the L channel:

$$\sigma^2(\mathbf{x}) = \frac{1}{|\Omega|} \sum_{\mathbf{y} \in \Omega} (L(\mathbf{y}) - \mu)^2 \tag{6}$$

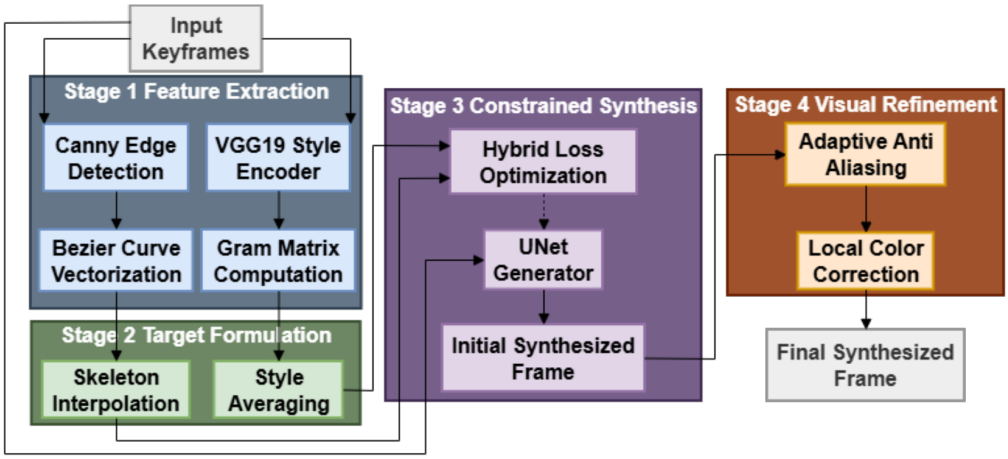

FIGURE 1. Overall framework of the method presented in this paper

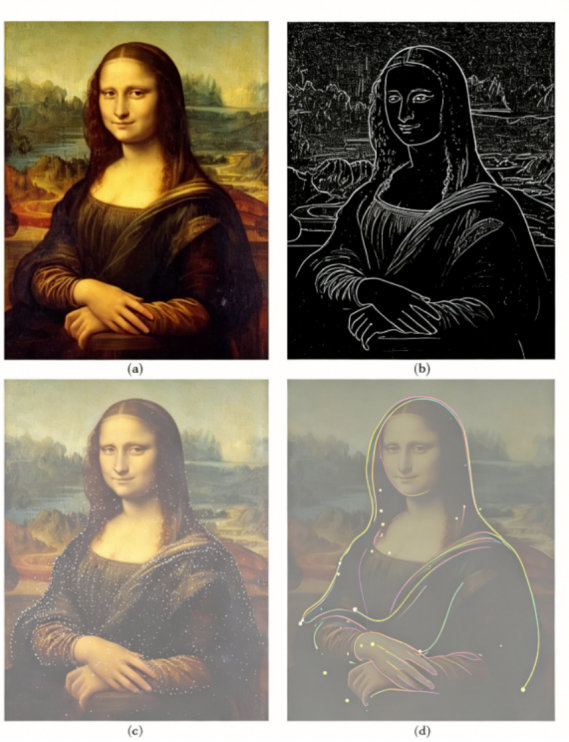

FIGURE 2. Flowchart of vectorized extraction of geometric features of brushstrokes. (a) Original digital art image. (b) Adaptive Canny edge detection results. (c) Edge feature point extraction. (d) Vectorized fitting results of Bézier curves

$$H(\mathbf{x}) = -\sum_{k} p_k \log p_k \tag{7}$$

p_k is the probability distribution of the gray-level histogram within the window. The brushstroke transparency field $\alpha(\mathbf{x})$ and the particle density field $\rho(\mathbf{x})$ are generated through nonlinear mapping:

$$\alpha(\mathbf{x}) = \frac{1}{1+\exp[-\kappa_1(\sigma(\mathbf{x})-\tau_1)]}, \quad \rho(\mathbf{x}) = \frac{H(\mathbf{x})}{H_{max}} \cdot \alpha(\mathbf{x}) \tag{8}$$

This field quantifies the microscopic statistical characteristics of brushstrokes within a single frame, serving as the physical basis for monitoring inter-frame temporal consistency.

B. MULTI-SCALE SEMANTIC STYLE FEATURE ENCODING

A constructed multi-scale use of a semantic style encoder that is based on a previously trained VGG-19 model will extract high dimensional feature representations that express

the essence of the artistic style in an original digital art image[24]. Each semantic style encoder will use the same input image and run through multiple forward propagations occurring in parallel at various scales (resolution levels) of the input image and generating hierarchy style descriptions of the original image from localised brushstroke particles to global colour composition.

Given an input image I, its multi-scale pyramid representation is first constructed. Let $s \in \mathcal{S}$ be a downsampling factor, and obtain the scale-transformed image sequence $\{I_s\}_{s\in\mathcal{S}}$ through bilinear interpolation, where $s = 1$ corresponds to the original resolution. Images at each scale are input into the VGG-19 network, and feature map responses are truncated at a specified set of feature extraction layers $\mathcal{L} = \{\text{conv1\_1}, \text{conv2\_1}, \text{conv3\_1}, \text{conv4\_1}, \text{conv5\_1}\}$. For the feature map extracted at the l-th layer at scale s, denoted as $F_s^l \in \mathbb{R}^{H_l \times W_l \times C_l}$, where H_l, W_l, and C_l are the height, width, and number of channels of the feature map at that layer, respectively. Local stroke direction, grain distribution and edge perturbation sensitivity of the shallow features like conv1_1 and conv2_1 encode texture attributes at the micro level; whereas the larger receptive field of the deep features (conv4_1, conv5_1) capturing global compositional structure, colour layout and division within the semantic region are their macro-level counterparts.

The Gram matrix is determined by calculating the two-dimensional spatial correlations of the feature map through calculating the normalized form of the inner product between channels in each feature map layer. The components of the Gram matrix are given by:

$$G_s^l(i,j) = \frac{1}{H_l W_l C_l} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} F_s^l(h,w,i) \cdot F_s^l(h,w,j) \quad (9)$$

The channels are labelled i and j respectively. The Gram matrix retains a form of correlation for activation modes, however the Gram matrix does not integrate any location information of the spatial locations of the activations and therefore can be viewed a valid and unique representation of style texture. The Gram matrix is calculated for every layer of feature activity at every different level of scale; afterwards, all of the Gram matrices will be expanded row-wise and concatenated creating the entire vector of style feature of the original input image.

$$\Phi(I) = \bigoplus_{s\in\mathcal{S}} \bigoplus_{l\in\mathcal{L}} \text{vec}(G_s^l) \quad (10)$$

$\text{vec}(\cdot)$ denotes a vectorization of matrices. This construction method enables the texture statistics encoded in $\Phi(I)$ to simultaneously be represented in different receptive field scales and at many resolutions of style. Therefore, $\Phi(I)$ provides hierarchical and differentiable supervisory signals to enforce style consistency in the generation of subsequent intermediate frames. The style constraint is applied in a spatially weighted manner, where the weight of the Gram matrix constraint at each spatial location is inversely proportional to the local optical flow amplitude.

