ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Architecture Design of digital painting system integrating computer graphics

Xuan Wang[1], *Wei Lu*[2] *and Weiya Yang*[3,*]

[1] Department of Urban Design, College of Technology, Hubei Engineering University, Xiaogan 432000, Hubei, China
[2] School of Civil Engineering, Hubei Engineering University, Xiaogan 432000, Hubei, China
[3] School of Animation and Digital Arts, Communication University of Nanjing, Nanjing, Jiangsu 211172, China

SUMMARY: *Aiming at the problems of the separation of brush stroke rendering and aesthetic evaluation, the lack of local texture control and the lag of interactive response in digital painting system, a digital painting system architecture combining computer graphics and aesthetic optimization was designed. The system encodes stroke geometry, layer state, color distribution, composition structure and texture response as fusion features, and constructs stroke cover calculation, texture perturbation rendering, aesthetic score feedback and local incremental update mechanisms. The experimental results show that the FPS of our system reaches 58.7, the average delay is 24.6 ms, the SSIM reaches 0.902, and the LPIPS is reduced to 0.118. After Aesthetic optimization, the aesthetic scores of oil painting, watercolor, ink painting and sketch samples are increased to 8.61, 8.42, 8.36 and 8.29, respectively. Ablation experiments show that the aesthetic optimization module contributes the most to the score improvement, and the local update module can significantly reduce the interaction delay. The research can provide technical reference for real-time generation and picture quality optimization of digital painting system.*

KEYWORDS: *Computer Graphics; Digital Painting System; Aesthetic Optimization; Brushstroke Rendering Deep Learning*

1 Introduction

Digital painting systems have shifted from traditional bitmap editing, layer overlaying and fixed brush rendering to intelligent rendering environments supported by computer graphics rendering, deep feature extraction and parameter optimization algorithms. Stroke trajectories, layer blending, texture diffusion, color transitions, and local detail enhancement are no longer just interface results, but image generation units that can be encoded, computed, and updated with feedback. Brush-based neural rendering methods can integrate painting regions, stroke trajectories and stylization results into a unified generation process, which provides an iterative rendering mechanism for digital painting systems [1]. Image aesthetic evaluation models have also gradually shifted from artificial experience judgment to deep network scoring, so that the integrity of composition, color coordination, texture clarity and style consistency have a computable expression basis [2-5]. This kind of research provides technical support for the transformation of digital painting system from "manual drawing

*aluoya6161616@163.com
https://doi.org/10.65102/is2026970
Published: 30 April, 2026

tool" to "collaborative system of rendering generation and aesthetic feedback".

In the existing research on digital painting, computer graphics methods pay more attention to stroke morphology, texture mapping, illumination level and layer synthesis, while aesthetic optimization methods pay more attention to image quality scoring, style consistency judgment and visual feature selection. If the two techniques run independently, it is easy to have problems such as good visualization effect but uncontrollable aesthetic evaluation of rendering results, or high aesthetic score but lack of real brush structure in rendering details. In order to achieve stable generation of digital painting system, it should not only rely on a single style transfer model or post-processing enhancement algorithm, but also organize graphics rendering pipeline, aesthetic feature representation, scoring feedback, and parameter update into the same architecture, forming a continuous computing link from low-level rendering to high-level evaluation. Style transfer models such as PhotoGAN have proved that the digital image generation process can obtain stronger visual expression ability through content preservation and style transfer [6], but the integrated design for digital painting system architecture still needs further refinement.

Based on this, this paper focuses on the architecture of digital painting system that integrates computer graphics and aesthetic optimization. The system takes stroke geometric parameters, layer rendering features, color distribution, composition structure and texture aesthetic features as input, and constructs a fusion model of graphic rendering features and aesthetic evaluation features. At the algorithm level, brush stroke generation, texture rendering, aesthetic scoring, parameter optimization and local interactive update modules are designed. At the implementation level, the effectiveness of the system is verified by rendering efficiency, image quality, aesthetic score and interactive response performance. This paper attempts to form an engineering technology mainline of "graphics rendering-aesthetic representation-parameter optimization-system verification", and provides reproducible technical solutions for the architecture design and algorithm implementation of digital painting systems.

2 Literature analysis

Concerning the architecture design of digital painting system, the existing research mainly focuses on four directions: brush rendering, style transfer, aesthetic evaluation and interactive image processing. In terms of stroke rendering, the neural rendering method based on curve strokes integrates painting trajectory, region filling and texture generation into a unified network structure, which can simulate continuous brush morphology and local detail transition [7]. MambaPainter compreses the neural stroke generation process, so that brush-level rendering can be completed in a single step calculation, and provides a lightweight idea for real-time painting systems [8]. HRL-Painter introduces hierarchical reinforcement learning into painting planning, and optimizes stroke sequence, position and area coverage through action layering, indicating that digital painting generation can be controlled by sequential decision model [9]. Watercolor digitization research focuses on pigment diffusion, transparent color superposition and edge blur, which provides a graphical reference for texture layer rendering and material expression [10].

The research of style transfer and image generation provides content structure preservation, style texture conversion and local visual reconstruction methods for digital painting systems. The multimodal semantic enhancement algorithm uses text semantics and image features to constrain the transfer process, so that the generated results are consistent between content boundaries and decorative styles [11]. The aesthetic-aware adversarial

learning network empathizes aesthetic constraints into style transfer to reduce color drift and texture fragmentation [12]. The contrast attention and fine-grained feature fusion method strengthen the matching relationship between content structure and style texture, which can be used for local detail enhancement and style consistency control in digital painting [13].

The study of image aesthetic evaluation provides the basis for the aesthetic optimization module of this paper. Personalized aesthetic evaluation methods learn user preferences through graph neural networks and collaborative filtering, so that the scoring results are no longer limited to general aesthetic standards [14]. The multi-scale regional attribute weighting model extracts color, texture, composition and saliency features from local regions, which is suitable for serving the quality judgment of painting images [15]. CNN-Transformer meta-learning method fuses local texture features and global layout information to improve the adaptation ability of small sample personalized aesthetic scores [16]. Self-supervised aesthetic evaluation methods use mask modeling and contrastive learning to reduce the dependence on manual labeling and provide training strategies for automatic evaluation [17]. AesMamba uses the state space model for general image aesthetic evaluation, which has advantages in long-term visual dependency modeling [18].

In order to summarize the connection relationship between related research and the system architecture of this paper, Table 1 is sorted out from three dimensions: technical direction, representative methods and referable content.

Table 1: Alignment of Relevant Research Directions with the System Architecture of This Paper

Technical Direction	Representative Research	Inspiration for This System Architecture
Brushstroke Rendering and Neural Painting	Curve stroke neural painting, MambaPainter, HRL-Painter [7-9]	Support brushstroke geometry encoding, drawing order control, and local rendering design
Material Texture Expression	Research on watercolor flow and digital texture generation [10]	Support texture diffusion, transparent color overlay, and layer blending modeling
Style Transfer and Generation Optimization	Semantic-enhanced transfer, adversarial learning transfer, fine-grained feature fusion [11-13]	Support style consistency preservation, local detail enhancement, and generation quality optimization
Aesthetic Evaluation Model	Graph neural network, regional attribute weighting, CNN-Transformer, self-supervised learning, AesMamba [14-18]	Support scoring of color, composition, texture, and overall visual quality
Systematic Image Processing	Mobile image aesthetic processing, AI graphic design review, generated image quality evaluation [19-22]	Support system implementation, interactive processing, evaluation indicators, and experimental validation

The existing research has provided the algorithmic foundation for digital painting systems, but the overall is still relatively scattered. Brush stroke rendering emphasizes the underlying rendering behavior, style transfer emphasizes visual transformation, aesthetic evaluation emphasizes score prediction, and systematic research is more oriented to general image processing or design assistance [19-22]. These achievements have not yet formed a

continuous architecture of "graphics rendering output - aesthetic scoring feedback - painting parameter update - interactive result verification". In this paper, brush geometry, layer rendering, texture expression, color composition and aesthetic scoring are incorporated into the unified system design, so that the model construction, algorithm implementation and experimental verification maintain the same technical main line.

3 The construction of digital painting model combining graphics rendering and aesthetic representation

3.1 Stroke geometry parameters and layer rendering feature encoding

The underlying input of the digital painting system consists of user strokes, canvas coordinates, pressure changes, layer states and texture control parameters. If the rendering results are directly sent to the subsequent module as image pixels, the model can only obtain the final picture information, and it is difficult to trace the stroke shape, rendering strength and layer stacking process. In this paper, the single stroke is defined as a computable geometric object, and the layer rendering state is encoded synchronically, so that the system can preserve the continuous relationship of "stroke track-layer depth-transparent overlay-texture response" in the generation stage. Suppose that the sampling point of the ith stroke at time t is (xi,t,yi,t), and the stroke width, direction, pressure, velocity, transparency and curvature are wi,t,θi,t,pi,t,vi,t,αi,t,κi,t, respectively. Then the geometric parameter vector of the stroke can be expressed as follows.

$$s_{i,t}=\left[x_{i,t},y_{i,t},w_{i,t},\theta_{i,t},p_{i,t},v_{i,t},\alpha_{i,t},\kappa_{i,t}\right]^{T} \tag{1}$$

where si,t denotes the geometric state vector of the ith stroke at sampling time t; xi,t denotes the center position of the stroke; wi,t denotes the stroke width; Let θi,t denote the orientation Angle; pi,t denotes the pressure sensing intensity; vi,t is the moving speed; αi,t denotes transparency; Let κi,t denote the trajectory curvature. The vector transforms the user's drawing behavior from pixel results to structural parameters, and provides basic input for stroke reconstruction, texture rendering and local optimization.

