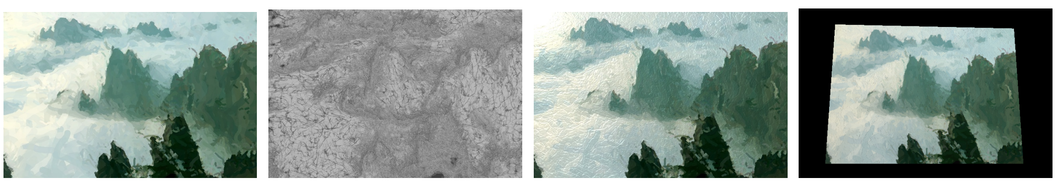
Synthesizing Artistic Realism: Stroke Painting Algorithms with Shader Enhancements

Qixiang Chen* **Huilin Chen**

The Australian National University

Canberra, Australia



(a) Stroke (Raw) Image (b) Height Field (c) 2D Canvas (d) 3D Canvas

Abstract

In this paper, we present an enhanced method for synthesizing artistic realism through advanced stroke painting algorithms with shader enhancements. Building upon the foundational work [3] our approach extends the rendering of realistic brush stroke textures to both 2D and 3D canvases. We begin by generating brush strokes on real photos, combining multiple layers of strokes of varying sizes to create depth and complexity. These strokes are used to generate height field based on pre-defined opacity and height maps, which are essential for calculating normals and applying Phong shading for realistic lighting effects. Our novel contributions include the implementation and comparison of bump mapping and displacement mapping techniques to analyze their effects on visual realism in 3D. While bump mapping provides a smoother and more integrated appearance, displacement mapping offers more pronounced and detailed textures by altering the mesh geometry. Additionally, we introduce a dynamic rotating mesh object demonstration that simulates the experience of viewing a painting in an exhibition hall, enhancing the overall viewer experience. The results demonstrate that our method significantly improves the realism of digital brush strokes, offering a more vivid and lifelike representation. Our extension to 3D canvases and the comparative analysis of mapping techniques provide valuable insights and pave the way for future research in non-photorealistic rendering. This work marks a significant advancement in the field of digital painting, bridging the gap between 2D and 3D applications and contributing to the ongoing evolution of artistic realism in computer graphics.

1 Introduction

In recent years, the pursuit of photorealistic rendering in computer graphics has extended beyond the mere replication of real-world objects to the synthesis of artistic realism. One area of particular

*Both authors contributed equally to this work while enrolled as Bachelor students at ANU, specifically for the course: Computer Graphics (COMP4610) in 2024.

interest is the rendering of brush stroke textures in digital paintings, aiming to mimic the intricate details and tactile qualities of traditional art. This project builds upon the foundational work presented by Aaron Hertzmann [3], which introduced methods for rendering realistic brush stroke textures in 2D. While Hertzmann's work [4] laid the groundwork for generating convincing textures, it was limited to two-dimensional(plain) implementations and did not explore the potential of three-dimensional applications.

The primary objective of our research is to enhance the realism of brush stroke textures and extend these techniques to 3D canvases, thereby offering a more immersive and comprehensive visualization of digital paintings. To achieve this, we introduced several novel contributions and improvements. We began by generating brush strokes on a real photo, combining multiple layers of strokes of varying sizes to create depth and complexity. These strokes were then used to create opacity and height maps, which were essential for subsequent normal and height field calculations.

Building on this 2D foundation, we implemented both bump mapping and displacement mapping techniques to analyze their effects on the visual realism of the textured paint. Our comparative study provided valuable insights into the strengths and limitations of each method, particularly under different lighting conditions. Furthermore, we extended our work to 3D by creating a cuboid triangular mesh with detailed UV maps, allowing us to render the paint on both sides of a real canvas. This 3D implementation introduced a dynamic scene, simulating the experience of viewing a painting in an exhibition hall and adding a new dimension to the viewer's experience.

Our research not only enhances the visual realism of digital brush strokes but also bridges the gap between 2D and 3D rendering techniques. By integrating sophisticated shaders and employing Phong shading for dynamic scenes, we have achieved a more real and lifelike representation of digital paintings. This paper elaborates on the methodology, experiments, and findings of our work, underscoring the advancements made over previous techniques. It provides a solid foundation for future exploration in the field of artistic realism within computer graphics.