C. MOTION ESTIMATION AND BRUSH FLOW FIELD CALCULATION

To establish pixel-level correspondences between adjacent frames, a lightweight optical flow network $\mathcal{F}_{opt}(\cdot)$ is used to estimate the forward optical flow field $\mathbf{V}_{t\to t+1}(\mathbf{x}) = [u(\mathbf{x}), v(\mathbf{x})]^T$ from time t to $t+1$. To suppress boundary blurring caused by non-rigid motion, a motion boundary confidence mask $M_{edge}(\mathbf{x}) = \exp(-\|\nabla\mathbf{V}(\mathbf{x})\|_2/\sigma_v)$ is introduced, dynamically adjusting the penalty intensity in subsequent constraints. An occlusion confidence mask $M_{occ}(\mathbf{x})$ is constructed based on the forward-backward optical flow consistency check. Pixels with forward-backward flow error exceeding 2 pixels are classified as occluded regions.

If one is to directly translate the brushstroke using optical flow, it would create large-scale structural damage (tearing). Using the orientation field D and the thickness field T parameterized in section 2.1.2, this paper develops a local affine deformation guidance matrix that is used to create an appropriate direction for each pixel in the image:

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} \cos\theta(\mathbf{x}) & -\sin\theta(\mathbf{x}) \\ \sin\theta(\mathbf{x}) & \cos\theta(\mathbf{x}) \end{bmatrix} \cdot \text{diag}\left(1 + \frac{\partial u}{\partial x}, 1 + \frac{\partial v}{\partial y}\right) \quad (11)$$

The angle, $\theta(\mathbf{x})$, between the optical flow divergence and the angle of the two orientation fields, is calculated to establish a reference frame for remapping. The resultant matrix will non-rigidly stretch or bend the stroke direction, and thickness of the pen along the motion path, to create the expected pen stroke flow field $[\mathbf{D}^*, T^*]$ in the target frame. For regions exhibiting complex non-rigid deformations such as fluid-like watercolor bleeds, a thin-plate spline transformation is applied in addition to the local affine transformation. The thin-plate spline transformation is computed using corresponding control points extracted from the vectorized brushstroke contours, enabling accurate modeling of smooth non-rigid deformations that cannot be represented by affine transforms.

D. GENERATION OF INTERMEDIATE FRAMES WITH MULTIPLE CONSTRAINTS

1) Geometric Contour and Style Statistical Constraints

Given a first frame image I_0 and a last frame image I_1, the goal is to synthesize a transition frame I_t located at the intermediate time $t \in (0,1)$. To this end, a dual-domain joint optimization generator G is constructed. Its input is the concatenation of the two original images along the channel dimension, and its output is the predicted intermediate frame $\hat{I}_t$. During generator training, in addition to the conventional reconstruction loss, two specific constraint terms are introduced to ensure the coherence of the geometric structure and the consistency of the artistic style, respectively.

First, the geometric constraint term L_{geo} is constructed. Using the vectorized brushstroke contour set C_0 of the first frame and the vectorized brushstroke contour set C_1 of the last frame obtained in Section 2.1.1, linear interpolation is performed on the control points of each corresponding

brushstroke to obtain the skeleton contour C_{mid} at the intermediate time. Specifically, for the control point pair P_0 and P_1, the intermediate position P_{mid} is calculated as follows:

$$P_{mid} = (1-t)P_0 + tP_1 \tag{12}$$

In this step, it calculates the desired edge map, designated as E_{target}, for the intermediate frame by replacing the initial control points with those found in the Bézier curve representations (parametric equations) defined above. The actual edge responses provide measured quantity values (E_{pred}) using the generated frame $\hat{I}_t$ via the Canny Operator, thus providing us with a geometric constraint term that it uses to measure the error through a mean square error calculation.

$$L_{geo} = \frac{1}{N}\sum_{x,y} \|E_{pred}(x,y) - E_{target}(x,y)\|^2 \tag{13}$$

(x, y) represents the image spatial domain coordinates, and N is the total number of pixels. This constraint forces the generated frame to maintain structural information consistent with the interpolated skeleton at the stroke edge position, thereby effectively suppressing breaks and blurring at motion boundaries.

As shown in Figure 3(a), without explicit geometric guidance, the generated result is prone to structural breaks and artifacts in the object contours and areas with dense textures (shown by white cracks in the figure). However, after introducing geometric constraints, as shown in Figure 3(c), the skeleton structure of the generated frame remains intact, and the edge transitions are smooth and coherent. Next, a style constraint term L_{style} is constructed. Using the multi-scale semantic style encoder described in Section 2.2, the Gram matrices G_0 and G_1 corresponding to the style feature vectors of the original first and last frames are extracted respectively. The target style representation is computed as a time-weighted and semantic-region-weighted average of the Gram matrices from the two keyframes

$$G_{target} = (1-t)W_0G_0 + tW_1G_1 \tag{14}$$

where t is the temporal interpolation factor and W_0, W_0 are semantic region-specific weight matrices derived from the segmentation masks of the input keyframes.

Simultaneously, the corresponding style feature vector of the generated frame $\hat{I}_t$ is extracted, and its Gram matrix G_{gen} is calculated. The semantic region weight matrices are computed based on the similarity of local feature descriptors between corresponding regions in the two keyframes, ensuring that regions with distinct artistic styles in each keyframe retain their unique characteristics rather than being averaged into a generic texture. The style constraint term is defined as the weighted Euclidean distance between the two in the Gram matrix space:

$$L_{style} = \sum_{l\in\mathcal{L}} \lambda_l \|G^l_{gen} - G^l_{target}\|^2_F \tag{15}$$

In the formula, $\mathcal{L}$ represents the set of multi-scale sampling factors, l is the selected feature layer index, λ_l is the normalized weight corresponding to each layer, $\|\cdot\|_F$ represents the Frobenius norm, and G^l_{gen} and G^l_{target} are the Gram matrices of the generated frame and the target style at layer l, respectively. This loss function forces the generated frame to maintain consistent texture statistics with the original work at different receptive field scales, thereby preventing the loss of brushstroke graininess and color shift during frame interpolation. The Gram matrix calculation is modified to incorporate spatial weighting based on the motion heatmap. For each feature map F^l_s, the spatially weighted Gram matrix is computed as:

$$G^l_s(i,j) = \frac{1}{H_lW_lC_l}\sum_{h=1}^{H_l}\sum_{w=1}^{W_l} w_m(h,w) \times F^l_s(h,w,i)\cdot F^l_s(h,w,j) \tag{16}$$

where $w_m(h,w) = 1 - \|V(h,w)\|/\max(\|V\|)$ is the motion-dependent weight. This modification reduces the influence of global texture statistics in regions with significant motion, prioritizing temporal coherence over strict style consistency in these areas.

As shown in the magnified details of Figures 3(b) and 3(d), under the influence of style constraints, the microscopic brushstroke direction and paint stacking texture of the painting surface are accurately reconstructed without texture degradation caused by excessive smoothing. In addition to the above two constraints, the generator training also uses pixel-level L_1 reconstruction loss to supervise low-frequency structures, i.e., $L_{rec} = \|\hat{I}_t - I_{gt}\|_1$, where I_{gt} is the real intermediate frame in the dataset. Combining all losses, the overall optimization objective is expressed as:

$$L_{total} = \lambda_{rec}L_{rec} + \lambda_{geo}L_{geo} + \lambda_{style}L_{style} \tag{17}$$

The three hyperparameters, λ_{rec}, λ_{geo}, and λ_{style} are all non-negative, and represent the contribution rate of each of the loss terms to the overall loss during the generator training process. The generator parameters, theta, are updated iteratively via the Adam optimizer until the total loss converges. Ultimately, the generator is able to generate a faithful reproduction of the stylized texture of the input digital artwork while preserving the geometric continuity of the contours of the brushstrokes.

