ISSN 2464-4617 (print)
SSN 2464-4625 (DVD)
Computer Science Research Notes
CSRN 3101
WSCG 2021 Proceedings

A Mobile Augmented Reality Application For Simulating Claude Monet's Impressionistic Art Style

Neil Patrick Del Gallego
De La Salle University
Taft Avenue, Malate
1004 Metro Manila, Philippines
neil.delgallego@dlsu.edu.ph

Cedric Lance Viaje
De La Salle University
Taft Avenue, Malate
1004 Metro Manila, Philippines
cedric_viaje@dlsu.edu.ph

Michael Ryan Gerra-Clarin
De La Salle University
Taft Avenue, Malate
1004 Metro Manila, Philippines
michael_gerra-clarin@dlsu.edu.ph

John Marvic Roque
De La Salle University
Taft Avenue, Malate
1004 Metro Manila, Philippines
john_marvic_roque@dlsu.edu.ph

Gary Steven Non
De La Salle University
Taft Avenue, Malate
1004 Metro Manila, Philippines
gary_non@dlsu.edu.ph

Jesin Jarod Martinez
De La Salle University
Taft Avenue, Malate
1004 Metro Manila, Philippines
jesin_martinez@dlsu.edu.ph

Jose Antonio Gana
De La Salle University
Taft Avenue, Malate
1004 Metro Manila, Philippines
jose_antonio_gana@dlsu.edu.ph

ABSTRACT

In this study, we showcase a mobile augmented reality application where a user places various 3D models in a tabletop scene. The scene is captured and then rendered as Claude Monet's impressionistic art style. One possible use case for this application is to demonstrate the behavior of the impressionistic art style of Claude Monet, by applying this to tabletop scenes, which can be useful especially for art students. This allows the user to create their own "still life" composition and study how the scene is painted. Our proposed framework is composed of three steps. The system first identifies the context of the tabletop scene, through GIST descriptors, which are used as features to identify the color palette to be used for painting. Our application supports three different color palettes, representing different eras of Monet's work. The second step performs color mixing of two different colors in the chosen palette. The last step involves applying a three-stage brush stroke algorithm where the image is rendered with a customized brush stroke pattern applied in each stage. While deep learning techniques are already capable of performing style transfer from paintings to real-world images, such as the success of CycleGAN, results show that our proposed framework achieves comparable performance to deep learning style transfer methods on tabletop scenes.

Keywords

augmented reality, mobile devices, image filter, image stylization, style transfer, painterly rendering

1 INTRODUCTION

Augmented reality is an interactive experience wherein digital objects are placed on the physical environment [BCL15]. This study proposes a mobile augmented reality application that allows the user to capture a scene with virtual 3D models and then applies a rendering technique that "paints" the scene using Claude Monet's impressionistic art style (Figure 1). A specific use case for this application is to demonstrate the visuals of Claude Monet's impressionistic art style, which can be useful to art students who are learning the fundamentals of painting. Using the mobile AR application, the user can place virtual 3D models on top of other tabletop objects which allows the user to compose a still life scene containing virtual models that are not readily available (e.g. A virtual dragon model placed with other tabletop objects, such as drinking glass and basket of fruits). These 3D models can coexist within the same scene in real-time.

Recently, deep learning methods are used to generate images painted with a certain art style [GEB16]. Image-to-image translation using generative adversarial networks was proposed, such as the release of Pix2Pix [IZZE17]. Then the release of CycleGAN introduced a new class of image-to-image translation methods that

Corresponding author email: neil.delgallego@dlsu.edu.ph.
The source code is available at Github: `https://github.com/NeilDG/AR-Impress-Me-Version-2`