2 Previous Works

Our work builds upon a rich history of research in the field of non-photorealistic rendering (NPR), particularly focusing on the rendering of brush stroke textures [5]. The foundational work of stroke-based rendering (SBR) has been instrumental in shaping the techniques and methodologies we employed and extended in our research. It [3] introduced a method for rendering realistic brush stroke textures by combining an ordered list of brush strokes, a shading model, and stroke height maps. This technique produced a raw color image by compositing brush strokes and then generating a height field for bump mapping using the Phong shading model. Our work directly builds on this foundation by enhancing the realism of the brush stroke textures. We achieved this by combining multiple layers of strokes with varying sizes, generating more complex and lifelike textures. Additionally, we extended Hertzmann's approach to 3D canvases, which was not explored in the original work.

In [1] Hertzmann introduced a technique for creating painterly images using curved brush strokes of different sizes. This approach aimed to mimic the aesthetic qualities of hand-painted images by varying the stroke sizes and orientations. Our method incorporates this idea by using multiple layers to add depth and complexity to the visual representation. This layering technique enhances the overall realism of the textured paint, aligning with Hertzmann's goal of capturing the essence of traditional painting techniques in digital form.

The paper [2] expanded the application of painterly rendering techniques to video and interactive systems. This work demonstrated the potential for real-time application of NPR techniques, allowing users to interact with and manipulate the rendered images dynamically. Our project draws inspiration from this interactive aspect by introducing a dynamic rotating scene, simulating the experience of viewing a painting in an exhibition hall. This dynamic element adds a new dimension to the viewer's experience, making the digital paintings appear more vivid and lifelike from different angles.

The connections between our work and Hertzmann's foundational research are evident in several key areas. We have adopted and extended the techniques for generating and compositing brush strokes, utilizing height fields and normal mapping to enhance realism. By integrating bump and displacement mapping, we provided a comparative analysis of these techniques, highlighting their respective advantages and disadvantages. Furthermore, our extension to 3D canvases and the incorporation of dynamic scene rotation represent significant advancements over the original 2D-focused methods.

3 Proposed Method

3.1 Baseline Overview

Our method consists of several key components that build upon previous work in stroke generation algorithm [1, 3, 5, 6]. Combine with the different shaders in 2D and 3D rendering, we aim compare the visual effect that how these combination can enhance the realism of digital paintings, where the pipeline can be divided into four stages:

1. **Stroke-based rendering (SBR):** This stage involves producing a "raw" color image by compositing brush strokes, simulating traditional painting techniques.
2. **Height field generation:** A height field is calculated for each pixel, representing the height of the painted surface caused by the brush strokes.
3. **Fast 2D rendering:** The raw color image is enhanced using bump-mapping and the Phong shading model to generate a realistic 2D painting.
4. **3D demonstration with different shaders:** This stage involves 3D paint rendering using both bump and displacement mapping to create a more immersive and detailed visual effect.

3.2 Stage 1: SBR Approach Detail Revisit

Brush Stroke Model.[2] **[1]** the algorithm with a stroke model that consists of parametric customizable strokes to produce different stroke positions and appearances. Each stroke point (stipple) has an (x, y) position with radius r and texture color information. Together, these form a stroke that is painted on a canvas. The canvas is initialized as blank and later used by the algorithm to paint each stroke.

To create a painterly image, the list of strokes is key for simulating the paint effect. We apply a Gaussian blur to the image, using r as the kernel size for the reference image, and calculate the difference with the original image to determine the loss of each pixel, which helps in determining the start point of each stroke. With the start point identified, we follow the gradient changes over the x and y axes to generate lists of control points until the maximum stroke length is reached. The cubic B-spline curve [7] is then used to smooth the sparse control points, following the equation:

$$\mathbf{C}(t) = \sum_{i=0}^{n} \mathbf{P}_i B_{i,k}(t)$$

where $\mathbf{C}(t)$ is the position on the curve at parameter t, $\mathbf{P}_i$ are the control points, and $B_{i,k}(t)$ are the B-spline basis functions of degree k. The cubic B-spline specifically uses $k = 3$. The basis functions $B_{i,k}(t)$ are defined recursively as:

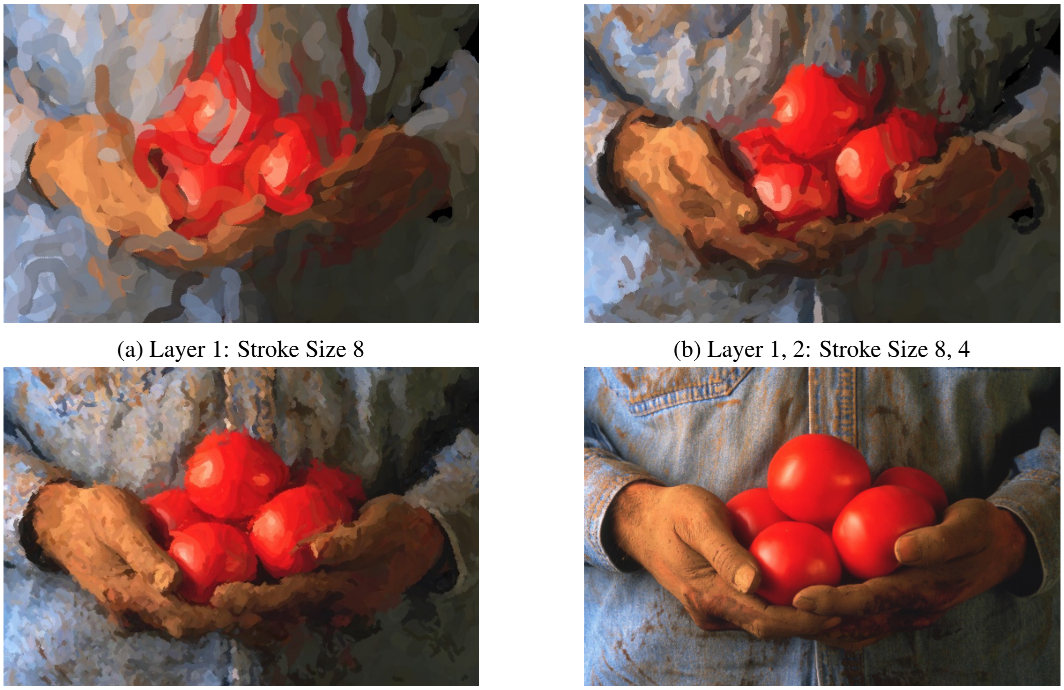
$$B_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$B_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} B_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(t)$$

The samples of the stroked path are shown in Figure 4a, and each stroke is rendered by alpha-compositing on the canvas [5].

Multiple-Layer Rendering. To generate realistic brush strokes, we create an ordered list of brush strokes with varying sizes. Specifically, we use three layers of strokes with brush sizes of 8, 4, and 2. These three layers are combined to create a composite image that mimics the appearance of a real painting. Each layer contributes to the final texture, ensuring that the brush strokes are visible and well-integrated. Figure 2 demonstrates the accumulation of layers.

[2]Despite the availability of an open-source implementation that relies on OpenGL, we opted to build the project from scratch using the paper's pseudocode to ensure full control and customization.



(a) Layer 1: Stroke Size 8

(b) Layer 1, 2: Stroke Size 8, 4

(c) Layer 1, 2, 3: Stroke Size 8, 4, 2

(d) Original image

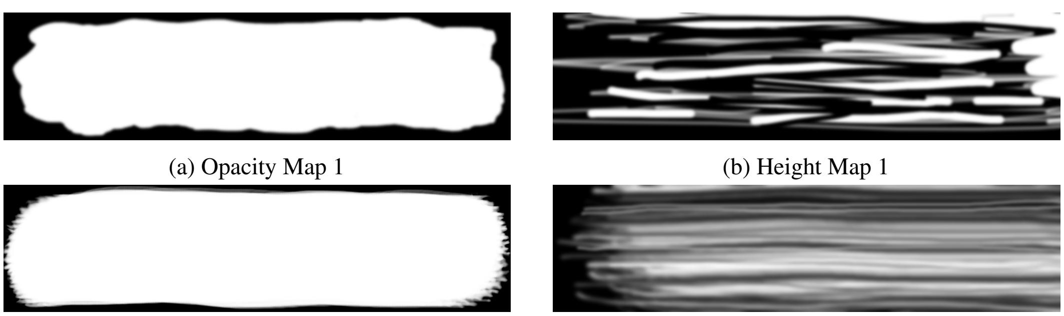
Figure 2: Layer accumulation of strokes with different sizes (8, 4, and 2) to create a composite image.

3.3 Stage 2: Height Field Computation

Opacity and Height Map Creation. To simulate the height of the brush strokes on the canvas, we create both opacity and height maps, as shown in Figure 3. These maps are essential for later stages where we perform bump mapping and displacement mapping.

The opacity map (3a, 3c) represents the transparency level of each brush stroke, effectively capturing the varying intensity and thickness of paint applied by different strokes. This map is crucial for ensuring that overlapping strokes interact correctly, allowing for the creation of complex textures and color blending effects that mimic real-world painting techniques.