2) Temporal Constraints on Brushstroke Attributes

To suppress high-frequency jitter between frames of brushstroke particles, temporal smoothing of the density and transparency fields is forced in the optical flow alignment region. Let $\mathbf{V}_{warp}$ be the result of the density/transparency field of the previous frame after optical flow distortion, and define the temporal coherence loss as follows:

$$L_{temp} = \frac{1}{N}\sum_x M_{static}(\mathbf{x})(\|\rho_t(\mathbf{x}) - \rho_{t+1}(\mathbf{x})\|_2 + \|\alpha_t(\mathbf{x}) - \alpha_{t+1}(\mathbf{x})\|_2) \tag{18}$$

FIGURE 3. Comparison of intermediate frame generation results under combined geometric and style constraints. (a) Structural breaks and artifacts due to lack of geometric constraints. (b) Preservation of details in local brushstroke textures under style constraints. (c) Coherent intermediate frames generated from the complete model. (d) Detail reconstruction results under dual geometric and style constraints

$M_{static}(\mathbf{x}) = 1 - \|\mathbf{V}(\mathbf{x})\| / \max(\|\mathbf{V}\|)$ is the static region mask. This constraint imposes a rigid temporal penalty in low-motion regions, fundamentally cutting off the texture flickering link caused by random sampling. The static region mask $M_{static}(\mathbf{x})$ is further weighted by a style preservation factor ($w_{style}(\mathbf{x}) = 1 - H(\mathbf{x})/H_{max}$), where $H(\mathbf{x})$ is the local Shannon entropy of the luminance channel.

The modified temporal coherence loss is expressed as:

$$L_{temp} = \frac{1}{N}\sum_{x} M_{static}(\mathbf{x}) w_{style}(\mathbf{x}) \times (\|\rho_t(\mathbf{x}) - \rho_{t+1}(\mathbf{x})\|_2 + \|\alpha_t(\mathbf{x}) - \alpha_{t+1}(\mathbf{x})\|_2) \tag{19}$$

3) Motion-Guided Stroke Remapping Constraint

In complex motion regions, strokes need to follow the nonlinear flow of the shape rather than simple translation. This paper introduces a direction-thickness joint remapping loss and combines it with a motion heatmap to implement a mask attenuation strategy:

$$L_{remap} = \frac{1}{N}\sum_{x} (1 - M_{high\_motion}(\mathbf{x})) \times (\|\mathbf{D}_{\hat{I}}(\mathbf{x}) - \mathbf{D}^*(\mathbf{x})\|_2 + \lambda_T \|T_{\hat{I}}(\mathbf{x}) - T^*(\mathbf{x})\|_2) \tag{20}$$

M_{high_motion} is generated based on an optical flow amplitude threshold. The motion heatmap mask value is computed using a continuous sigmoidal attenuation function

$$M_{high\_motion}(x) = \frac{1}{1 + \exp(-k(\|V(x)\| - \tau))} \tag{21}$$

where $\tau = 5$ pixels/frame is the motion threshold and k = 2 is the steepness parameter controlling the transition between weak and strong motion regions. In strong motion regions, the mask value approaches 0, and the loss automatically decays, allowing the generator to trigger natural brush redrawing based on style priors. For occluded regions identified by the forward-backward flow consistency check, the remapping constraint is completely removed and the generator synthesizes missing brushstroke textures using multi-scale style priors extracted from non-occluded regions of the same semantic category. The motion heatmap mask is further refined by multiplying with the occlusion confidence mask, ensuring that texture tearing caused by occlusion is addressed through style-consistent inpainting rather than forced warping of existing brushstrokes. In weak motion regions, it forces alignment with the expected flow field to avoid texture misalignment and duplicate mapping.

4) Joint Optimization and Generation

Considering the above constraints, a multi-objective loss function is constructed to drive the U-Net generator G_θ:

$$L_{total} = \lambda_{rec}L_{rec} + \lambda_{geo}L_{geo} + \lambda_{style}L_{style} + \lambda_{temp}L_{temp} + \lambda_{remap}L_{remap} \quad (22)$$

The Adam optimizer is used to iteratively update the parameters θ. After training convergence, the generator can explicitly maintain temporal stability of brush stroke attributes and achieve intelligent motion region tracking while preserving geometric coherence and style fidelity.

E. POST-PROCESSING OPTIMIZATION AND COLOR CORRECTION OF SYNTHESIZED FRAMES

While both the geometric structure and style texture of I_{gen} generated in the intermediate frame system initially meet or exceed the requirements for each of the prior systems, there could still be pixel-level jagged edges in areas with high-contrast edges, as well as some variance in local colour distribution compared to the reference frame caused by the global normalisation process that occurs during I_{gen} creation. To eliminate these visual defects, this section introduces a fast anti-aliasing processing based on edge direction adaptation and a color correction module based on local histogram matching to achieve the final refinement of visual quality [25-26].

To begin, the generated frame goes through anti-aliased optimized processing. The "Sobel" operator is applied to get the gradient magnitude $M(x, y)$ and gradient direction $\theta(x, y)$ used in determining sampling neighborhoods at the xy coordinate. If a given xy has a gradient magnitude greater than a threshold value, then a one-dimensional sampling neighborhood is created in the y-axis direction of the xy location. The length of this neighborhood is directly related to the local contrast (e.g., longer neighborhoods have higher contrast). A Gaussian weighted average of pixels in the adjacent neighborhood on both sides of the edge is used to combine the color for the pixel located at the edge.

$$\bar{C}(x, y) = \frac{\sum_{(u,v)\in\mathcal{N}(x,y)} w(u, v)I_{gen}(u, v)}{\sum_{(u,v)\in\mathcal{N}(x,y)} w(u, v)} \quad (23)$$

The weight function $w(u, v)$ is determined by both spatial distance and color similarity.

$$w(u, v) = \exp\left(-\frac{\|(u, v) - (x, y)\|^2}{2\sigma_s^2}\right) \exp\left(-\frac{\|I_{gen}(u, v) - I_{gen}(x, y)\|^2}{2\sigma_c^2}\right) \quad (24)$$

In the formula, σ_s controls the range of spatial smoothing, and σ_c controls the intensity of color edge preservation. This mechanism only applies to areas with significant gradients, while maintaining the original pixel values in flat areas. Therefore, it eliminates jagged edges without introducing overall blur, effectively preserving the sharpness of the brush stroke geometry extracted in Section 2.1.

Next, color correction is performed on the anti-aliased image I_{smooth}. Since the tonal distribution of the generated frame may have an overall offset from the original input frame I_0 or I_1, a local histogram matching strategy is used for channel-by-channel correction. Both I_{smooth} and the reference image I_{ref} are converted to the YCbCr color space. Histogram matching is performed only on the luminance channel Y and the two chrominance channels Cb and Cr to avoid color distortion introduced by cross-channel coupling. For each color channel $k \in \{Y, Cb, Cr\}$, the cumulative distribution functions F^k_{gen} and F^k_{ref} of the image to be corrected and the reference image are first calculated, where the independent variable is the normalized pixel value. Then, a mapping function $\mathcal{M}^k$ is established such that for any pixel value v, the following condition is met:

$$\mathcal{M}^k(v) = F^{k,-1}_{ref}(F^k_{gen}(v)) \quad (25)$$

The mapping function is constructed using piecewise linear interpolation. To suppress color spots and block artifacts caused by overcorrection in local areas, the image is divided into image blocks with overlapping ratios. The mapping function is calculated independently for each block, and spatial smoothing is performed using the weights at the block center positions. Finally, the color-corrected composite frame I_{final} is obtained.