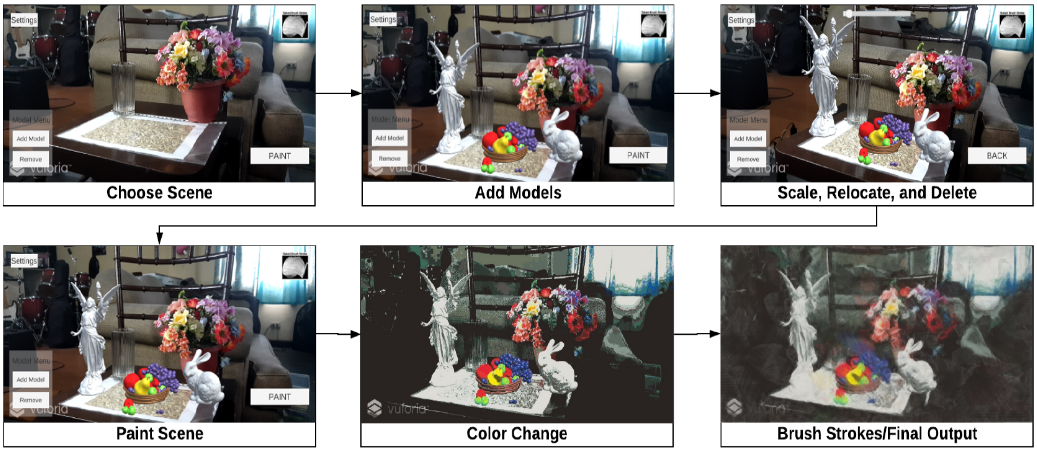


Figure 1: The proposed application. The user assembles a tabletop scene and then places one or more virtual 3D models. The scene is then captured and processed, which produces an art style similar to Claude Monet's paintings, using a pre-defined color palette and brushstroke settings consistent with an impressionistic art style.

Figure 2: Example of virtual models that can be placed in a tabletop scene. Standard 3D models commonly used for computer graphics applications are shown. From left to right: Lucy, Stanford Bunny, Stanford Armadillo, Utah Teapot. Our application can also load other custom-made 3D models.

do not need paired data [ZPIE17]. CycleGAN has been used for style transfer approaches and shows that it is comparable to other style transfer methods such as the method proposed by Gatys et.al (2016) [GEB16].

Early work showed a style of processing images and video with an impressionistic effect that looks like it was hand-painted [Lit97]. An optical flow field technique was applied that determines the brushstroke pattern frame by frame. Randomness is applied to the brush stroke attributes, such as length, color, and orientation, to enhance the hand-touched look. Similarly, a method proposed by Hertzmann (1998) paints an image with a series of spline brush strokes [Her98]. We observed that this method is effective in simulating impressionistic brush strokes and applied the same concept in the application. Other early brush stroke techniques were observed in literature [Hae90, HE04, KS04, KS06, CS07].

While deep learning methods are becoming the trend for artistic painting and non-photorealistic rendering, we observe that these methods are not tailored for images that consist of augmented 3D models, such as the case of using an AR application for painting a scene. We later show in our results that existing style transfer methods generate impressionistic images where the 3D models stand out unnaturally and do not blend properly with the overall image composition. Furthermore, our proposed framework is grounded on an analytical study of Monet's artworks, unlike other deep learning approaches that completely rely on individual paintings as training samples.

We organized this paper as follows: related work, concepts on impressionism, architectural framework, then lastly, discussion of results and conclusion.

2 RELATED WORK

2.1 Image Stylization

Image stylization is the process of taking an image as input, and producing a stylized version of it [RAGS01]. The contents of the image should remain the same. To achieve an artistic stylized image: computer vision techniques and machine learning are mostly used to transform the target images and their pixels [KCWI12]. We further discuss different image stylization methods under two categories, algorithmic and learning-based image stylization.

Algorithmic approaches in image stylization are those that rely on apriori information and typically render images in a deterministic or sometimes stochastic manner. Region-based techniques, which are typically applied to cartoon shading styles [CRH05, WXSC04], simulating stained glass [Mou03], simulating cubist painting [CH03], involve image segmentation as a pre-processing step [KCWI12]. Image stylization techniques that are applied to fine details of an image [SLKD16, KD08, KKD09b, STD$^+$16] typically involve edge-preserving smoothing approaches, local image statistics, and approaches based on morphological filtering [KCWI12].

DOI:10.24132/CSRN.2021.3101.9

ISBN 978-80-86943-34-3

Learning-based approaches or machine learning approaches in image stylization rely on painting examples as training input. Notable examples in recent years are deep learning style transfer techniques [GEB16, MOT15]. Network architectures that deal with image-to-image translation, paired [IZZE17] or unpaired [ZPIE17], are shown to work on different applications, including style transfer. Other loss terms aside from L1 or L2 norms are employed to further improve the performance of previous style transfer approaches. For example, the perceptual loss function is applied during training, wherein the difference of feature maps, between the input image and the reference image are minimized [JAL16]. Feature maps are typically extracted from the convolutional layers of a pre-trained VGG-19 network [SZ14]. Another loss term was proposed in the works of Ulyanov, Vedaldi, and Lempitsky (2017)[UVL17], that improves the diversity of image stylization through the use of Julesz ensemble [Jul81], which is based on a concept that textures are treated as a family of visual patterns that share local statistical regularities.