In the continuous drawing process, the mouse, stylus or digital tablet collect discrete coordinate points. If directly connected, it is easy to appear polyline edge and local jitter. In this paper, the weighted curve interpolation method is used to smooth the stroke trajectory, so that the drawing path obtains stable geometric continuity on the basis of maintaining the user's original motion trend. The stroke trajectory sampling function is expressed as follows.

$$q_i(\tau)=\sum_{r=0}^{R} B_{r,R}(\tau)P_{i,r}, \quad 0\le\tau\le1 \tag{2}$$

where, qi(τ) represents the trajectory point of the ith stroke at the normalized position τ. Pi,r denote the control points; Br,R of τ denotes the Bernstein basis functions of order R. The computational process is able to transform discrete inputs into smooth paths, reducing brush edge aliasing and local breaks.

The stroke geometry parameters need to enter the model together with the layer rendering state. To avoid only listing concepts in the table, Table 2 organizes the key parameters involved in coding in the digital painting system into four categories: geometry, dynamic, layer and texture, so as to facilitate subsequent rendering module calls.

Table 2: Geometric Parameters of Brush Strokes and Layer Rendering Feature Configuration Table

Feature Category	Parameter Item	Function Description
Stroke Geometry	Position, Width, Direction, Curvature	Describe the spatial position, thickness, direction, and curvature of brushstrokes
Stroke Dynamics	Pressure, Speed, Sampling Interval	Describe drawing force, movement speed, and sampling continuity
Layer Rendering	Alpha, Blend Mode, Layer Depth	Control layer transparency, blending mode, and overlay order
Texture Control	Roughness, Diffusion, Edge Softness	Control texture granularity, diffusion range, and edge softening effect

In multi-layer digital painting, a single stroke does not directly cover the canvas, but transparently blends with the existing pixels in the specified layer. Suppose the original color of the LTH layer at pixel position (u,v) is Cl−1(u,v), the current stroke color is ci(u,v), the layer transparency is αl, and the stroke cover weight is mi(u,v), then the layer blending rendering result is as follows.

$$C_l(u,v)=(1-\alpha_l m_i(u,v))C_{l-1}(u,v)+\alpha_l m_i(u,v)c_i(u,v) \quad (3)$$

where, Cl(u,v) represents the pixel color after fusion at the LTH layer; mi(u,v) represents the coverage intensity of the current stroke at that pixel. This formula unifies layer transparency, stroke coverage and color contribution into the same rendering expression, so that the system can support local erasing, color overlapping, translucent covering and multi-layer detail enhancement.

There is a clear pre-post computational relationship between stroke parameters, trajectory interpolation, and layer blending. To illustrate how the coding links are organized in the system, Figure 1 shows the coding structure of stroke geometric parameters and layer rendering features.

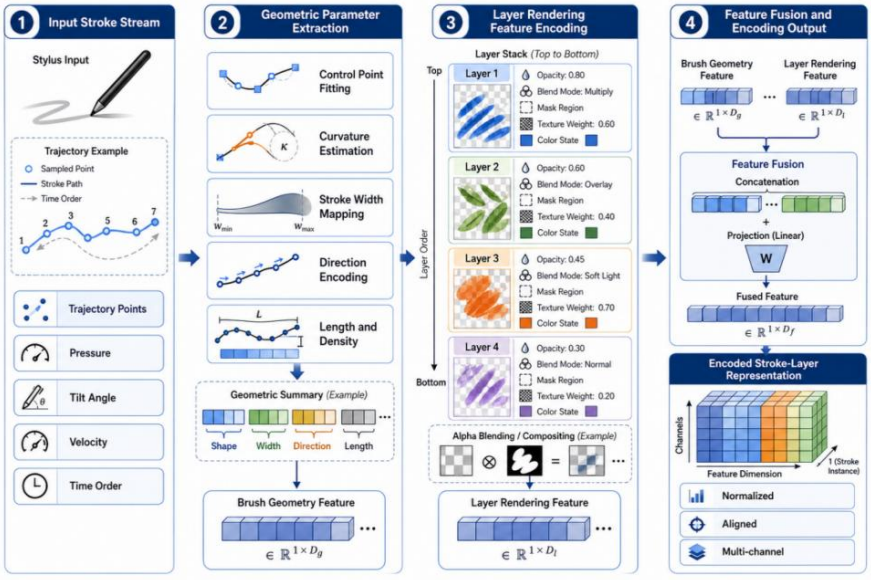

Figure 1: Coding structure diagram of stroke geometry parameters and layer rendering features

Through the above encoding, the digital painting system no longer treats the picture as a single pixel matrix, but converts the drawing actions, layer relationships and texture control into feature vectors that are trainable, renderable and feedback-able. The vector can be used as the basic data for subsequent color distribution, composition structure, texture aesthetic feature extraction and fusion modeling.

3.2 Color distribution, composition structure and texture aesthetic feature extraction

After encoding stroke geometry parameters and layer rendering features, the system needs to extract visual features from the overall image that can be used for aesthetic judgment. In this paper, aesthetic features are classified into three categories: color distribution, composition structure, and texture response. Color distribution is used to describe hue, saturation and brightness ratio. The composition structure is used to describe the main area, the visual center of gravity and the spatial balance. Texture response is used to describe edge variation, local detail density, and brushstroke texture continuity. The three types of features together form the input of subsequent aesthetic scoring and parameter optimization.

Color feature extraction is based on HSV color space. Compared with RGB space, HSV can more directly express the hue change, saturation and light and dark levels, which is suitable for the analysis of color coordination in digital painting images. Suppose that the screen contains N valid pixels, the hue, saturation and brightness of pixel n are hn,sn and vn, respectively, and the indicator function of the BTH color interval is Ib(.), the color distribution vector can be expressed as follows.

$$d_c=\left[\frac{1}{N}\sum_{n=1}^{N} I_1(h_n,s_n,v_n),\frac{1}{N}\sum_{n=1}^{N} I_2(h_n,s_n,v_n),\cdots,\frac{1}{N}\sum_{n=1}^{N} I_B(h_n,s_n,v_n)\right]^T \tag{4}$$

where, dc represents the color distribution vector of the screen; B denotes the number of color ranges; $I_b(h_n,s_n,v_n)$indicates whether pixel n falls into the BTH color range. This vector can transform the dominant color ratio, auxiliary color distribution and local high saturation area into fixed dimensional features, which can provide a basis for color matching for aesthetic scoring.

Compositional structure features are used to describe the relationship between the subject and the spatial layout. If the digital painting system only focuses on pixel-level quality, it is prone to problems such as subject shift, white space imbalance or visual center dispersion. In this paper, the weight of salient region is used to describe the structure distribution of the picture. Suppose that the picture is divided into M composition regions, the significance weight of the MTH region is am, the regional center coordinates are (xm,ym), and the geometric center of the picture is (x0,y0), then the visual center offset can be expressed as:

$$D_{comp}=\sqrt{\left(\sum_{m=1}^{M} a_m x_m - x_0\right)^2+\left(\sum_{m=1}^{M} a_m y_m - y_0\right)^2} \tag{5}$$

where, Dcomp represents the visual center offset; am represents the significance weight of the MTH region and satisfies $\sum_{m=1}^{M} a_m=1$; (xm,ym) denotes the region center coordinate. (x0,y0) is the center coordinate of the screen. The larger the value of this index is, the more obvious the deviation from the center of the screen main body or high saliency region is, which can be used to constrain the regional layout adjustment in the subsequent parameter optimization.

Texture aesthetic features mainly reflect the local details of the picture, brush texture and

edge level. Digital painting textures contain both edge gradients and local variations produced by brush roughness, pigment stacking, paper surface particles, and transparent color stacking. In this paper, the gradient response and local variance are jointly used to describe the texture intensity. Suppose that the gray-scaled painting image is I(u,v), the horizontal and vertical gradients of pixel position (u,v) are ∇xI(u,v) and ∇yI(u,v), respectively, and the mean value within the local window Ωu,v is μΩ, then the texture response value can be expressed as follows.

$$T(u,v)=\sqrt{(\nabla_x I(u,v))^2+\left(\nabla_y I(u,v)\right)^2}+\lambda_t \frac{1}{|\Omega_{u,v}|}\sum_{(r,k)\in\Omega_{u,v}}\left(I(r,k)-\mu_\Omega\right)^2 \tag{6}$$

where, T(u,v) represents the texture response value at pixel position (u,v); Let λt denote the local variance term weight; Let Ωu,v denote the local window; ｜Ωu,v｜ denote the number of pixels in the window. The former term extracts the edge gradient and stroke contour, and the latter term measures the local gray fluctuation and texture grain strength, which avoids the system misjudging only strong edges as high-quality textures.