The height map (3b, 3d), on the other hand, encodes the elevation of the paint on the canvas. This is particularly important for achieving realistic 3D effects, as it allows us to simulate the physical depth and volume of the paint. By representing the height of the brush strokes, the height map facilitates accurate lighting and shading effects in the rendering process, enhancing the overall realism of the digital painting.



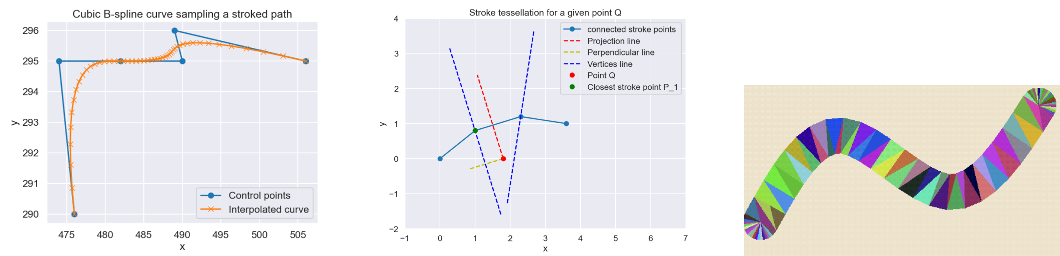
(a) Opacity Map 1

(b) Height Map 1

(c) Opacity Map 2

(d) Height Map 2

Figure 3: Stroke opacity and height maps



(a) Example of a cubic B-spline curve used for stroke path sampling.

(b) One stroke tessellation example

(c) Stroke tessellation overview [4]

Figure 4: Stroke sampling and tessellation

Stroke Brush Texture Mapping Using UV Coordinates. To map the curved strokes onto the predefined opacity or height maps, we employ stroke tessellation in the form of a triangle strip, following the method outlined by Hertzmann [3]. As depicted in Figure 4c, the stroke is segmented into a series of triangles. For each triangle, the line of vertices runs perpendicular to the line connecting adjacent stroke points, serving as the extended normal at each point, as illustrated in Figure 4b.

For points within the stroke, we compute the UV coordinates utilizing the tangents and normals associated with each stroke point. This ensures that the textures conform precisely to the strokes' curvature. First, we compute the tangents and normals for the stroke points as follows:

$$\mathbf{T}_i = \frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{\|\mathbf{P}_{i+1} - \mathbf{P}_i\|},$$
$$\mathbf{N}_i = (-\mathbf{T}_i^y, \mathbf{T}_i^x)$$

where $\mathbf{T}_i$ is the tangent vector and $\mathbf{N}_i$ is the normal vector at the i-th stroke point $\mathbf{P}_i$. Next, for a point Q inside the stroke, we identify the closest stroke point P_i and calculate its projection onto the tangent of that stroke point. The process is detailed as follows:

- **Identify the closest stroke point P_i:**
$$P_i = \arg\min_i \|Q - P_i\|$$
- **Calculate the u coordinate:**
$$l = \mathbf{T}_i \cdot (Q - P_i),$$
$$s = \|\mathbf{T}_i\|^2$$
- **Adjust for points to the left of the stroke point (if $l < 0$):**
$$l_{new} = \mathbf{T}_{i-1} \cdot (Q - P_{i-1}),$$
$$s_{new} = \|\mathbf{T}_{i-1}\|^2$$
- **Determine u based on position:**
$$u = \begin{cases} \frac{(i-1) + \frac{l_{new}}{s_{new}}}{n}, & \text{if } l < 0 \\ \frac{i + \frac{l}{s}}{n}, & \text{otherwise} \end{cases}$$
- **Calculate the v coordinate:**
$$v = \left(\frac{\mathbf{N}_i \cdot (Q - P_i)}{r}\right) \times 0.5 + 0.5$$

This algorithm precisely maps each point on the stroke to the connected stroke line, represented by the tangent vector, and to the perpendicular line, represented by the normal vector. This precision ensures that the stroke textures are accurately aligned with the opacity and height maps. By strictly adhering to the geometric properties of the strokes, the algorithm not only enhances the visual continuity and integrity of the texture mapping but also facilitates seamless transitions between textures. This preserves the natural aesthetics and dynamism of the original brush strokes, effectively capturing the depth and height of each stroke.

3.4 Stage 3: Fast 2D Rendering Process

Painterly Rendering. With the list of strokes, the algorithm is able to composite the brush strokes to produce a raw color image. During each stroke's rendering process, a constant opacity of 0.7 is applied to approximate the effects observed in real paintings, blending the current texture color with the underlying canvas color.