2.2 Stroke-Based Rendering

Vanderhaeghe and Collomose (2013) defined stroke-based rendering (SBR) as a process of synthesizing artwork by combining rendering marks (such as lines, brush strokes, or even larger primitives such as tiles) to a digital canvas [VC13]. The work of Litwinowicz (1997) [Lit97] is among the earliest methods for stroke-based rendering where impressionistic paintings are simulated on video input. Using multiple frames, the brush stroke and intensity are determined based on the movement of pixels identified using optical flow [TZ99]. The video abstraction method proposed by Kyprianidis, Kang, and Döllner (2009) [KKD09a] were inspired from the Kuwahara filter [KHEK76], which is a non-linear smoothing filter typically used for image processing that applies a blurring effect while preserving the edges. The modified filter generates a painting-like effect while preserving shape boundaries which is observed to work well in cartoon illustrations and oil paintings. More recently, deep learning generative models were utilized for SBR, such as a neural model capable of simulating learned hand strokes (NeuralPainter) [Nak19], and a neural painting environment where a position and a brush encoder are trained for learning different brush strokes for simulating drawings (StrokeNet) [ZJH18].

2.3 Augmented Reality Art

AR commonly applied in art galleries, exhibits, or cultural heritage, provides features for augmenting artificial objects that can be used to enhance various art forms [Mar18, Ger18]. Sketch-based and virtual sculpting applications were also observed to use AR [BC20]. Similarly, interactive modeling applications make use of AR, such as MagicToon, which is an interactive modeling tool for mobile AR that allows a user to augment 3D cartoon scenes based on hand-drawn sketches on paper [FYX17].

3 MONET'S IMPRESSIONISTIC ART STYLE: OVERVIEW

Impressionism is an art movement that started in the late 19th century where paintings are characterized as sketch-like, while others praised it for its depiction of modern life [Sam04]. Notable impressionist painters were Claude Monet, Edgar Degas, Camille Pissarro, and Pierre-Auguste Renoir [Bre74]. An impressionistic painting captures the effects of sunlight, gray and dark tones produced as a product of mixing complementary colors. Pure impressionism is often characterized by the lack of black paint. Rather than neutral white, grays, and blacks, painters typically draw shadows and tones in color. Outdoor paintings are described as fresh, due to the use of bright and light brush colors [Sam04, FMSGA18]. In this study, we primarily focused on the works of Claude Monet, where his works accurately depict the descriptions described earlier.

To the best of our knowledge, there are limited studies that describe Claude Monet's work. A book was written by Forge (1995) [For95], and another book was written by Wildenstein (1978) [MW78] that described the life of Monet and his works in chronological order. Concepts discussed in the succeeding sections are supported by the book of Forge (1995) and serves as the main reference for designing our framework.

3.1 Color Palettes

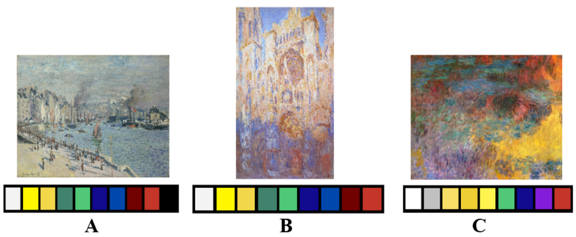


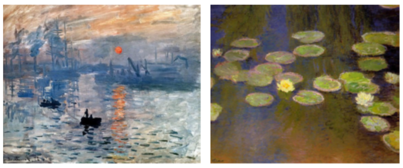
Figure 3: Three color palettes identified in the works of Monet, and corresponding paintings that contain a specified color palette. A: Pre-1886 (Port of Le Havre). B: Post-1886 (Rouen Cathedral: The Portal). C: Final years (Water Lily Pond, Evening).