Color, composition and texture features correspond to the global color scheme, spatial layout and local details of the picture, respectively. To illustrate how the three types of features are extracted in the system, Figure 2 shows the joint processing of color distribution, composition structure and texture response.

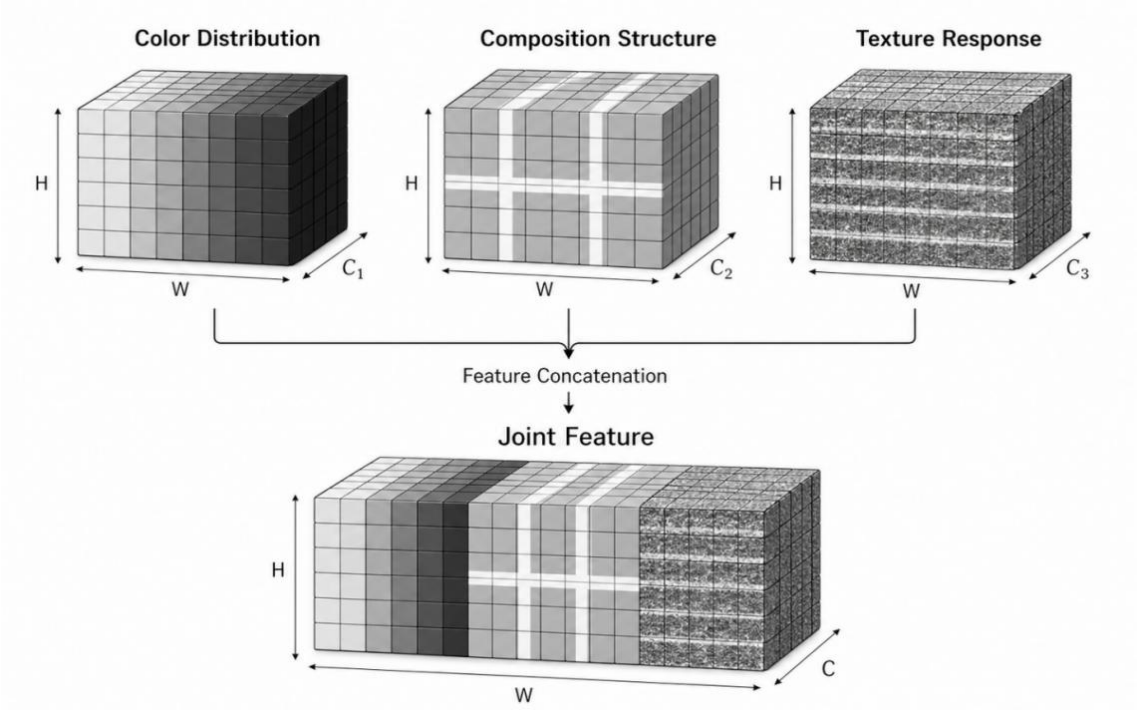

Figure 2: Schematic diagram of color distribution, composition structure and texture aesthetic feature extraction

The input painting image is transformed into HSV distribution vector. The composition analysis module divides the salient regions, calculates the visual center, region weight and spatial balance relationship. Texture extraction module generates texture intensity map based on gradient response and local variance. The three types of outputs are normalized to form a set of aesthetic features, which provide input for the fusion modeling of graphic rendering features and aesthetic evaluation features in the next section.

3.3 Fusion modeling of graphic rendering features and aesthetic evaluation features

Stroke geometry, layer rendering, color distribution, composition structure and texture response describe different information layers in a digital painting system, respectively. If all kinds of features enter the aesthetic scoring module independently, the model is prone to two types of deviations. One is that it only focuses on the quality of the image surface layer and ignores the brush stroke path, layer depth and transparent superposition relationship. The other is that only the underlying rendering parameters are retained, and it is difficult to judge the overall color scheme, composition balance and texture beauty of the picture. Based on the feature extraction results of the previous two sections, this paper constructs the fusion expression of graphic rendering features and aesthetic evaluation features, so that the system can use the rendering process information and the screen aesthetic information at the same time.

Let the graphic rendering feature formed by the ith stroke be gi, which consists of stroke geometry parameters, trajectory interpolation results, layer blending states and texture control parameters. The frame-level aesthetic evaluation feature is a, which consists of color distribution, composition shift, and texture response statistics. In order to eliminate the differences in dimension and numerical range between the two types of features, a unified representation is first obtained by linear mapping and nonlinear activation:

$$z_i=\sigma\left(W_g g_i+W_a a+b_z\right) \tag{7}$$

where, zi represents the fused initial feature corresponding to the ith stroke; Wg and Wa represent the mapping matrix of graphic rendering features and aesthetic evaluation features, respectively. bz represents the bias term; Let $\sigma(\cdot)$ denote the nonlinear activation function. This formula compresses the underlying rendering information and screen level aesthetic information into the same feature space, avoiding scale inconsistency when the subsequent models deal with the two types of features separately.

Different strokes and different layer regions do not contribute equally to the final picture quality. Brush strokes near the edges of the subject, areas of high saturation, areas of dense texture, and the visual center generally have a greater impact on aesthetic ratings. In this paper, attention weight is introduced to dynamically screen the fusion features, so that the model automatically adjusts the importance of each stroke area according to the screen structure. The attention weights are calculated as follows:

$$\beta_i=\frac{\exp\left(w_\beta^T z_i\right)}{\sum_{j=1}^{K}\exp\left(w_\beta^T z_j\right)} \tag{8}$$

where, βi represents the attention weight of the ith fused feature unit. wβ represents the trainable weight vector; K represents the number of stroke or region features involved in fusion. This weight can improve the proportion of key strokes, main regions and high texture regions in the fusion expression, and reduce the interference of low contribution background regions on the scoring results.

After obtaining the attention weights, the system performs weighted aggregation on all fused feature units to form the overall representation of the digital painting screen. The representation not only preserves the brushstroke generation process, but also incorporates the aesthetic evaluation information of the screen, which can be used as a unified input for subsequent aesthetic scoring, parameter optimization and interactive feedback. The fusion

output is expressed as follows.

$$f_p = \sum_{i=1}^{K} \beta_i z_i \tag{9}$$

where, fp represents the fused digital painting feature representation; Let βi denote the attention weight of the ith feature unit; zi denotes the corresponding fused initial features. Through this aggregation method, the model can unify the local stroke, layer rendering, color distribution, composition structure and texture response into a frame-level representation, reducing the evaluation bias caused by a single feature source.

In order to present the convergence relationship of the two types of features in the model, Figure 3 shows the fusion model of graphic rendering features and aesthetic evaluation features. The graphics rendering branch inputs brushstroke geometry, layer blending, texture control, and trajectory encoding results, while the aesthetics evaluation branch inputs color distribution, composition shift, and texture response statistics. After dimension mapping, the two branches enter the attention fusion module. The model generates the screen-level fusion representation according to the regional contribution weight, and passes the representation to the subsequent aesthetic scoring and parameter optimization module.