Generating the Height Field. Once the composite image is created, we generate the height field for the entire image to represent the surface variations induced by the brush strokes. The strokes are layered according to their drawing and layer order. Each layer's rendering process involves calculating temporal weights to simulate the actual painting process. The temporal weight diminishes the drawing depth of each stroke as follows:

$$(1 + \cos(0.5 \cdot \pi \cdot \text{stroke\_idx}/\text{total\_strokes}))$$

Another factor is the stroke weight, which determines the balance between the opacity map and the height map applied to a stroke with radius r. We utilize a logistic function for this mapping, ensuring lower opacity for smaller brushes, where the parameters are chosen after extensive experiment:

$$\frac{1}{1 + \exp(-0.7 \cdot (\mathrm{r} - 4.5))}$$

Finally, we adjust the contrast of the height fields to reduce the sharpness of the brush strokes, thereby enhancing the visual quality of the rendered image.

Normal Mapping with Phong Shading. Using the height field, we perform normal mapping on the raw color image. This process involves using sobel filter to compute the surface normals from the height field and applying the Phong illumination model to simulate realistic lighting effects on the painting to have a fast inference result.

3.5 Stage4: 3D Paint rendering

Creating the Cuboid Triangular Mesh. For the 3D part of our work, we create a cuboid triangular mesh that serves as a canvas. This mesh has a default texture map that displays the painting on the front and back sides, while other orientations are set to a pre-defined color by default.

Generating UV Maps for the 3D Canvas. We generate two UV maps for the 3D canvas: one for the height field and one for the raw(stroked) paint. These maps allow us to accurately apply the textures onto the 3D mesh object. Using the generated UV maps, we map the stroked paint and height field onto the 3D canvas. This step ensures that the painting is accurately represented on the 3D surface.

Bump Mapping and Displacement Mapping. When doing the 3D mapping, we take the texture color of the paint and use the normal provided by the height map. Bump mapping perturbs the surface normals to create the illusion of depth. The new normal is computed using the height map. Displacement mapping [6] modifies the vertex positions directly based on the height map:

$$\mathbf{p}' = \mathbf{p} + k_n \mathbf{n} H(u, v)$$

Dynamic Scene Rendering. To enhance realism, we set up lights and cameras in the scene. Phong shading is used to simulate realistic lighting effects on the painting. We introduce scene rotation to create a dynamic view of the 3D painting. This rotation allows viewers to appreciate the depth and texture of the painting from different angles, enhancing the overall realism.

4 Experiments

4.1 Height Field Computation

As illustrated in Figure 6, the results of the height map generation and its compositional counterpart shown a well overlap effects of the stroke. While the algorithm effectively generates the map for the curved stroke using the predefined rectangular opacity and height maps, the edges of the height field appear rough due to intersections between the vertex lines of consecutive stroke points. Additionally, our approach does not account for the left and right circles at the start and end of the stroke points. Attempts to implement an approximation to smooth these transitions resulted in inconsistent and visually unappealing outcomes, particularly in cases of varying stroke lengths.