Claude Monet used three color palettes in his paintings. The first color palette was used in his pre-1886 works, which consists of lead white, chrome yellow, cadmium yellow, viridian green, emerald green, french ultramarine, cobalt blue, alizarin crimson, vermilion, and ivory black. After the year 1886, Claude Monet refrained from using ivory black. Few years before his death, Claude Monet used another color palette, that consists of the following colors: silver, white, cadmium yellow light, cadmium yellow dark, and lemon yellow, emerald green, ultramarine extra-fine, cobalt violet light, and vermilion. We classify these as follows: *pre-1886*, *post-1886* and *final years (FY)*, which refers to the last color palette a few years before Monet's death. We illustrate these color palettes and sample paintings in Figure 3.

3.2 Brush Stroke Techniques

Claude Monet uses fast brush strokes to portray light, a technique honed throughout his career in painting [For95]. A primary example depicting this behavior can be observed in his "Sunrise" painting (Figure 4). Monet uses his signature short, choppy strokes which coaxes the viewer to "optically blend" the strokes and values when painting distant objects close to the horizon [Dun76]. Lastly, fast and broken brush strokes with little to no smoothing are also applied, where viewers see the traces of the brush strokes that sometimes give the painting an unfinished appearance. Impressionist painters like Monet typically paint in one sitting, giving their work a spontaneous feel.

Claude Monet's approach to lighting depends on the brushstroke technique. For example, Monet uses brushstrokes that are soft and diffused while using impasto,for soft, light, and outdoor scenes, which is evident in the "Water Lilies" painting (Figure 4). Impasto is a painting technique where thick layers of paint are laid on an area. The thick layers create a visual effect wherein the brush or painting knife strokes are very visible.



A **B**

Figure 4: Sample paintings of Monet showing his brushstroke techniques. A: In the Sunrise painting, the details of the water were created using rapid brush strokes where rough patches can be observed throughout. B: The Water Lilies painting illustrates soft and diffused brush strokes. It has an ample amount of indigo underneath each lily in addition to using the impasto technique.

4 ARCHITECTURAL FRAMEWORK

The system accepts an image captured from the mobile AR application after the user places virtual 3D models, which can be moved, rotated, and scaled accordingly to fit a given tabletop scene. The captured image undergoes three steps namely, color palette selection, color mixing, then brushstroke rendering. Figure 5 shows the architectural framework.

4.1 Color Palette Selection

The paintings of Monet were gathered and grouped accordingly to the year it was painted. There are 471 paintings in the pre-1886, 238 in the post-1886, and 149 in the final years. Given an input image captured from a given tabletop scene, the goal is to find the most suitable color palette identified by year. One method to do this is to find the nearest color composition from the cluster of paintings. Finding the nearest color composition among Monet's paintings and comparing it against the input image, by using RGB information, is not appropriate because real-world colors do not necessarily map to Monet's limited color palette. Using image edges, extracted using Sobel operator [KVB88] are sensitive to gradient properties of the image. Furthermore, different brush strokes are used across all paintings which may affect edge detection methods.

We find that using GIST descriptors [OT01] produces the most accurate classification results when selecting a color palette as it appears to be robust against changes in camera view and lighting which may vary in a tabletop scene, especially when captured indoors. GIST descriptors are extracted from the input image which is then compared to the GIST descriptors of Monet's paintings. Since GIST descriptors contain gradient information from varying scales and orientations which are used to recognize various scenes, scenes in Monet's paintings that are similar in composition with the input image will most likely be selected. To perform the color palette selection, using the norm values of GIST descriptors, a k-nearest neighbor algorithm is applied to determine the closest set of paintings for the input image using Euclidean distance ($k = 10$). The class (pre-1886, post-1886, final years) is identified by majority voting and the color palette from the class identified will be selected.

4.2 Color Mixing

With the selected palette, each image pixel's color is changed. Using hue values, the two closest colors in the palette, in terms of Euclidean distance are selected. We observe that performing blending of these two colors produces the best results.

4.3 Brush Stroke Rendering

For rendering brush strokes, a stroke rendering module is implemented where a modified implementation of Capeto's Painter software is utilized [Cap18]. The base implementation of Capeto (2018) [Cap18] was inspired from the painterly algorithm proposed by Hertzmann (1998) [Her98], combined with the brush stroke algorithm devised by Shiraishi and Yamaguchi (2000) [SY00]. To simulate Monet's brush stroke styles, three brush stroke patterns are used by default. The color-changed image from the color mixing step undergoes