III. EXPERIMENTAL SETUP AND EVALUATION BENCHMARKS

A. EXPERIMENTAL DATASET AND PREPROCESSING

Two datasets are used in this study: one published dataset that captures the variety of styles of digital artworks from the WikiArt Digital Art Archive, and one dataset that captures the variety of motion complexity in natural video sequences. The WikiArt Digital Art Archive supplied a collection of digital artwork images that represented five common styles: Impressionism, Expressionism, Post-Impressionism, Watercolor, and Impasto [27-28]. The selection process used for the digital art images complied with two selection criteria: images had to have a minimum resolution of 1024 × 768 pixels and they had to have a clear representation of brushstroke textures. Following manual checks for excessively abstract compositions or missing brushstroke detail, it is able to establish that there are 5000 individual image pairs remaining for further use in subsequent analyses. Each image pair is comprised of two sequential frames of a single painting at a single-frame time interval. The image pair's temporal relationship to the other image in the image pair is the same as the temporal relationship of the two sequential frames of the same painting when they are recreated as a still artwork being closely reviewed by a camera at a slow pan or zoom rate.

The DAVIS 2023 video object segmentation benchmark dataset [29-30] is also used to extract a portion of video frames from continuous live-action video and create high-level motion annotations at the pixel level to assess how well the method predicts optical flow and how accurate the

geometric constraint are in natural motion. From this data, thirty video sequences that contained clear translations and rotations of a boundary between two objects are selected, and a consecutive ten frames for each video sequence are used for the test sets.

During preprocessing, images are resized to uniform resolution of 512 by 512 pixels, using bicubic interpolation to ensure that edge sharpness and texture detail of brushstrokes are maintained. For the WikiArt dataset, two frames are extracted from a continuous sequence as the generator input: The first frame I_0 serves as the input to the generator (the last frame) and serves as the label for training. The mid-point in-frame $I_{0.5}$ is used to compute reconstruction loss for supervised learning and is used as the label. For the DAVIS 2023 dataset, a set of three frames (I_t, I_{t+2}, I_{t+1}) are extracted with a stride of two. Here, frame I_{t+1} serves as the label for the mid-point frame. All image pairs for both datasets are randomly divided into four equal portions, ensuring that any individual artwork or video sequence consists of only one image pair in either the training or the testing dataset. The pixel values of the preprocessed image pairs are represented in tensor format with pixel value normalized to the range between 0 and 1.

B. COMPARISON METHODS AND EVALUATION METRICS

For a thorough assessment of the performance of the algorithm proposed in this paper, three representative animation compositing algorithms have been chosen as comparison 'baseline' algorithms. The first algorithm is RIFE (Real-time Intermediate Flow Estimation), which estimates optical flow using a neural network in order to synthesize real-time intermediate frames, providing excellent levels of smooth motion; however, it is not explicitly designed to create artistic-style content. The second algorithm is SoftSplat, which employs forward warping and utilizes a soft compositing approach to deal with occlusions caused by optical flows in the forward-facing directions, resulting in an output with relatively sharp motion boundaries. The third is EbSynth, which applies style transfer via keyframes as its primary methodology, resulting in style texture continuity throughout an entire video sequence; however, it is sensitive to non-rigid motions. The selected algorithms are used for the evaluation of the proposed algorithm based on criteria of motion accuracy, boundary retention and quality of stylization transfers.

The evaluation metrics cover four aspects: motion smoothness, artistic style fidelity, brushstroke temporal stability, and computational efficiency.

Motion smoothness is quantified by two metrics: Average Endpoint Error (AEE) and Optical Flow Error (OFE). AEE is defined as the mean Euclidean distance between the predicted and actual optical flow vectors. Let the predicted optical flow field be $\mathbf{f}_{pred} = (u_{pred}, v_{pred})$, and the actual optical flow field be $\mathbf{f}_{gt} = (u_{gt}, v_{gt})$, then the formula for AEE (Average Endpoint Error) is:

$$\text{AEE} = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} \sqrt{(u_{pred} - u_{gt})^2 + (v_{pred} - v_{gt})^2} \tag{26}$$

Ω represents the effective pixel region of the image. Optical flow error is further calculated as a statistical measure of the pixel-by-pixel angular difference between the predicted and actual optical flow, defined as:

$$\text{OF Error} = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} \arccos\left(\frac{\mathbf{f}_{pred} \cdot \mathbf{f}_{gt}}{\|\mathbf{f}_{pred}\|\|\mathbf{f}_{gt}\| + \epsilon}\right) \tag{27}$$

In the formula, ϵ is a small constant to avoid division by zero. AEE focuses on the accuracy of displacement amplitude, while OF Error (Optical Flow Error) focuses on the fidelity of motion direction. Together, they are used to measure the smoothness of motion in the generated intermediate frames.

Artistic style fidelity is evaluated using the Fréchet Inception Distance and structural similarity index. FID measures style consistency by calculating the difference in distribution between the generated image set and the original image set in the Inception network feature space. Let the features of the real image follow a distribution $\mathcal{N}(\mu_r, \Sigma_r)$, and the features of the generated image follow a distribution $\mathcal{N}(\mu_g, \Sigma_g)$, then FID is defined as:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}\right) \tag{28}$$

μ and Σ are the mean and covariance matrices of the eigenvectors, respectively. A lower FID value indicates that the statistical properties of the generated image are closer to the original artwork. SSIM (Structural Similarity Index) measures the local similarity between two images in terms of brightness, contrast, and structure. Its calculation formula is:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{29}$$

μ_x and μ_y are local means; σ_x^2 and σ_y^2 are local variances; σ_{xy} is the covariance; and C_1 and C_2 are stability constants. The closer the SSIM value is to 1, the more structurally similar the generated frame is to the original frame.

Computational efficiency is measured by the average inference time per frame, in milliseconds. This metric provides an indication of the average processing time of one method against another, when using the same hardware environment to process a single pair of images to yield an intermediate frame. It reveals the general practical usability of a given method and, thus, its potential for generating a result in "real-time".

Temporal stability of strokes: this is used to objectively quantify the coherence of individual strokes across-frame(s), as well as the amount of high-frequency flicker present in the stroke particulate. Temporal stability is represented here using three separate measures:

Temporal Consistency Measure (TCM): this measure represents the mean values of the structural similarity (SSIM) among strokes within the stroke mask region following interface alignment using optical flow methods;

Perceptual Flicker Distance (as quantified using LPIPS-T): quantifies the cumulative amount of time between corresponding points in three sequential images as assessed using VGG-16 perceptual feature sets with high discriminative capabilities for texture changes that are perceptible to humans;

Temporal Structural Similarity (SSIM-temporal): this measure assesses the last known local structural preservation of each pixel within the stroke dominant mask region between approximately three sequentially captured images.