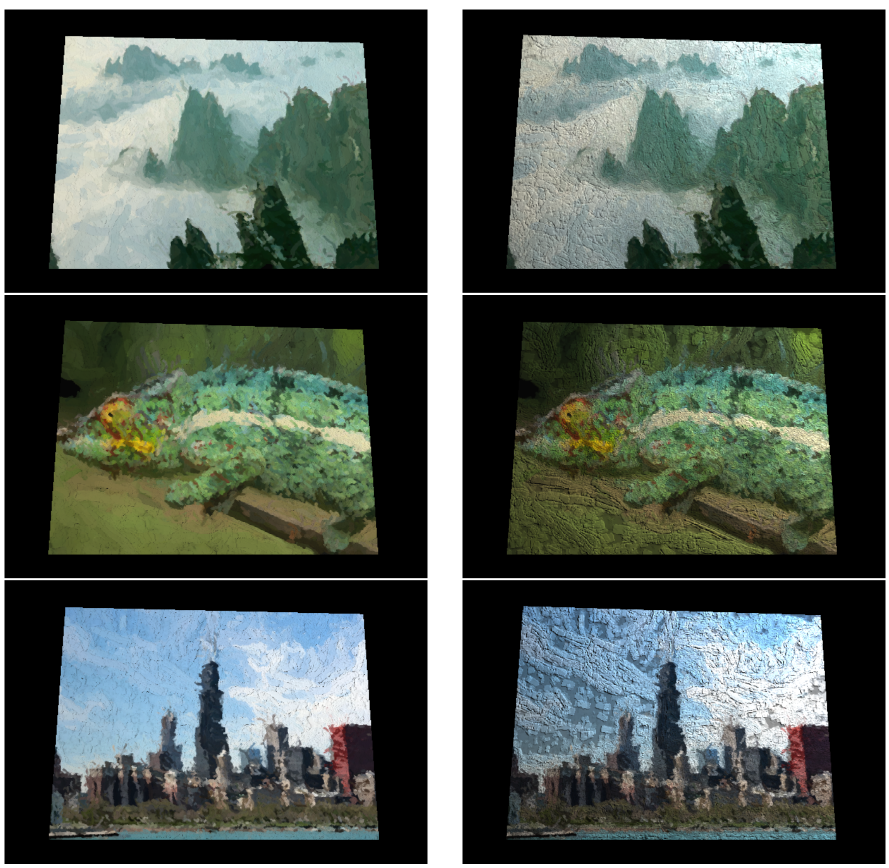
4.2 2D Experiments

To validate the effectiveness of our method in generating realistic textured paintings, we conducted a series of 2D experiments. The first step involved generating height fields from the brush strokes. Next, we focused on normal mapping and applying lighting effects to the generated height field to achieve realistic textured paint effects. In Figure 9, the left side are the generated height fields by calculating the normals and reducing contrasts. The right side are the rendered 2D paint with brush textured by phong shading.

4.3 Bump & Displacement

We compare the performance of bump mapping and displacement mapping on the 3D canvas. Bump mapping uses the height field normals to simulate surface variations, while displacement mapping actually deforms the mesh based on the height field. As shown in Figure 5, the strokes in the displacement mapping are more pronounced and the edges of the strokes are sharper, as this method changes the actual geometry of the paint surface. This results in a more tangible and distinct texture, especially noticeable at the edges where the geometry alters significantly. In bump mapping, the strokes appear more shallow and integrated into the surface, providing a smoother and more realistic appearance overall. However, the illusion of depth is limited to the lighting effects and does not affect the mesh geometry. Both methods produce visually appealing results, but displacement mapping shows a more exaggerated texture, making the brush strokes more prominent.

When in low lighting condition shown in Figure 8, the differences between the two techniques become more apparent. The brush strokes in the displacement mapping stand out more due to the actual changes in geometry, while the bump mapping maintains a subtle, less pronounced texture.



(a) Bump Mapping (b) Displacement Mapping

Figure 5: Bump and Displacement Mapping on 3D Canvas

5 Discussion

In this project, we significantly advanced the realism of brush stroke textures by utilizing a three-layer combination that adds depth and complexity to the visual representation. We expanded beyond the traditional 2D focus by applying our texture mapping techniques to both sides of a 3D cuboid triangular mesh, enabling a more immersive visualization. We also explored the differences between bump and displacement mapping, finding that while displacement mapping offers more pronounced textures, it is computationally demanding. Bump mapping, while less detailed, provides a smoother integration of strokes, enhancing realism.

The computational demands of mapping both color textures and height fields for each frame are extensive, resulting in a processing speed of almost 1 fps. This highlights the necessity for further optimization in the 3D rendering process to improve performance without compromising visual quality.

Overall, the results are compelling and validate the final achievements of our methodology. They underscore the effectiveness of our rendering techniques in creating visually compelling and realistic digital paintings, thus confirming the potential of our approach in bridging the gap between 2D and 3D artistic representations.

6 Conclusion

We set out to enhance the realism of digital brush stroke textures and extend the techniques to 3D implementations. Our project successfully achieved these goals by combining multiple layers of strokes and using sophisticated normal mapping techniques, resulting in highly realistic 2D and 3D textures. The extension to 3D provided a new perspective on digital paintings, offering a more immersive and detailed visualization. The comparison of bump and displacement mapping highlighted the strengths and weaknesses of each method, providing valuable insights into their applications. Through this project, we have learned that combining multiple layers of brush strokes significantly enhances the realism of digital paintings and that extending texture mapping techniques to 3D surfaces requires careful handling of UV maps and mesh geometry.

6.1 Future Work

Future research can build on the strengths of this project, such as generating example normal maps for desired stroke settings and establishing a dataset containing pairs of paint images and corresponding normal maps. This dataset can then be utilized to train advanced neural networks to generate normal maps for any given paint image, potentially revolutionizing the way textures are applied in digital art. Additionally, exploring less computationally intensive alternatives to displacement mapping could enhance the feasibility of real-time applications, broadening the potential uses of our techniques in interactive media and virtual reality settings.