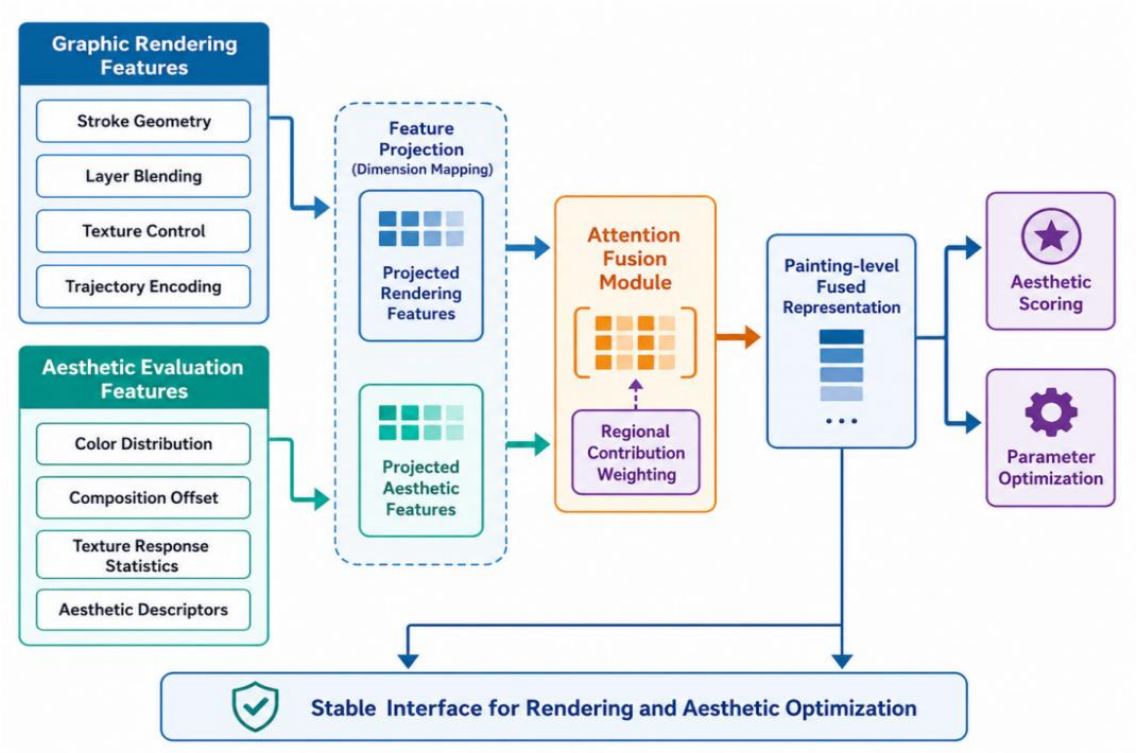

Figure 3: Fusion model diagram of graphic rendering features and aesthetic evaluation features

The fusion model connects "rendering process parameters" and "image aesthetic characteristics" into the same calculation link, so that the digital painting system can not only evaluate the final pixel results, but also interpret the image quality changes by using stroke trajectories, layer relationships and texture control information. The subsequent system algorithms take fp as the core input when performing aesthetic scoring and painting parameter updating, so as to ensure a stable data interface between rendering generation and aesthetic optimization.

4 System algorithm and architecture design for digital painting generation optimization

4.1 Overall architecture and data processing flow of digital painting system

The generation and optimization process of digital painting system consists of user input, graphics rendering, aesthetic evaluation, parameter optimization and interactive feedback. After receiving the stylus trajectory, reference image, layer state and local editing instructions, the system completes stroke sampling, layer coding and texture parameter reading, and then sends the rendering results to the aesthetic evaluation module to output the picture quality score and local adjustment weight. In order to ensure that data from different sources can enter the unified computing link, this paper represents the system input as a multi-modal painting state tensor:

$$X_t=[S_t,L_t,R_t,A_{t-1},U_t] \tag{10}$$

where, Xt represents the system input state at time t; St represents stroke trajectories and geometric parameters; Lt represents the layer transparency, blending mode, and depth relationship. Rt represents the reference image or target style feature; At−1 represents the aesthetic evaluation feedback of the previous round; Ut represents user local editing instructions. This representation unifies user actions, rendering states and aesthetic feedback into the same input space, and provides data basis for subsequent stroke generation, texture rendering and parameter update.

The data transfer relationship between the modules of the system is shown in Table 3. This table corresponds the input, output and function division to facilitate the definition of processing boundaries in the implementation of the algorithm.

Table 3: Digital painting system module and input and output configuration table

Module Name	Input	Output	Primary Function
Stroke Encoding Module	Stroke trajectory, pressure data	Stroke feature vector	Extract brushstroke geometry and dynamic features
Layer Rendering Module	Stroke feature, layer state	Rendered layer image	Complete layer blending and pixel rendering
Texture Generation Module	Brush parameter, texture control	Texture response map	Generate brush texture and edge diffusion effects
Aesthetic Evaluation Module	Rendered image, fused feature	Aesthetic score	Output image quality score and regional weights
Parameter Optimization Module	Score, rendering parameter	Updated parameter set	Adjust color, transparency, and texture intensity
Interactive Feedback Module	Local editing command	Updated canvas region	Complete local refreshing and real-time feedback

In order to present the overall organizational relationship of the digital painting system in the generation and optimization process, Figure 4 shows the overall system architecture and data processing flow.

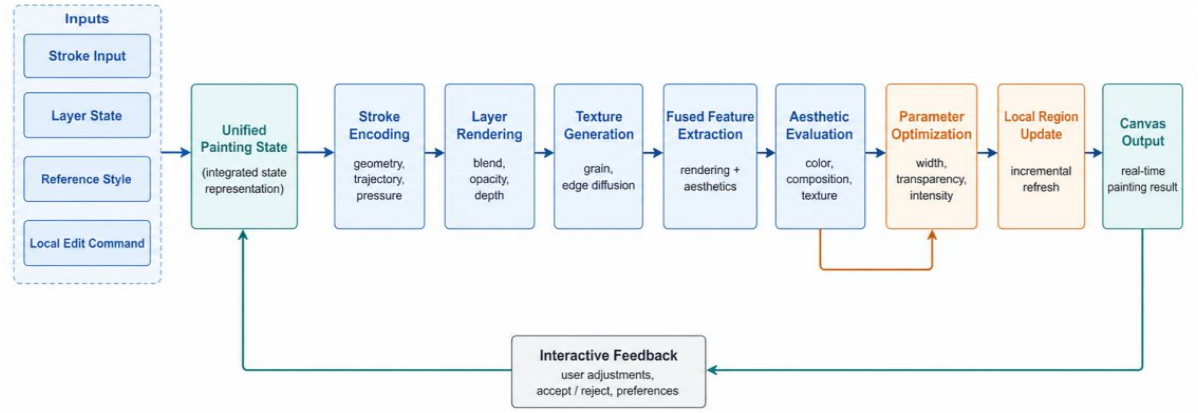

Figure 4: Digital painting system overall architecture and data processing flow chart

When the system runs, user strokes, layer states, reference images and local editing instructions are uniformly encoded as input states, and the initial drawing results are formed by strokes encoding, layer rendering and texture generation modules. The aesthetic evaluation module scores the quality of the rendering results, and transmits the score value and the region weight to the parameter optimization module, which is used to adjust the stroke width, transparency, texture intensity and color weight. The interactive feedback module only refreates the local area affected by editing, which reduces the computational overhead caused by the repeated rendering of the whole screen, and makes the system maintain the real-time response ability in the state of continuous rendering.

4.2 Brush stroke generation and texture rendering algorithm based on computer graphics

The stroke generation module undertakes the transformation task from the parameter vector to the pixel coverage area. Brush strokes in digital painting are not simply line segments, but are determined by the center trajectory, brush radius, pressure intensity, direction change and edge attenuation. Assuming that the current stroke center trajectory is $q_i(\tau)$, the canvas pixel position is $p_{u,v}=(u,v)$, the stroke radius is $r_i(\tau)$, and the edge softening factor is ρ_i, the pixel coverage can be expressed as follows.

$$M_i(u,v)=\max_{\tau\in[0,1]}\left[\exp\left(-\frac{\left\|p_{u,v}-q_i(\tau)\right\|_2^2}{2\rho_i^2 r_i^2(\tau)}\right)\right] \tag{11}$$

where, $M_i(u,v)$ represents the coverage of the ith stroke at pixel (u,v); $p_{u,v}$ represents the pixel coordinates of the canvas; $q_i(\tau)$ represents the stroke center trajectory; $r_i(\tau)$ is the brush radius at the trajectory position τ. Let ρ_i denote the edge softening factor. This calculation method can make the stroke boundary gradually decay from the center to the outside, and avoid the jagged feeling caused by the hard edge coverage.

The texture rendering module adds paper particles, brush roughness and edge diffusion on the basis of coverage, so that the generated results are closer to the local texture of real paintings. Assuming that the basic stroke color is c_i, the paper texture noise is $N_p(u,v)$, the brush texture disturbance is $N_b(u,v)$, the diffusion weight is δ_i, and the roughness weight is η_i, the texture rendering result can be expressed as follows.

$$P_i(u,v)=M_i(u,v)c_i+\eta_i M_i(u,v)N_b(u,v)+\delta_i(1-M_i(u,v))N_p(u,v) \tag{12}$$

where, Pi(u,v) represents the texture rendering result of the ith stroke at pixel (u,v); ci denotes the base stroke color; Nb(u,v) represents the brush texture perturbation; Np(u,v) represents the paper texture noise; Let ηi denote the roughness weight; Let δi denote the edge diffusion weight. The formula unifies brush coverage, local roughness and paper grain into the rendering process, so that the system can generate different painting effects such as hard edge strokes, soft edge strokes, dry brush textures and diffuse smudges.

To show the influence of stroke parameters change on texture rendering results, Figure 5 shows the comparison of texture rendering effects under different stroke parameters.

Figure 5: Comparison of texture rendering effects under different stroke parameters

Different combinations of parameters will change stroke edges, texture grains and color diffusion patterns. The high pressure sense parameter makes the coverage area more full, the roughness enhancement will improve the grain sense of the brush, and the large edge diffusion weight will form a soft blur effect. The algorithm combines graphics coverage calculation with texture perturbation modeling, and can provide rendering results with clear parameter sources for subsequent aesthetic scoring modules.