C. EXPERIMENTAL ENVIRONMENT AND PARAMETER CONFIGURATION

The configuration of both the hardware and software systems that are utilized in the experiment can be found in Table 1. The hardware environment is based on an Intel Xeon Gold 6226R processor along with an NVIDIA RTX 3090 graphics card (24GB of VRAM) and 128GB of RAM. The software environment is based on the Ubuntu 20.04 operating system, using PyTorch 1.12 as the deep learning framework, with CUDA version 11.6, and cuDNN (CUDA Deep Neural Network library) enabled to accelerate convolution operations.

The generator network uses a U-Net structure as its basic framework. Both the encoder and decoder parts contain six symmetrical downsampling and upsampling stages, and spatial detail information is transferred between corresponding scales through skip connections. The multi-scale style encoder is built based on a pre-trained VGG-19 network, selecting conv1_1, conv2_1, conv3_1, conv4_1, and conv5_1 as feature extraction layers. To balance the contribution of different receptive fields, the style loss weights of each layer are uniformly set as follows:

$$\omega_l = \frac{1}{5}, \quad l \in \mathcal{L} \tag{30}$$

$\mathcal{L}$ represents the feature layer set.

During the training phase, the Adam optimizer is used to update the generator parameters. The initial learning rate is set to 1×10^{-4}, and decays to 1×10^{-5} after the 100th epoch. The batch size is set to 8, and the total number of training epochs is 200. The weights of the hyperparameters in the loss function are determined by grid search, and the final configuration is as follows: reconstruction loss weight $\lambda_{rec} = 1.0$, geometric constraint weight $\lambda_{geo} = 0.5$, and style constraint weight $\lambda_{style} = 0.1$. In the post-processing optimization phase, the spatial smoothing parameter of the anti-aliasing algorithm is set to $\sigma_d = 1.2$, and the color edge preservation parameter is set to $\sigma_r = 0.05$. The color correction module divides the image into 64×64 pixel image blocks, and the overlap ratio between blocks is set to 0.25. The above parameter configurations are fine-tuned on the WikiArt training set using a 5-fold cross-validation strategy, and the final values minimize the overall loss on the validation set.

IV. EXPERIMENTAL RESULTS ANALYSIS AND DISCUSSION

A. ANALYSIS OF MOTION SMOOTHNESS AND OPTICAL FLOW ERROR

On 30 video sequences in the DAVIS 2023 test set, the average endpoint error and optical flow error of the proposed method are calculated compared with those of two representative baseline algorithms, RIFE and SoftSplat. During evaluation, the real intermediate frames in the original continuous frames are used as a reference, and the differences in optical flow fields between the frames generated by each method and the real frames are included in the statistics.

The quantification results are shown in Table 2. The proposed method achieved an average error of 1.47 pixels in the AEE index, a 22.2% reduction compared to RIFE's 1.89 pixels and a 14.5% reduction compared to SoftSplat's 1.72 pixels. Regarding optical flow error, the proposed method achieved an average angular deviation of 8.34°, compared to 10.61° for RIFE and 9.87° for SoftSplat, representing relative reductions of 21.4% and 15.5%, respectively. All of these numerical differences are statistically significant in the paired t-test at a significance level of $\alpha = 0.05$.

RIFE and SoftSplat rely entirely on convolutional networks to implicitly learn motion cues from pixel data when predicting optical flow. When encountering non-realistic textures such as oil paintings or watercolors, the random distribution of brushstroke particles and color gradients are easily misinterpreted as motion edges, causing the optical flow field to diffuse and break at the object's contour, thus increasing the global average error. In contrast, the proposed method extracts a pure geometric skeleton independent of the underlying texture through Bézier curve vectorization, and uses the expected edge map E_{target} generated by interpolation of this skeleton as the supervision signal for the geometric constraint term L_{geo}. During training, the generator is forced to learn to align the displacement of the edge region with the interpolated skeleton, ensuring that the optical flow prediction remains compact and continuous at the motion boundary. The integration of forward-backward optical flow consistency checking and occlusion-aware inpainting significantly reduces texture tearing artifacts in scenes with fast motion and extreme occlusion, as the generator relies on learned style priors rather than unreliable optical flow estimates to synthesize content in occluded regions. Directional deviation is a major contributor to optical flow erroneous measurements and thus, a 21.4% decrease shows an improvement in generator accuracy when predicting the direction of brushstroke contour resulting in less shift due to texture interference in the motion direction.

B. ARTISTIC STYLE FEATURE FIDELITY EVALUATION

It quantitatively examines the style fidelity of the intermediate frames generated from the WikiArt digital art dataset's

TABLE 1. Experimental Environment and Key Hyperparameter Configuration

Category	Parameter	Value / Specification
Hardware	CPU(Central Processing Unit)	Intel Xeon Gold 6226R
Hardware	GPU(Graphics Processing Unit)	NVIDIA RTX 3090 (24 GB VRAM)
Hardware	RAM(Random Access Memory)	128 GB
Software	OS(Operating System)	Ubuntu 20.04
Software	Framework	PyTorch 1.12
Software	CUDA / cuDNN	11.6 / Enabled
Network Architecture	Generator Backbone	U-Net (6 encoder/decoder stages, skip connections)
Network Architecture	Style Encoder	Pre-trained VGG-19
Network Architecture	Feature Layers	conv1_1, conv2_1, conv3_1, conv4_1, conv5_1
Network Architecture	Style Loss Weights (ω_l)	1/5 (uniform across layers)
Training Settings	Optimizer	Adam
Training Settings	Initial Learning Rate	1×10^{-4}
Training Settings	Learning Rate Decay	1×10^{-5} (after epoch 100)
Training Settings	Batch Size	8
Training Settings	Total Epochs	200
Loss Function Weights	Reconstruction (λ_{rec})	1
Loss Function Weights	Geometric Constraint (λ_{geo})	0.5
Loss Function Weights	Style Constraint (λ_{style})	0.1
Loss Function Weights	Temporal Constraint (λ_{temp})	0.3
Loss Function Weights	Remap Constraint (λ_{remap})	0.2
Post-processing	Anti-aliasing: σ_d/σ_r	1.2 / 0.05
Post-processing	Color Correction: Patch Size / Overlap	64×64 / 0.25
Validation	Strategy	5-fold cross-validation on WikiArt training set

TABLE 2. Comparison of Motion Smoothness Measurements

Method	AEE(pixel)	Optical Flow Error(°)
RIFE	1.89	10.61
SoftSplat	1.72	9.87
Ours	1.47	8.34

TABLE 3. Comparison of Quantitative Measurements of Artistic Style Fidelity

Method	FID	SSIM
RIFE	28.3	0.82
SoftSplat	26.7	0.85
EbSynth	22.4	0.89
Ours	15.6	0.94

test subset. It compares these generated intermediate frames to those created by three baseline algorithms (RIFE, SoftSplat, and EbSynth) by employing two evaluation metrics: Fréchet Inception Distance (FID) and Structural Similarity Index (SSIM). FID measures the agreement between the distribution of the generated intermediate frames and its original art source in the deep feature space (i.e., feature representations at the image's pixel level), while SSIM measures the similarity (at the local structural level) of a generated intermediate frame with a real intermediate frame.

All of the quantitative evaluation results can be seen in Table 3. The proposed method generated intermediate frames with a mean FID of 15.6, which is 44.9%, 41.6%, and 30.4% lower than those generated using RIFE (28.3), SoftSplat (26.7), and EbSynth (22.4), respectively. Similarly, the proposed method achieved a mean SSIM of 0.94, which is higher than RIFE (0.82), SoftSplat (0.85), and EbSynth (0.89), corresponding to relative improvements of 14.6%, 10.6%, and 5.6%, respectively. The above results indicate that the proposed method significantly outperforms the baseline in maintaining visual consistency of artistic style.