References

[1] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 453–460, 1998.

[2] Aaron Hertzmann. Painterly rendering for video and interaction. In *First International Symposium on Non-Photorealistic Animation and Rendering*, 2000.

[3] Aaron Hertzmann. Fast paint texture. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 91–ff, 2002.

[4] Aaron Hertzmann. Stroke-based rendering. In *Proceedings of the 2nd International Symposium on Non-Photorealistic Animation and Rendering*, pages 123–128. ACM, 2002.

[5] Aaron Hertzmann. A survey of stroke-based rendering. Institute of Electrical and Electronics Engineers, 2003.

[6] László Szirmay-Kalos and Tamás Umenhoffer. Displacement mapping on the gpu—state of the art. In *Computer graphics forum*, volume 27, pages 1567–1592. Wiley Online Library, 2008.

[7] Wikipedia contributors. B-spline — Wikipedia, the free encyclopedia, 2024. [Online; accessed 21-May-2024].

Additional Experiment Results

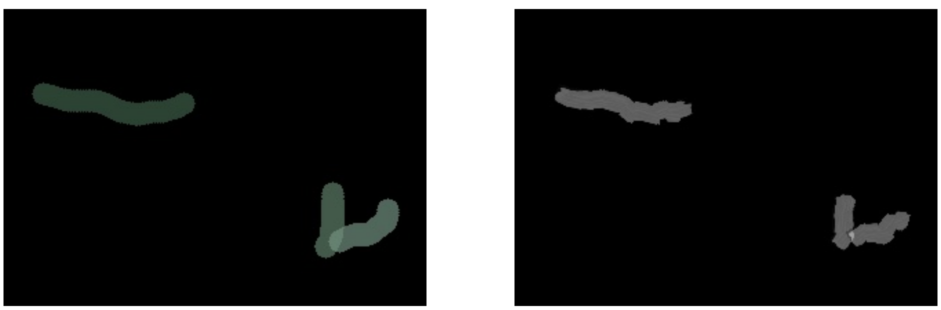
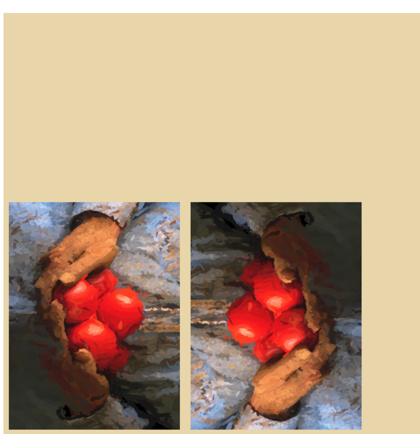
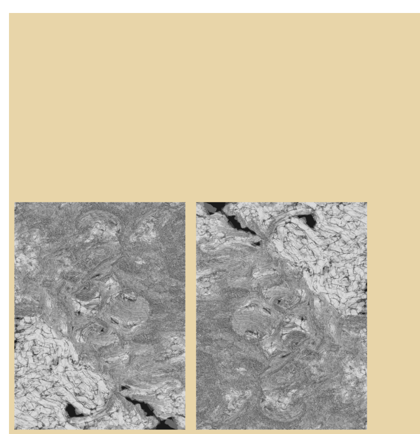


Figure 6: Example of a stroke with a radius of 8. *(left)* compositional rendering *(right)* height field rendering