4.3 Screen Aesthetic Scoring and Parameter Optimization Algorithm Based on Deep learning

The screen aesthetics scoring module is used to convert the rendering results into numerical feedback that can be used for parameter updates. The fusion feature fp obtained above already contains the information of stroke geometry, layer state, color distribution, composition structure and texture response, but these features still cannot directly guide the system to

adjust painting parameters. In this paper, the multi-layer perceptron scoring network is connected to fuse the features, and the overall quality of the picture, color coordination, composition balance and texture details are uniformly mapped into aesthetic scores. Set the scoring network parameters as Θe, the hidden layer weights as W1,W2, and the bias terms as b1,b2, then the screen aesthetic score prediction can be expressed as follows.

$$\hat{y}_a=\sigma\left(W_2\phi\left(W_1 f_p+b_1\right)+b_2\right) \tag{13}$$

where, $\hat{y}_a$ represents the predicted aesthetic score; fp represents the fused digital painting features; Let $\phi(\cdot)$ denote the nonlinear activation function of the hidden layer; $\sigma(\cdot)$ is used to compress the ratings to a fixed interval; Θe denotes all trainable parameters in the scoring network. The scoring result can not only be used to evaluate the current picture quality, but also be used as a feedback signal for the subsequent optimization module.

In order to avoid the model only pursuing a single score value and ignoring the structural stability of the painting picture, we incorporate aesthetic scoring error, composition shift, color coordination and texture smoothness constraints into the loss function. Suppose the target score given by manual annotation or baseline model is ya, the composition offset is Dcomp, the color distribution deviation is Ec, and the texture overresponse penalty is Et, then the comprehensive aesthetic optimization loss is as follows.

$$L_a=\left(\hat{y}_a-y_a\right)^2+\lambda_1 D_{comp}+\lambda_2 E_c+\lambda_3 E_t \tag{14}$$

where, La represents the comprehensive aesthetic optimization loss; λ1,λ2,λ3 denote the composition, color, and texture constraint weights, respectively. The first term controls the score prediction error, the second term suppresses the excessive deviation of the main area, the third term restricts the color distribution to deviate from the target style, and the fourth term reduces the noise sensation caused by excessive local texture. The loss function transforms the image quality assessment into an optimization objective that can be backpropagated, so that the system can obtain a clear parameter adjustment direction from the aesthetic score.

The parameter optimization module iteratively updates the painting parameters according to the loss function. In this paper, the parameters to be optimized are denoted as θp, which include stroke width, transparency, texture intensity, edge diffusion, color weight, and layer blending coefficient. In order to ensure the stability of the update process, the step size control term and regular constraint are introduced, and the parameter update formula is as follows:

$$\theta_p^{(k+1)}=\theta_p^{(k)}-\gamma\frac{\partial L_a}{\partial\theta_p}-\gamma\lambda_r\left(\theta_p^{(k)}-\theta_p^{(0)}\right) \tag{15}$$

where, $\theta_p^{(k)}$ represents the drawing parameter of the KTH iteration; Let γ denote the learning rate; Let λr denote the regularization weight; Let $\theta_p^{(0)}$ denote the initial drawing parameters. The first gradient update is used to improve the aesthetic score of the screen, and the second regularization constraint is used to prevent the parameters from deviating too far from the user's original intent. After several iterations, the system can automatically correct color harmony, texture details and composition balance while maintaining stroke structure and layer relationship. In this way, aesthetic optimization no longer stops at the post-processing filter level, but directly acts on interpretable painting generation parameters.

4.4 Local Update and Real-time Feedback for Interactive rendering

In the continuous interaction process of digital painting system, users often perform color complement, erase, texture enhancement and brush correction on local areas. If the whole image is re-rendered for each operation, it will cause computational redundancy and affect the real-time rendering experience. In this paper, the canvas is divided into several local blocks, and the region to be updated is determined according to the coverage of the current stroke and the influence relationship of the layers. Let the user's editing region at time t be Et, the stroke influence radius be rt, and the neighborhood set with layer dependence to the region be N(Et). Then the local update region can be expressed as follows.

$$U_t=\{(u,v) \mid \mathrm{dist}((u,v),E_t)\le r_t\} \cup N(E_t) \tag{16}$$

where, Ut denotes the local update region at time t; (u,v) represents the pixel coordinates of the canvas; dist(·) is the distance from the pixel to the edit area. N(Et) denotes the neighborhood that has a layer overlay, transparent blending, or texture diffusion relationship with the current editing region. This region selection method can avoid the repeated calculation of irrelevant canvas blocks, and only the local pixels affected by user operations are retained to participate in the update.

The real-time feedback mechanism generates interactive outputs based on local rendering results, cached canvas, and aesthetic score changes. Suppose the canvas of the previous frame is Yt−1, the local re-render result is Gt(Ut), the region mask is Mt, and the aesthetic feedback correction term is ΔAt. Then the output of the updated canvas is as follows.

$$Y_t=(1-M_t)\odot Y_{t-1}+M_t\odot G_t(U_t)+\lambda_f \Delta A_t \tag{17}$$

where Yt represents the canvas output of the current frame; Mt represents the local update mask; ⊙ for element-wise multiplication; Let λf denote the aesthetic feedback correction weight. This mechanism retains the cached results of the unmodified area, only re-renders the affected area, and lightly modifies the local color, transparency and texture intensity according to the change of aesthetic score. Through local update and cache reuse, the system can reduce the amount of repeated calculation in continuous rendering, and keep stroke response, screen refresh and aesthetic feedback in sync.

5 Realization and experimental verification of digital painting system

5.1 Experimental environment, painting dataset and evaluation index configuration

In order to verify the performance of the digital painting system combining computer graphics and aesthetic optimization in terms of rendering efficiency, image quality, aesthetic scoring and interactive response, a unified experimental environment is built, and an image sample set for digital painting task is constructed. The experimental platform uses a high-performance GPU to complete model training, brush rendering and aesthetic scoring reasoning, and the CPU is responsible for data preprocessing, layer scheduling and local update task management. In the process of system implementation, stroke geometry coding, texture perturbation generation, layer hybrid rendering and aesthetic scoring network are completed under the PyTorch framework. The local canvas refresh module uses a cache index

mechanism to record the pixel blocks affected by the editing area to reduce the computational consumption caused by repeated rendering.

The painting dataset consists of public digital painting images, stylized image samples, and system-generated samples, and contains a total of 12,000 images. The samples cover four main painting styles, including oil painting, watercolor, Chinese painting and ink painting, and sketch. Each type of image retains the original image, style label, resolution information, local texture area, and manual aesthetic score records. In order to meet the needs of model training and system validation, the data set is divided into training set, validation set and test set according to the proportion of 70%, 15% and 15%. The training set is used to learn the parameters of the aesthetic scoring model and optimize the painting parameters, the validation set is used to adjust the loss weight, learning rate and texture perturbation parameters, and the test set is used to compare the rendering quality and interactive performance under different system schemes.

The evaluation metrics are set around the system rendering results and user interaction process. Rendering efficiency is measured by frame rate and average delay to reflect the real-time response ability of the system in continuous rendering state. Image quality is evaluated by SSIM and LPIPS, which describe structural similarity and perceptual difference respectively. The Aesthetic optimization effect was evaluated by Aesthetic Score, Color Harmony, Composition Balance and Texture Detail to measure color harmony, composition stability and texture detail performance. The resource consumption is recorded by GPU Memory to avoid excessive computational burden when the model improves the picture quality. The above indicators can form a more complete experimental evaluation system from the four perspectives of efficiency, quality, aesthetics and resources. To ensure the rendering performance, image quality and aesthetic scoring results are comparable, the hardware platform, software framework, dataset size, training parameters and evaluation indicators are uniformly set in the experimental process, and the specific configuration is shown in Table 4.

Table 4: Experimental environment, painting dataset and evaluation index configuration table

Categories	Project	Configuration values
Hardware	CPU	Intel Xeon Silver 4314 × 2
	GPU	NVIDIA RTX 4090 24GB
	Memory	128 GB
Software	Operating System	Ubuntu 22.04
	Framework	PyTorch 2.2
Dataset	Painting Samples	12000 images
	Style Categories	Oil Painting, Watercolor, Ink Painting, Sketch
	Training/Validation/Test	70%/15%/15%
Rendering	Canvas Resolution	1024 × 1024
	Stroke Sampling Rate	120 Hz
Optimization	Batch Size	32
	Learning Rate	0.001
	Epoch	150
Evaluation	Efficiency Metrics	FPS, Latency, GPU Memory
	Image Quality Metrics	SSIM, LPIPS
	Aesthetic Metrics	Aesthetic Score, Color Harmony, Composition Balance, Texture Detail

During the experiment, the same data partition, canvas resolution and hardware environment are used in each comparison method to avoid the interference of platform differences on the performance results. All images were uniformly scaled to 1024 × 1024 resolution and pixel normalized before input to the model. The output range of the aesthetic scoring model was set to 0-10 points, and the higher the score, the better the overall quality of the picture. A lower LPIPS value indicates a smaller perceptual difference; Higher FPS and lower Latency indicate a more responsive system interaction. This configuration provides a unified test basis for subsequent experiments on rendering efficiency, image quality and aesthetic optimization.