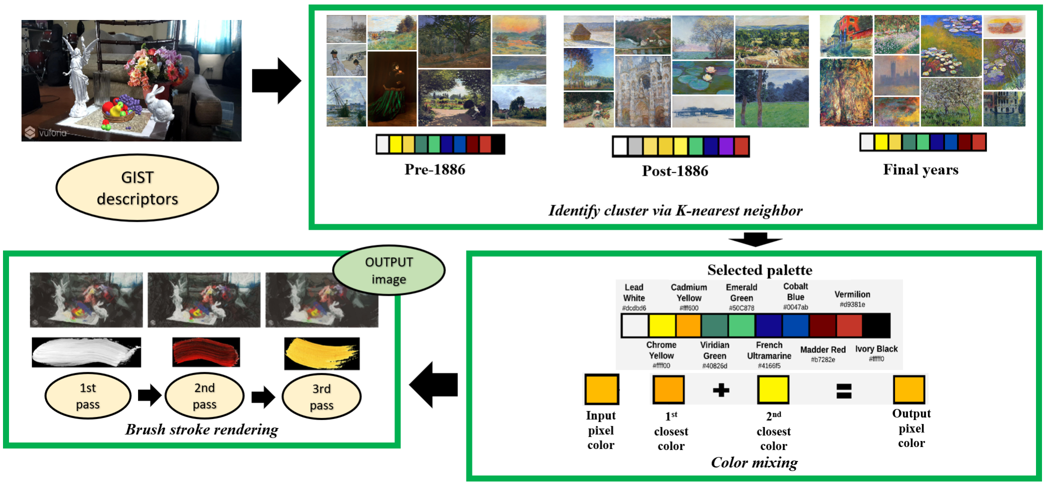


Figure 5: The architectural framework for simulating Claude Monet's impressionistic art style using images captured from a mobile device.

three passes in our stroke rendering module, selecting the corresponding brush stroke pattern. For each pass, a balanced alpha blending is performed as a method for combining the new and previously rendered image.

Curved brush strokes are sometimes not present in the paintings of Monet [For95]. As a refinement step, after the third pass, we perturb the brushstroke pattern by occasionally making the brush strokes straight, using the brush stroke algorithm proposed by Shiraishi and Yamaguchi (2000) [SY00]. Based on our initial experiments, setting a 50% probability of using straight brush strokes produce the best results.

No further major changes were created in the brush stroke implementations mentioned. We recommend the reader to refer to the papers of Hertzmann (1998), Shiraishi and Yamaguchi (2000), and the software designed by Capeto (2018) [Her98, SY00, Cap18].

4.4 AR Application Walkthrough

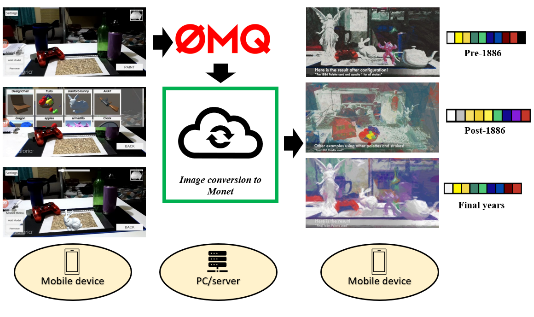
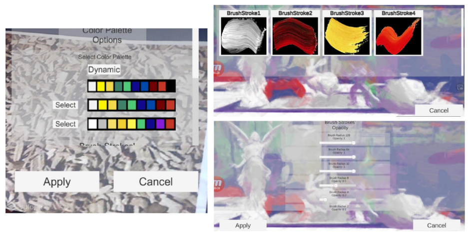


Figure 6: General walkthrough of the AR application. The user assembles their tabletop scene. Multiple 3D models can be placed, rotated, and scaled according to their preference. The image is captured and sent over to a server for processing via ZeroMQ. The rendered image using Monet's art style is sent back to the mobile device for viewing.



A B

Figure 7: Configurable parameters in our AR application. A: The user can decide whether to use a certain palette or allow the system to dynamically choose a palette based on scene context. B: Additional brush stroke patterns can be imported as a texture file, as well as modify the various parameters for brush strokes, such as opacity, size, and radius.