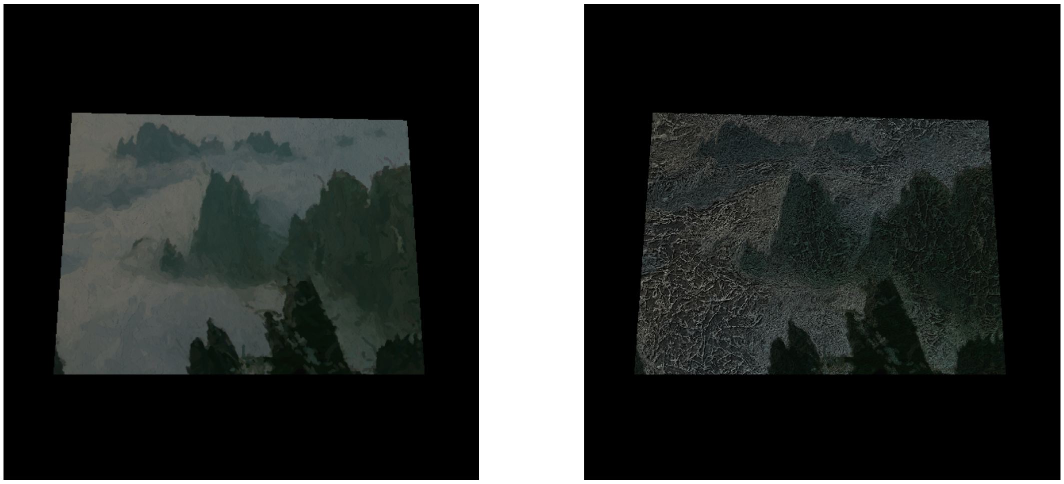


(a) UV Map Paint



(b) UV Map Height Field

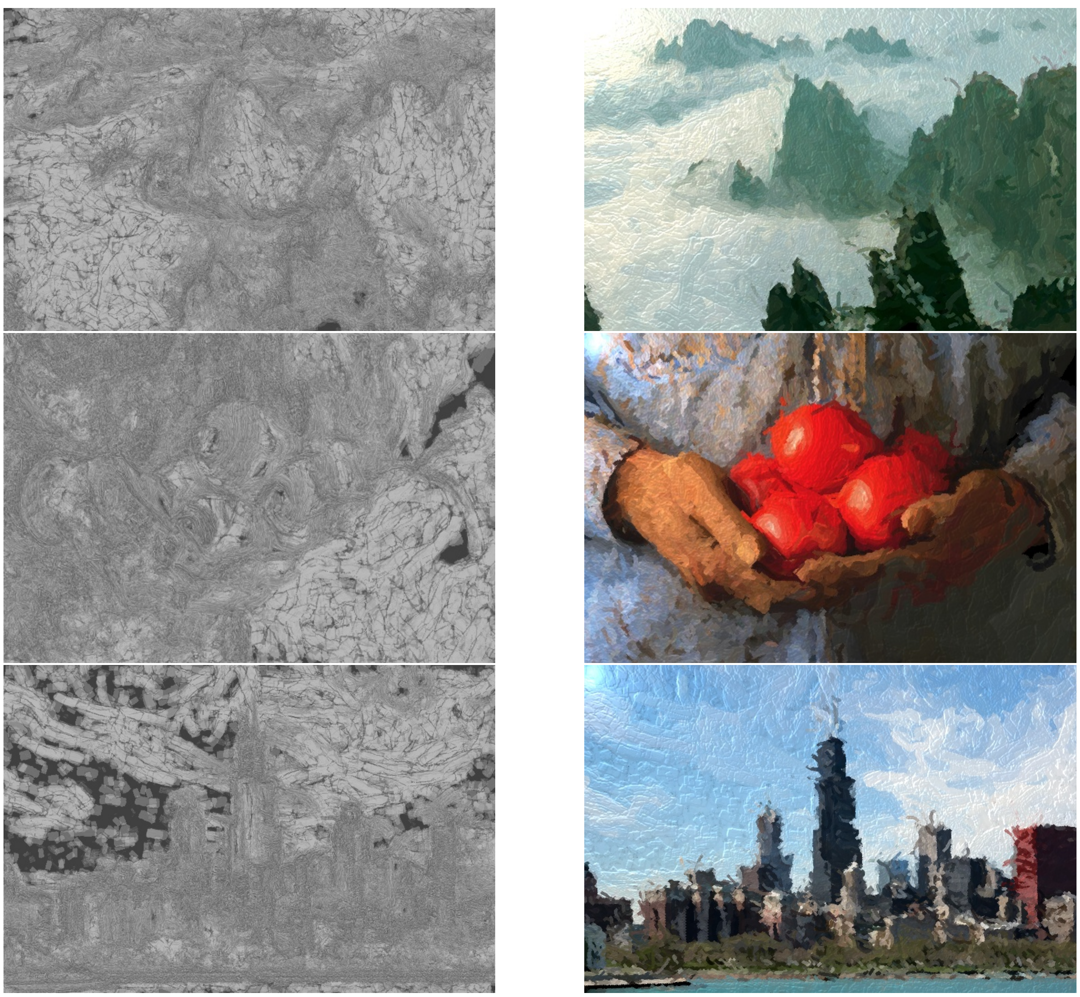
Figure 7: UV Maps for rendering textures on 3D canvas



(a) Bump Mapping in Dark Condition

(b) Displacement Mapping in Dark Condition

Figure 8: Bump and Displacement Mapping on 3D Canvas in Dark Lighting Condition



(a) Height Field

(b) Textured Paint

Figure 9: Height Fields and 2D textured paint