5.2 Rendering efficiency, image quality, and Interactive Response performance verification

In a unified experimental environment, this paper selects CPU Rendering, GPU Rasterization, Neural Style Rendering and the Proposed System for comparison and verification. CPU Rendering adopts traditional pixel-by-pixel Rendering method, GPU Rasterization adopts conventional graphics pipeline acceleration method, Neural Style Rendering adopts whole image neural style rendering method. The Proposed System is a digital painting system that integrates brushstroke coverage, texture perturbation, aesthetic feedback and local update mechanism. In order to uniformly compare the change range of different methods in efficiency and quality indicators, the relative change rate is defined as follows:

$$\Delta P=\frac{P_{test}-P_{base}}{P_{base}}\times 100\% \tag{18}$$

where, ΔP represents the relative rate of change; Ptest represents the index value of the method to be verified. Pbase represents the index value of the baseline method. FPS and SSIM are positive indicators, and higher values represent better frame rate and structure preservation ability. Latency, LPIPS, and GPU Memory are inverse metrics, where lower values represent lower interaction latency, perceptual difference, and graphics memory footprint. In the experiment, CPU Rendering is used as the basic reference, and the other methods are compared horizontally.

To present the differences in running efficiency, image quality and resource overhead of different rendering schemes, Table 5 presents the average performance results of the four categories of methods on the test set.

Table 5: Comparison results table of rendering efficiency and image quality under different methods

Methods	FPS	Latency/ms	SSIM	LPIPS	GPU Memory/GB
CPU Rendering	18.6	92.4	0.821	0.214	2.1
GPU Rasterization	42.3	41.7	0.846	0.185	4.8
Neural Style Rendering	31.5	63.2	0.858	0.172	6.3
Proposed System	58.7	24.6	0.902	0.118	5.6

From the perspective of data distribution, CPU Rendering has low video memory occupation, but the frame rate and latency metrics are difficult to support continuous rendering. GPU Rasterization is better than CPU Rendering in FPS and Latency, which indicates that graphics hardware parallel computing can significantly reduce pixel-level rendering overhead. SSIM and LPIPS of Neural Style Rendering outperform traditional

rendering methods, but whole-graph reasoning keeps their interaction latency at a high level. The FPS of the Proposed System reaches 58.7, the Latency drops to 24.6 ms, the SSIM reaches 0.902, the LPIPS drops to 0.118, and the video memory occupation is 5.6 GB. It shows that the system has a good balance between improving the picture quality and maintaining the interactive performance.

In order to visualize the speed difference of each method in the real-time rendering process, Figure 6 shows the comparison between the rendering frame rate and the interaction delay.

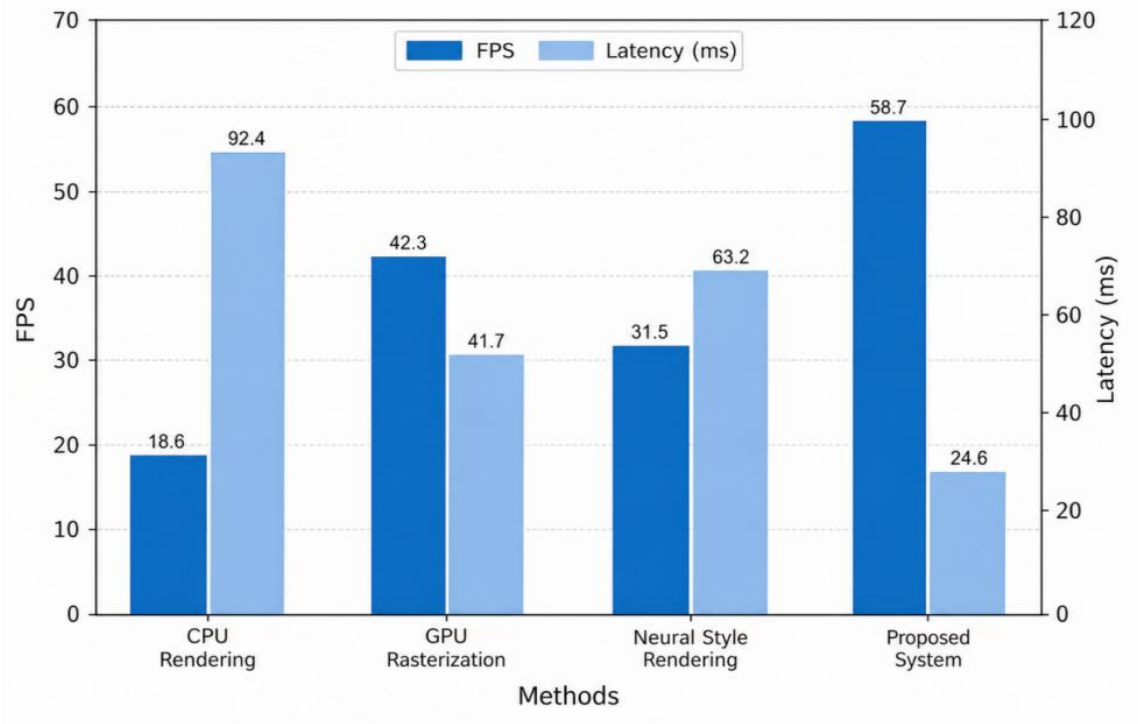

Figure 6: Comparison of rendering frame rate and interaction delay under different methods

In the comparison of Rendering frame rate and interaction delay, the corresponding FPS of the Proposed System is higher than that of GPU Rasterization and Neural Style rendering, and the Latency is lower than that of the three comparison methods. This phenomenon is related to local region update, cache reuse, and lightweight aesthetic score inference. The system only refreshed the affected area when the user drew continuously, and did not repeat the generation of the whole canvas, so that the graphics rendering pipeline could provide a real-time computing basis for subsequent aesthetic optimization.

In order to further verify the output Stability of the system under different painting styles, the five indicators of FPS, SSIM, Aesthetic Score, Texture Detail and Response Stability are calculated on four test samples of oil painting, watercolor, ink painting and sketch. FPS reflects rendering efficiency, SSIM reflects structure preservation ability, Aesthetic Score and Texture Detail reflect picture quality. Response Stability reflects the stability of the response during continuous interaction. The generation performance of the system under different painting styles is shown in Figure 7.

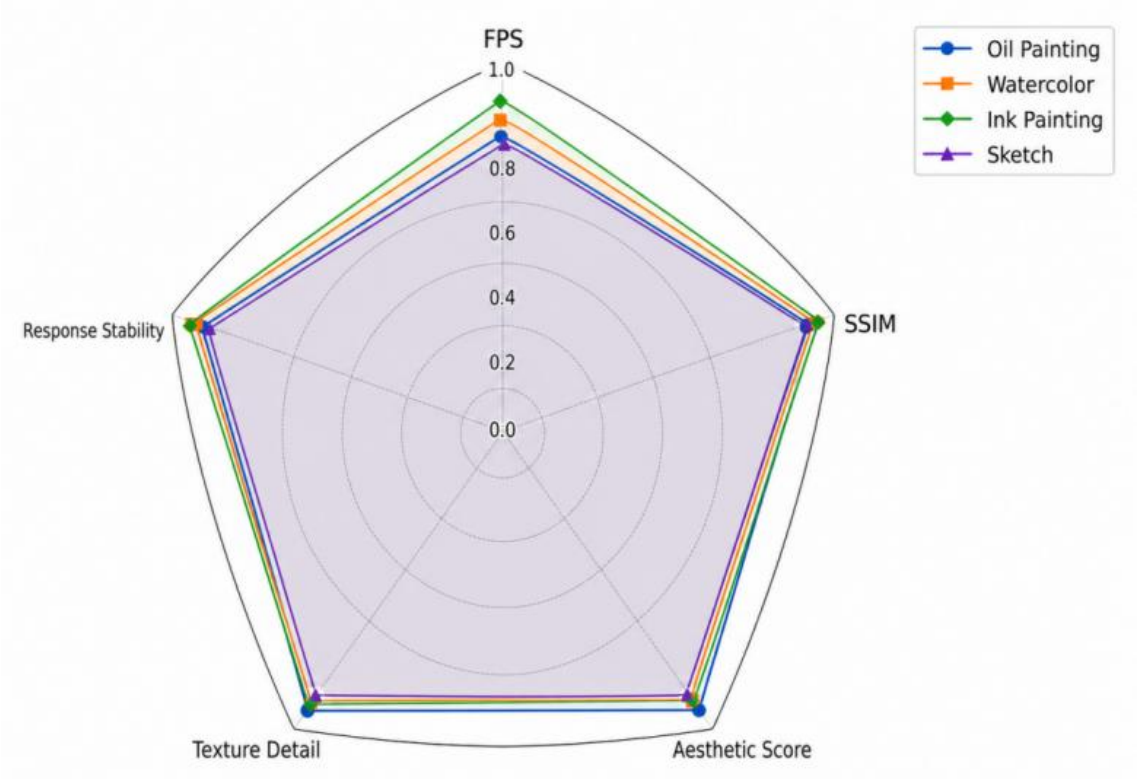

Figure 7: Radar chart of system generation performance index under different painting styles

In the performance distribution of different style samples, the oil painting samples have higher values in Texture Detail and Aesthetic Score, indicating that brush thickness, local texture and color level can be well preserved. The Response Stability of watercolor samples is relatively stable, which indicates that transparent overlapping and soft edge diffusion do not cause obvious response fluctuations. The FPS and SSIM of the ink samples remain at a high level, which indicates that the monochromatic hierarchy and structural boundaries have little impact on the rendering load. The sketch samples are more balanced in terms of structure preservation and response stability, and the areas with dense lines do not cause significant delay rise. This result corresponds to the overall performance data in Table 5, indicating that the proposed system does not only perform well on a single style sample, but maintains a relatively stable generation performance under different painting styles.