RIFE and SoftSplat prioritize pixel-level reconstruction accuracy during training, tending to linearly blend colors and textures between adjacent frames during frame interpolation. This results in reduced edge sharpness of brushstrokes and smoothing of the three-dimensional effect of pigment stacking into uniform color blocks. While EbSynth attempts to preserve stylistic features through a keyframe style transfer mechanism, its block-matching synthesis method is prone to texture tearing and repetitive patterns in non-rigid deformation areas. In contrast, the proposed method directly applies style constraints to the Gram matrix difference measure of the VGG-19 multi-scale feature layers. The Gram matrix encodes the correlation structure between feature channels, independent of pixel spatial location. Therefore, L_{style} forces generated frames to maintain consistent texture statistics with the original work across different receptive fields. Shallow Gram matrix constraints maintain the microscopic orientation and density distribution of brushstrokes, while deep Gram matrix constraints ensure stylistic consistency in global color layout and semantic regions. The semantic-region-weighted style averaging mechanism preserves intentional style variations between keyframes, preventing the ghosting effect and generic texture artifacts that would result from simple arithmetic averaging of global Gram matrices. The spatially weighted Gram matrix constraint resolves the conflict between global texture statistics and local motion requirements by dynamically adjusting the style constraint strength based on local motion amplitude, ensuring that temporal coherence is prioritized in moving regions while style fidelity is maintained in static regions.

TABLE 4. Quantitative Comparison of Pen Stroke Timing Stability

Method	TCM	LPIPS-T	SSIM-temporal
RIFE	0.71	0.087	0.78
SoftSplat	0.76	0.071	0.82
EbSynth	0.81	0.063	0.85
Ours	0.89	0.042	0.91

C. STABILITY ANALYSIS OF BRUSHSTROKE TIMING

To quantitatively evaluate the ability of the proposed method to suppress high-frequency flicker between brush strokes, three temporal consistency metrics are introduced on the WikiArt test set: Temporal Consistency Metric (TCM), defined as the mean structural similarity between the generated frame and the target frame within the brush stroke mask region after adjacent frames are aligned by optical flow warping; Perceptual Flicker Distance (LPIPS-temporal), which calculates the cumulative temporal difference between three consecutive frames based on VGG-16 perceptual features, and this metric has higher discriminative power for texture jumps that are sensitive to human vision; and Temporal Structural Similarity (SSIM-temporal), which calculates the local structural preservation of adjacent frames in the brush stroke-dominated region to verify the inter-frame coherence of the geometric skeleton.

The quantization results are shown in Table 4. The current method achieves significantly higher values for all metrics examined compared with existing solutions. Specifically: 1) TCM improves from RIFE's 0.71 to 0.89 by 25.4%, SoftSplat's 0.76 to 0.89 (17.1%), and EbSynth's 0.81 to 0.89 (9.9%); 2) LPIPS-temporal drops from the average of the three methods to the lowest value 0.042; and 3) SSIM-temporal achieves a value of 0.91 and verifies that the structure of the stroke skeleton is stable throughout dynamic transitions in time. Lowering of the stroke density field ρ and transparency α by strictly aligning them with respect to/in relation to one another after applying optical flow distortion has created a temporary rigid temporal penalty in the low-motion area thus breaking the connection between texture flickering caused by random sampling. The integration of thin-plate spline transformations and texture diffusion terms significantly improves the handling of complex non-rigid deformations, ensuring that fluid-like brushstroke behaviors such as watercolor bleeds are accurately reproduced in the animated sequence.

D. COMPARISON OF SINGLE-FRAME INFERENCE TIME AND PARAMETER QUANTITY FOR EACH ALGORITHM

Using an equivalent hardware setup, it measures the single-frame inference speed of the technique relative to RIFE (the most efficient deep learning method), SoftSplat, and EbSynth; the measurement is based on the separating out only the time required for pure inference (i.e. input image pair -> output intermediate frame), whereas data loading and post-processing times do not factor into the results. The WikiArt test dataset comprised 1000 image pairs; each pair is measured 10 times to account for potential variability caused by fluctuations in the performance of the hardware on which the tests are conducted.

The results of the quantification are displayed in Figure 4. The average single-frame inference times for the system (28.6 ms), RIFE (36.3 ms), SoftSplat (41.7 ms), and EbSynth (52.4 ms) are as follows. Accordingly, the proposed system demonstrated a 21.2% reduction compared to RIFE, while it claimed a 45.4% advantage relative to EbSynth. Additionally, in terms of model complexity, the proposed generator network contains 4.2M parameters; in contrast, RIFE has 9.8M parameters, while SoftSplat possesses 7.5M parameters. Since EbSynth's block-matching algorithm lacks any parameters eligible for training, it is omitted from this analysis.

The efficiency advantage stems from two design decisions. First, the generator network employs a lightweight U-Net structure, replacing some standard convolutional layers with depthwise separable convolutions, significantly reducing floating-point computation while maintaining receptive field coverage. Second, both the vectorized stroke extraction described in Section 2.1 and the post-processing module described in Section 2.5 are implemented using non-iterative parsing algorithms, running in parallel on the CPU and overlapping with the GPU inference pipeline without additional critical path time. In contrast, RIFE and SoftSplat rely on multiple optical flow iterations for refinement, while EbSynth requires full tile matching search for each output frame; all three have significantly higher computational loads than the method presented in this paper.

E. ABLATION EXPERIMENTS: PERFORMANCE CONTRIBUTION ANALYSIS OF EACH MODULE

To quantify the independent contribution of each key component in the proposed method to the final performance, all variants are independently trained and tested under a unified dataset partition, training protocol, and hyperparameter configuration to ensure the fairness and reproducibility of the comparison results. Based on the full model, ablation variants are constructed: w/o L_{temp}: removes the temporal constraint loss, retaining only geometric, style, and remapping constraints; w/o L_{remap}: removes the motion-guided remapping loss, retaining only geometric, style, and temporal constraints; w/o Both: removes both L_{temp} and L_{remap}, retaining only basic geometric and style constraints; Baseline: trained using only the pixel reconstruction loss L_{rec} as a lower bound reference for performance. The purpose of this paper is to provide a more accurate measurement of the actual benefit from using the post-processing optimization feature described in Section 2.5 because this similarity adds to its validity. A model that does not use post-processing has been created to use the same configuration and multi-constraint model during training but skips over the adaptive anti-aliasing and local colour correction phases after generating the final output images. Four primary metrics are used during the test phase to

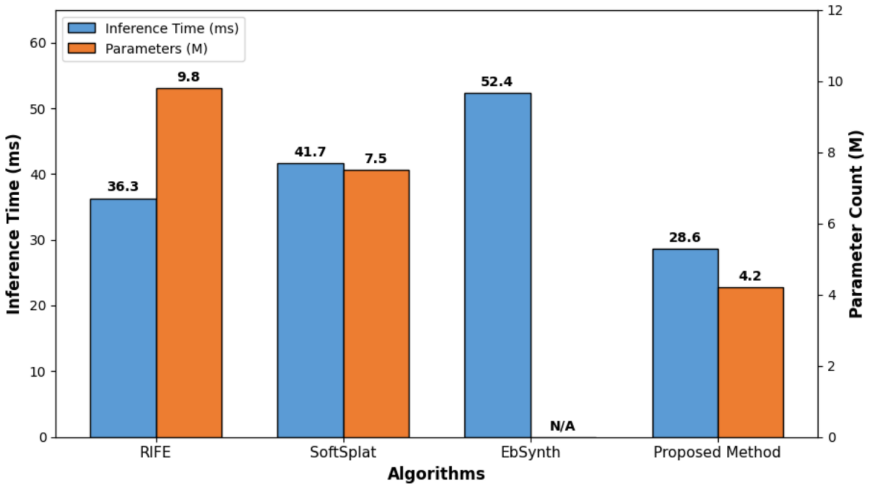

FIGURE 4. Bar chart comparing single-frame inference time and parameter count for each algorithm

quantitatively evaluate the images created with and without using a post-processing feature. FID measures how well an image preserves the style (fashion) from the input image. AEE measures how fluid an object's movement appears on the screen. TCM measures how consistent an object's strokes appear over time. LPIPS-T measures how much an object appears to be flickering. The measurement area of TCM and LPIPS-T will be strictly limited to the area of the mask used to generate the strokes as this is done in an attempt not to allow for any contribution of the background to be included in either metric.

Quantification is displayed in Table 5. The complete model produced optimum results on all four measurements and thus demonstrated that the multi-constraint collaborative design is a viable method of design. The Baseline variance using only the pixel reconstruction loss L_{rec}, as the bottom boundary for performance, is the least performing across all measurements, having a TCM = 0.61 and LPIPS-T = 0.112. This demonstrates that relying solely on pixel-level regression cannot resolve the issues of temporal jitter and style degradation in brushstrokes. The independent contributions of each constraint module analyzed below are as follows:

The independent contribution of the temporal constraint L_{temp}: After eliminating L_{temp}, the TCM dropped from 0.89 to 0.73. The LPIPS-T increased from 0.042 to 0.081, while FID and AEE remained relatively stable. The primary way that L_{temp} suppresses high-frequency flickering of the individual brushstroke particles is by forcing the density field ρ and the transparency field α to be aligned after optical flow distortion. Without this constraint, the generator degenerates to frame-by-frame independent optimization, and thus cannot effectively suppress the effects of random sampling noise; the result is a severe degradation in temporal consistency.

Impact of L_{remap}'s independent contribution: A removal of L_{remap} leads to the AEE dropping from 1.47 pixels to 2.05 pixels. The TCM and LPIPS-T remain fairly constant throughout the entire experiment. With the L_{remap} constraint in place, the D (orientation field) and T (thickness field) are used jointly to combine with a motion heatmap mask attenuation technique to allow for brushstroke flow along an intelligent nonlinear trajectory within non-linear motion regions. Without this constraint, the only means of guiding the generator with pixel-to-pixel optical flow at simple translation will ultimately create textures that tear and brushstrokes misalign at the motion boundaries, as such this represents a significant increase in motion error.

Through cooperative dual-module collaboration, the removal of both L_{temp} and L_{remap} yielded for a total set of four decimated metrics. The overall decline in these metrics illustrates that the two constraints operate on the orthogonal dimensions of "temporal texture stability" and "spatial motion following". The individual absence of either module will cause significant degradation in the specific dimension of system performance being represented. Only through mutual collaboration can both overscale static fidelity, and overscale dynamic smoothness.

Verification of the necessity of the basic modules: Further observation of the ablation results of the geometric constraint L_{geo} and the style constraint L_{style} reveals that when L_{geo} is missing, AEE increases from 1.47 pixels to 2.13 pixels, primarily ensuring the continuity of the skeleton structure; when L_{style} is missing, FID increases from 15.6 to 26.1, primarily maintaining the fidelity of texture statistics. Furthermore, the w/o Post-processing variant is almost on par with the full model in four core metrics (FID only changed from 15.6 to 16.8), indicating that the post-processing optimization module in Section 2.5 contributes relatively little to the evaluation metrics at the deep feature level. Its main value lies in eliminating visible jagged edges and color deviations, further enhancing the visual detail of

TABLE 5. Quantitative Comparison of Ablation Experiments

Model Variant	FID	AEE (pixel)	TCM	LPIPS-T
Full Model	15.6	1.47	0.89	0.042
w/o L_{temp}	15.9	1.48	0.73	0.081
w/o L_{remap}	16.1	2.05	0.88	0.043
w/o Both ($L_{temp} + L_{remap}$)	21.4	2.31	0.69	0.095
w/o Geometric Constraint (L_{geo})	24.7	2.13	0.87	0.044
w/o Style Constraint (L_{style})	26.1	1.52	0.88	0.043
Baseline (L_{rec})	32.4	2.67	0.61	0.112
w/o Post-processing	16.8	1.47	0.89	0.042

the image.

V. CONCLUSION

This paper addresses three core challenges in digital art image animation compositing: loss of temporal coherence of brushstroke particles, brushstroke misalignment in moving areas, and degradation of stylistic details. It proposes an animation compositing technique based on optical flow-guided brushstroke registration and dual-domain adaptive constraints. The construction of a unified technical framework for the modeling of brushstroke attributes, the encoding of style features and the guidance of motion deformation has resolved a major contradiction: How to maintain both the fidelity of static art texture and the natural flow of dynamic brushstrokes. The methodology models brushstrokes as separate macroscopic drawing units that work together to extract their geometrical shapes and their density-transparency attribute fields. High-frequency flickering is suppressed at its source through the use of rigid temporal constraints in the static portion of the brushstroke area. A motion heatmap mask attenuation technique has been developed that allows for each brushstroke to intelligently follow the artist as they draw and for the brushstrokes to be naturally redrawn under complex motion using local affine deformation. The use of multi-scale VGG-19 style encoding ensures that the artistic texture of every image frame produced using this technique is identical. Experimental results demonstrate that, in comparison to previous methods such as RIFE, the proposed method results in a 25.4% increase in temporal consistency, 51.7% reduction in perceived flickering distance, 1.47 pixels endpoint average error and 15.6 FID, along with a 21.2% decrease in single frame inference time compared to RIFE. Ablation experiments confirm that there is a beneficial synergy resulting from the combination of the various modules. This technique has many potential applications within a digital illustration animation context. Future studies will enhance the capability of this technique to produce extremely non-rigid motion and will expand its use in creating long sequence imagery.