5.3 Analysis of aesthetic optimization effect and ablation experimental results

The effectiveness of the aesthetic optimization module needs to be jointly verified by the picture quality change before and after optimization and the module ablation results. The experiment takes four test samples of oil painting, watercolor, ink painting and sketch as objects, and statistics the overall aesthetic score, color coordination, composition balance and texture detail performance. The key modules are removed on the same test set to compare the output quality and running performance of different model versions.

In order to compare the influence of aesthetic optimization on different painting styles, the score changes of various samples before and after optimization were counted in the experiment, and the results are shown in Table 6.

Table 6: Comparison table of image scores of different styles before and after aesthetic optimization

Style of painting	Optimization state	Aesthetic Score	Color Harmony	Composition Balance	Texture Detail
Oil Painting	Before Optimization	7.24	7.11	7.06	7.31
	After Optimization	8.61	8.48	8.29	8.57
Watercolor	Before Optimization	7.08	6.94	6.88	7.16
	After Optimization	8.42	8.35	8.12	8.33
Ink Painting	Before Optimization	7.15	7.03	6.97	7.09
	After Optimization	8.36	8.18	8.21	8.26
Sketch	Before Optimization	7.02	6.86	6.91	7.04
	After Optimization	8.29	8.07	8.13	8.22

In the data, four types of painting styles are steadily improved after aesthetic optimization. The Aesthetic Score of oil painting samples is increased from 7.24 to 8.61, and the Texture Detail is increased from 7.31 to 8.57, indicating that texture perturbation, edge diffusion and brush stroke thickness correction can improve the hierarchical performance of thick painted areas. The Color Harmony of watercolor samples is improved from 6.94 to 8.35, and the color offset in transparent overlapping areas is suppressed. The Composition Balance of ink samples is improved from 6.97 to 8.21, and the subject position, white space relationship and ink diffusion area are more stable. The Texture Detail of the sketch samples is increased from 7.04 to 8.22, and the line density, edge gradient, and local light and dark relations are more balanced. The lifting direction of the four types of samples is consistent with the color distribution, composition shift and texture response feature design described above.

To present the changes in local picture quality before and after aesthetic optimization, Figure 8 shows the comparison of local details under different painting styles.

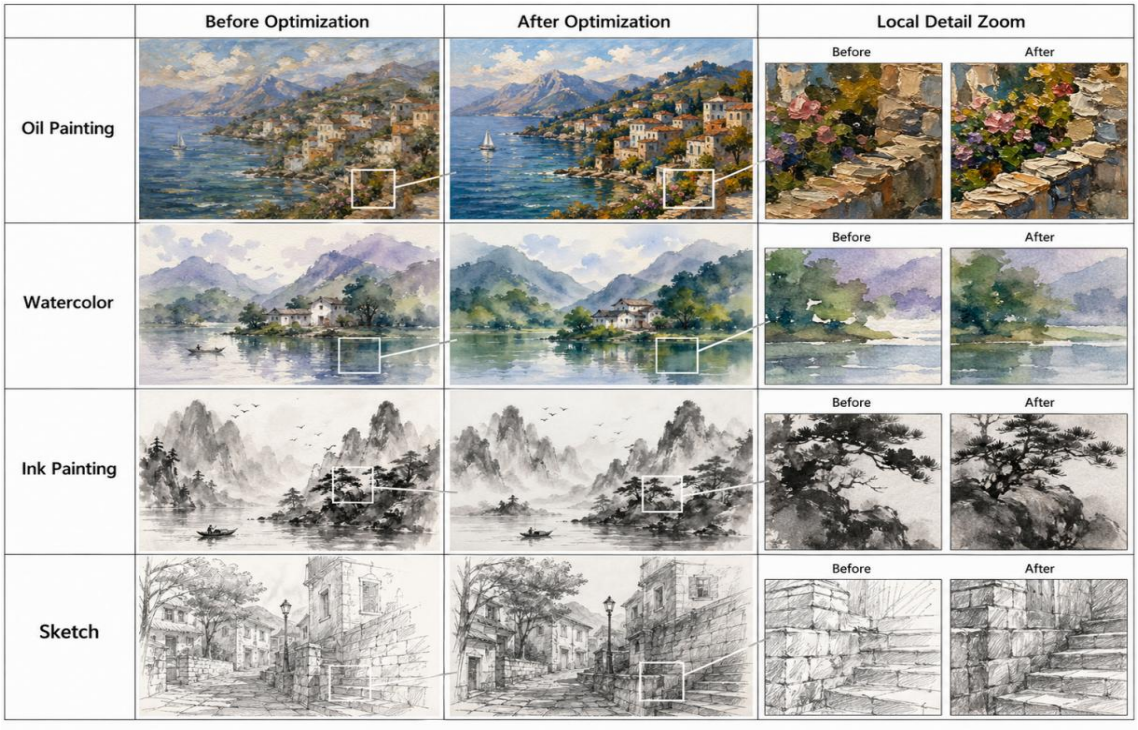

Figure 8: Comparison of local detail effects before and after aesthetic optimization

In the local detail comparison, the samples before optimization mainly have problems such as unstable edge transition, uneven texture response and insufficient local color level. The changes of the optimized samples are mainly reflected in three aspects: the boundary of thick brush strokes in the oil painting area is more continuous, the transparent and faint transition in the watercolor area is smoother, and the line density, gray level and edge structure in the ink and sketch samples are more stable. This visual change corresponds to the score improvement of Color Harmony, Composition Balance and Texture Detail in Table 6, indicating that the aesthetic optimization module not only improves the overall score, but also produces an observable correction effect on local textures, color transitions and structural details.

In order to further judge the contribution of each module to the overall performance of the system, four groups of ablation models were set up in the experiment: removing the stroke coding module, removing the texture rendering module, removing the aesthetic optimization module, and removing the local update module, and the results were compared with the complete model, and the results are shown in Table 7.

Table 7: Table of ablation experiment results

Model	Aesthetic Score	SSIM	LPIPS	Latency/ms	FPS
w/o Stroke Encoding	7.82	0.871	0.151	28.9	51.4
w/o Texture Rendering	7.76	0.864	0.162	26.7	54.2
w/o Aesthetic Optimization	7.21	0.852	0.176	24.1	59.3
w/o Local Update	8.23	0.895	0.124	39.8	36.6
Full Model	8.58	0.904	0.116	24.6	58.7

In the ablation data, after removing the Aesthetic optimization module, Aesthetic Score is reduced to 7.21, SSIM is reduced to 0.852, and LPIPS is increased to 0.176, indicating that although the system can still complete the basic rendering without scoring feedback, it is difficult to continue to correct the quality of color, composition and texture. After removing the texture rendering module, LPIPS rises to 0.162, and the perceptual difference increases significantly, reflecting that paper particles, edge diffusion and brush roughness have a direct impact on local realism. After removing the stroke coding module, the Aesthetic Score is reduced to 7.82, and the SSIM is reduced to 0.871, indicating that the position, pressure, curvature and direction information support the maintenance of the drawing structure. After removing the local update module, the Aesthetic Score still reaches 8.23, but the Latency rises to 39.8 ms, and the FPS drops to 36.6, indicating that the module mainly assumes the guarantee of interactive real-time, and does not directly determine the aesthetic quality of a single frame.

To more clearly present the difference in the impact of different module removal on each index, Figure 9 shows the performance degradation of the ablation model compared with the complete model.