AUTHOR CONTRIBUTIONS

Conceptualization – JianHui Yang and Yuan Shen; methodology – JianHui Yang and Yuan Shen; investigation – JianHui Yang and Yuan Shen; writing-original draft preparation – JianHui Yang and Yuan Shen. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] W. Fan and L. Fan, "A dynamic study of art image digitization based on metadata," J. Combin. Math. Combin. Comput., vol. 127, no. 1, pp. 8449-8466, 2025. DOI: https://doi.org/10.61091/jcmcc127b-462

[2] F. Colonnese, "The thread of the virtual movement from Wölfflin to Lynn," Dimensions-Journal of Architectural Knowledge, vol. 1, no. 2, pp. 79-96, 2021. [Online]. Available: https://iris.uniroma1.it/handle/11573/1655809

[3] Z. Wang and Z. Chu, "Research on intelligent keyframe in-betweening technology for character animation based on generative adversarial networks," Journal of Advanced Computing Systems, vol. 3, no. 5, pp. 78-89, 2023. DOI: https://doi.org/10.69987/JACS.2023.30507

[4] Z. Wang and Z. Chu, "GAN-based intelligent keyframe interpolation method for character animation: An automated in-betweening approach," Journal of Science, Innovation & Social Impact, vol. 1, no. 2, pp. 29-40, 2025. [Online]. Available: https://sagespress.com/index.php/JSISI/article/view/51

[5] S. Ding, "Fusion of motion smoothing algorithm and motion segmentation algorithm for human animation generation," PLOS ONE, vol. 20, no. 2, pp. 1-23, 2025. DOI: https://doi.org/10.1371/journal.pone.0318979

[6] P. Saha and C. Zhang, "From pixels to motion: A systematic analysis of translation-based video synthesis techniques," Information, vol. 16, no. 11, pp. 990-1019, 2025. DOI: https://doi.org/10.3390/info16110990

[7] L. Istead, J. Istead, A. Pocol, et al., "A simple, stroke-based method for gesture drawing," Virtual Reality & Intelligent Hardware, vol. 4, no. 5, pp. 381-392, 2022. DOI: https://doi.org/10.1016/j.vrih.2022.08.004

[8] X. Tian and S. Tirakoat, "The creative watercolor flow in the digital epoch: A novel approach to cross-disciplinary aesthetics integration," TEM Journal, vol. 13, no. 4, pp. 1-10, 2024. DOI: https://doi.org/10.18421/TEM134-19

[9] A. Pei, "Online oil painting teaching based on deep learning and image super-resolution reconstruction," Discover Computing, vol. 28, no. 1, pp. 334-360, 2025. DOI: https://doi.org/10.1007/s10791-025-09847-0

[10] V. Zabora, K. Kasianenko, S. Pashukova, et al., "Digital art in designing an artistic image," Amazonia Investiga, vol. 12, no. 64, pp. 300-305, 2023. [Online]. Available: https://eprints.zu.edu.ua/36188/7/A_23.pdf

[11] M. Yum, "Digital image color analysis method to extract fashion color semantics from artworks," Multimedia Tools and Applications, vol. 82, no. 11, pp. 17115-17133, 2023. DOI: https://doi.org/10.1007/s11042-022-14189-w

[12] Y. Yu, J. Qian, C. Wang, et al., "Animation line art colorization based on the optical flow method," Computer Animation and Virtual Worlds, vol. 35, no. 1, pp. e2229-e2238, 2024. DOI: https://doi.org/10.1002/cav.2229

[13] W. Zhang, Y. Wang, and Y. Liu, "Generating high-quality panorama by view synthesis based on optical flow estimation," Sensors, vol. 22, no. 2, pp. 470-485, 2022. DOI: https://doi.org/10.3390/s22020470

[14] J. Liu, "Animation art design online system based on mobile edge computing and user perception," Journal of Sensors, vol. 2021, no. 1, pp. 1-10, 2021. DOI: https://doi.org/10.1155/2021/9974170

[15] G. Liu and S.-B. Tsai, "Online optimization of animation art design user virtual perception in mobile edge computing environment," Scientific Programming, vol. 2022, no. 1, pp. 1-9, 2022. DOI: https://doi.org/10.1155/2022/1171905

[16] I. Santos, L. Castro, N. Rodriguez-Fernandez, et al., "Artificial neural networks and deep learning in the visual arts: A review," Neural Computing and Applications, vol. 33, no. 1, pp. 121-157, 2021. DOI: https://doi.org/10.1007/s00521-020-05565-4

[17] S. Zhang, Y. Qi, and J. Wu, "Applying deep learning for style transfer in digital art: Enhancing creative expression through neural networks," Scientific Reports, vol. 15, no. 1, pp. 11744-11759, 2025. DOI: https://doi.org/10.1038/s41598-025-95819-9

[18] X. Tian and T. Günther, "A survey of smooth vector graphics: Recent advances in representation, creation, rasterization, and image vectorization," IEEE Transactions on Visualization and Computer Graphics, vol. 30, no. 3, pp. 1652-1671, 2022. DOI: https://doi.org/10.1109/TVCG.2022.3220575

[19] R. Dragoș, "Traditional art in the digital age: Formats, pixels and vectors," Învățământ, Cercetare, Creație, vol. 11, no. 1, pp. 86-94, 2025. [Online]. Available: https://www.ceeol.com/search/article-detail?id=1355443

[20] Y. Li and D. Zhang, "Toward efficient edge detection: A novel optimization method based on integral image technology and canny edge detection," Processes, vol. 13, no. 2, pp. 293-311, 2025. DOI: https://doi.org/10.3390/pr13020293

[21] H. Song and Z. Wen, "Interactive design modeling of 3D styling combining Bezier curve and Harris corner point detection algorithm," PLOS ONE, vol. 20, no. 4, pp. 1-22, 2025. DOI: https://doi.org/10.1371/journal.pone.0319323

[22] Z. Zhao and S. Zhang, "Style transfer based on VGG network," International Journal of Advanced Network, Monitoring and Controls, vol. 7, no. 1, pp. 54-72, 2022. DOI: https://doi.org/10.2478/ijanmc-2022-0005

[23] Z. Zhang, X. Zhou, M. Qin, et al., "Chinese character style transfer based on multi-scale GAN," Signal, Image and Video Processing, vol. 16, no. 2, pp. 559-567, 2022. DOI: https://doi.org/10.1007/s11760-021-02000-6

[24] J. Yu, L. Jin, J. Chen, et al., "Deep semantic space guided multi-scale neural style transfer," Multimedia Tools and Applications, vol. 81, no. 3, pp. 3915-3938, 2022. DOI: https://doi.org/10.1007/s11042-021-11694-2

[25] Y. Zhang, X. Wang, G. Shi, et al., "Anti-aliasing and anti-color-artifact demosaicing for high-resolution CMOS image sensor," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 70, no. 12, pp. 4928-4937, 2023. DOI: https://doi.org/10.1109/TCSI.2023.3290157

[26] Q. Wen, Y. Liu, T. Luo, et al., "Research on crosstalk and color aliasing compensation of color image sensor based on artificial neural network," IEEE Photonics Journal, vol. 14, no. 3, pp. 1-9, 2022. DOI: https://doi.org/10.1109/JPHOT.2022.3176734

[27] C. Meinecke, C. Hall, and S. Jänicke, "Towards enhancing virtual museums by contextualizing art through interactive visualizations," ACM Journal on Computing and Cultural Heritage, vol. 15, no. 4, pp. 1-26, 2022. DOI: https://doi.org/10.1145/3527619

[28] B. Srinivasa Desikan, H. Shimao, and H. Miton, "Wikiartvectors: Style and color representations of artworks for cultural analysis via information theoretic measures," Entropy, vol. 24, no. 9, pp. 1175-1188, 2022. DOI: https://doi.org/10.3390/e24091175

[29] Z. Yang, Y. Wei, and Y. Yang, "Associating objects with transformers for video object segmentation," Advances in Neural Information Processing Systems, vol. 34, no. 1, pp. 2491-2502, 2021. DOI: https://doi.org/10.48550/arXiv.2106.02638

[30] J. Bertrand, G. Kordopatis Zilos, Y. Kalantidis, et al., "Test-time training for matching-based video object segmentation," Advances in Neural Information Processing Systems, vol. 36, no. 1, pp. 20918-20941, 2023. [Online]. Available: https://jbertrand89.github.io/test-time-training-vos/