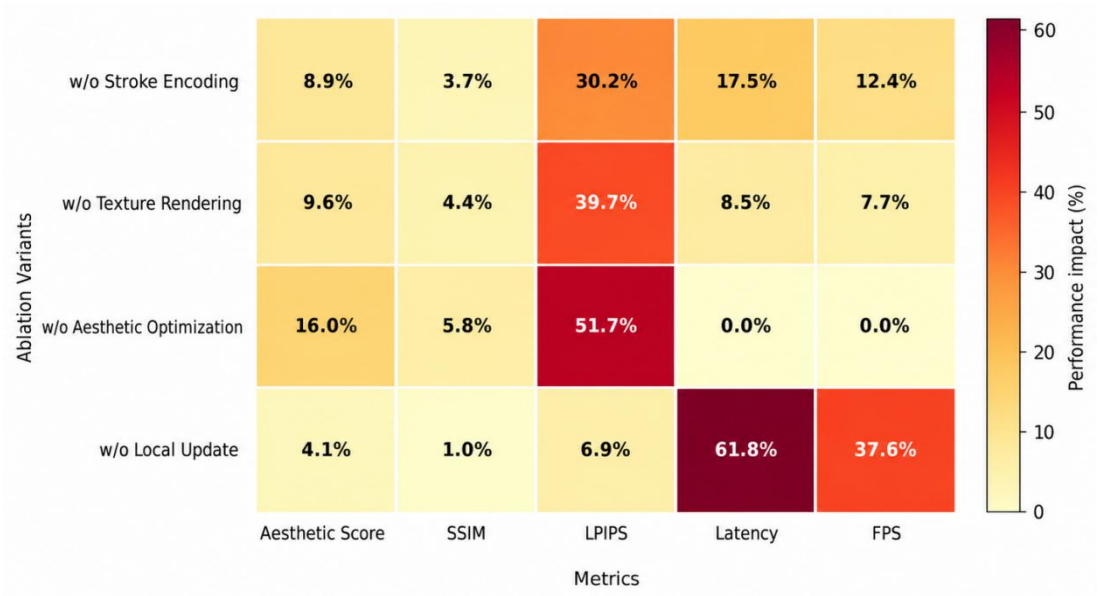

Figure 9: Heat map of ablation module impact on system performance

In the thermal distribution, removing the Aesthetic optimization module has more prominent changes in the dimensions of Aesthetic Score and LPIPS, removing the texture rendering module mainly affects LPIPS and SSIM, and removing the local update module mainly affects Latency and FPS. The performance degradation regions corresponding to different modules are not the same, which indicates that a clear division of function is formed inside the system: brush coding determines the reversibility of rendering structure, texture rendering determines the realism of local details, aesthetic optimization determines the improvement direction of picture quality, and local update determines the computational efficiency in the interaction process. The complete model maintains a good level in the five indicators of Aesthetic Score, SSIM, LPIPS, Latency and FPS, indicating that there is a complementary relationship between each module.

Based on the above data, it can be seen that the effectiveness of digital painting system does not come from a single algorithm stack, but from graphics rendering, aesthetic representation and interactive feedback together forming a closed loop. The graphics part converts stroke trajectories, layer states and texture parameters into controllable rendering results, the aesthetic scoring network converts color coordination, composition balance and texture details into optimization feedback, and the local update mechanism restricts the optimization results to the affected area to reduce the repeated calculation in the continuous rendering process. The performance comparison data verify the advantages of the system in frame rate, delay, structural similarity and perceptual difference. The aesthetic score data verify the adaptation ability of the optimization module to different painting styles. It can be seen that the proposed system can improve the visual quality of digital painting while maintaining real-time rendering response, and realize a complete technical link from stroke generation, graphics rendering, aesthetic evaluation to parameter feedback.

6 Conclusion

Focusing on the collaborative problem among graphics rendering, aesthetic evaluation and interactive feedback in digital painting system, this paper constructs a system architecture that integrates computer graphics and aesthetic optimization. The model integrates stroke

trajectory, layer state, color distribution, composition structure and texture response into a unified feature space. Through stroke coverage calculation, texture disturbance rendering, aesthetic score feedback and local incremental update, the continuous processing from rendering input to screen optimization is realized. Experimental results show that the system has good comprehensive performance in rendering frame rate, interaction delay, structural similarity, perceptual difference and aesthetic score, which can balance real-time rendering requirements and image quality improvement. Ablation experiments further verify the independent roles of stroke coding, texture rendering, aesthetic optimization and local update modules. In the future, it can be further optimized in larger scale painting samples, personalized aesthetic preference modeling, and cross-terminal deployment.

Funding

This research was supported by 2023 Annual Hubei Provincial Education Planning Project (General Project): "An Empirical Study on the Teaching Model of 'Promoting Learning and Teaching through Competitions' to Promote the Cultivation of Applied Capabilities—Taking the Visual Communication Major as an Example") (Grant No. 2023GB137)

References

[1] HU T, YI R, ZHU H, et al. Stroke-based neural painting and stylization with dynamically predicted painting region[C]//Proceedings of the 31st ACM International Conference on Multimedia. New York: ACM, 2023: 7470-7480.

[2] ZHANG X, XIAO Y, PENG J, et al. Confidence-based dynamic cross-modal memory network for image aesthetic assessment[J]. Pattern Recognition, 2024, 149: 110227.

[3] LI L, ZHI T, SHI G, et al. Anchor-based knowledge embedding for image aesthetics assessment[J]. Neurocomputing, 2023, 539: 126197.

[4] SHI T, CHEN C, LI X, et al. Semantic and style based multiple reference learning for artistic and general image aesthetic assessment[J]. Neurocomputing, 2024, 582: 127434.

[5] LI Z, YAN X, WEI X, et al. IAACLIP: Image aesthetics assessment via CLIP[J]. Electronics, 2025, 14(7): 1425.

[6] LI Q, WU M, CHEN D. PhotoGAN: A novel style transfer model for digital photographs[J]. Computers, Materials & Continua, 2025, 83(3): 4477-4494.

[7] TANG B, HU T, DU Y, et al. Curved-stroke-based neural painting and stylization through thin plate spline interpolation[J]. Scientia Sinica Informationis, 2024, 54(2): 301-315.

[8] SAWADA T, KATSURAI M. MambaPainter: neural stroke-based rendering in a single step[C]//SIGGRAPH Asia 2024 Posters. New York: ACM, 2024: 1-2.

[9] [9] ZHANG J, XU G, ZHANG X. HRL-Painter: optimal planning painter based on hierarchical reinforcement learning[J]. Neurocomputing, 2025, 636: 129972.

[10] TIAN X, TIRAKOAT S. The creative watercolor flow in the digital epoch: a novel approach to cross-disciplinary aesthetics integration[J]. TEM Journal, 2024, 13(4): 2829-2838.

[11] XIE Q, YU R. Image style transfer for exhibition hall design based on multimodal semantic-enhanced algorithm[J]. Computers, Materials & Continua, 2025, 84(1): 1123-1144.

[12] DONG Y, LIU S, LI Y, et al. Aesthetic-aware adversarial learning network for artistic style transfer[J]. Neurocomputing, 2025, 646: 130431.

[13] ZHAO H, ZHANG B, YANG Y J. Contrastive attention and fine-grained feature fusion for artistic style transfer[J]. Journal of Visual Communication and Image Representation, 2025, 110: 104451.

[14] SHI H, GUO J, KE Y, et al. Personalized image aesthetics assessment based on graph neural network and collaborative filtering[J]. Knowledge-Based Systems, 2024, 294: 111749.

[15] NIE X, HUANG S, GAO X, et al. MRAM: multi-scale regional attribute-weighting via meta-learning for personalized image aesthetics assessment[J]. Knowledge-Based Systems, 2024, 304: 112546.

[16] YAN X, SHAO F, CHEN H, et al. Hybrid CNN-transformer based meta-learning approach for personalized image aesthetics assessment[J]. Journal of Visual Communication and Image Representation, 2024, 98: 104044.

[17] YANG S, KE Y, WANG K, et al. A self-supervised image aesthetic assessment combining masked image modeling and contrastive learning[J]. Journal of Visual Communication and Image Representation, 2024, 101: 104184.

[18] GAO F, LIN Y, SHI J, et al. AesMamba: universal image aesthetic assessment with state space models[C]//Proceedings of the 32nd ACM International Conference on Multimedia. New York: ACM, 2024: 7444-7453.

[19] ZHAO X, SHI L, HAN Z, et al. A mobile image aesthetics processing system with intelligent scene perception[J]. Applied Sciences, 2024, 14(2): 822.

[20] JIN X, LV J, ZHOU X, et al. Aesthetic image captioning on the FAE-Captions dataset[J]. Computers and Electrical Engineering, 2022, 101: 107866.

[21] ZOU X, CHEN Y, WANG Z, et al. From fragment to one piece: a review on AI-driven graphic design[J]. Journal of Imaging, 2025, 11(9): 289.

[22] ZHOU T, LI Y, ZHANG X, et al. Cross-modality interactive attention network for AI-generated image quality assessment[J]. Pattern Recognition, 2025, 166: 111644.