Figure 6 illustrates a walkthrough of the AR application. The user assembles their tabletop scene where multiple 3D models are placed by tapping on-screen their desired location in the virtual space. The models can then be rotated and scaled through a drag-and-drop interface. The composited scene is captured, and the image is sent to a server for processing via ZeroMQ, using the framework shown in Figure 5. ZeroMQ is a networking library that can transport messages and files between systems through inter-process communication [Hin13]. Offloading the rendering pipeline to a server is preferable as this saves computation time on the mobile device. Based on our initial system designs, we found that it takes approximately 4 to 5 minutes processing time for a 1024×768 image if the rendering step is performed on a mobile device. Further increasing the image resolution causes out-of-memory issues on low-end devices. Offloading the rendering takes approximately 2 minutes for the output image (1024×768 size) to be displayed on a mobile device, subject to network latency.

4.5 Other Technical Details

The AR application is implemented using Unity Engine and Vuforia [LB17]. Image results reported in this paper were retrieved from the actual AR application running on a 4GB mobile device. 3D models are preloaded in the application. Public 3D models such as Stanford Bunny, Dragon, Armadillo, and others are available by default. Our proposed framework is fairly algorithmic and contains numerous hyperparameters (e.g. brush stroke size, orientation, frequency, color values, etc) that directly affect the results, in terms of the images produced. Furthermore, the visual perception and evaluation of the final rendered image are very much subjective to the viewer. Basing on this rationale, we give full control to the user to configure the hyperparameters in real-time (Figure 7). Additionally, the user can import their brushstroke pattern as a texture file to replace one of the rendering passes.

5 RESULTS AND DISCUSSION

We performed a quantitative evaluation of images produced by our framework. Due to the very subjective nature of evaluating images that follow a certain art style, and since there is no universal methodology agreed upon, we solely relied on classical image metrics such as Peak-Signal-to-Noise ratio (PSNR) and Structural Image Similarity (SSIM). The evaluation is performed as follows: Since PSNR and SSIM require paired data, each test image rendered by our system is compared to every painting by Monet, and the corresponding PSNR and SSIM are measured. Hence, mean values are provided for each test image in our discussions and compared with other existing approaches. We refer to these measurements as **MPSNR** and **MSSIM** to denote the mean PSNR and mean SSIM respectively. A total of 859 artworks, combining paintings from pre-1886, post-1886, and final years from Monet, were used for this approach. To speed up the measurement time, we resized the test images and Monet paintings to 256×256.

We compared our method to CycleGAN [ZPIE17], as it is one of the state-of-the-art methods in image-to-image translation and style transfer, and to the style-aware network model developed by Sanakoyeu et. al (2018) [SKLO18] which is trained to cover different art styles. We refer to their model as AdaptiveStyleNet. Both works provide pre-trained models for converting images to Monet paintings. To validate the effectiveness of our color palette selection scheme and brushstroke rendering, we devised an alternative framework that uses only the pre-1886 color palette that we compare against. The brushstroke rendering is replaced with an Anisotropic Kuhawara filter by Kyprianidis, Kang, and Dollner (2009) using default parameters specified in their paper [KKD09b]. We refer to this alternative framework as the baseline.

5.1 Analysis of Results: Tabletop Scenes

A total of 9 test images, consisting of indoor tabletop scenes with varying object compositions, were gathered. To maintain consistency and possibly minimize variance in the results, we limited our 3D models placed in the scene to only Stanford Bunny, Lucy, and Stanford Dragon. The best results are shown in Figure 8. Table 1 summarizes the results.

In terms of MPSNR, the difference among the methods are negligible. In terms of MSSIM, we see bigger differences with our method as compared to CycleGAN, AdaptiveStyleNet, and the baseline. Judging the images visually, our proposed method generates images where the brush strokes can be seen better as compared to CycleGAN and the baseline which positively affected the results in terms of image metrics. In AdaptiveStyleNet, the brush strokes are denser and the strokes drastically changed the edges and fine details of the image. Another interesting observation is that virtual 3D models harmoniously blend with the other foreground objects in the final rendered image, which is not evident in both the baseline and CycleGAN, where the virtual 3D models from these methods stand out in the images. To some extent, we observe that AdaptiveStyleNet exhibits this same property.

5.2 Analysis of Results: Still Life and Outdoor Sceneries

To further validate the performance of our method in terms of closeness to Monet's impressionistic art style, we gathered 11 different images available on the web, most of which are still life compositions and outdoor scenes with vibrant colors. Since most of Monet's paintings were typically still life and outdoor sceneries, we wanted to observe if our proposed method is comparable if we use these compositions as well. The best results are shown in Figure 9. Table 2 summarizes the results.

Based on the MPSNR and MSSIM, our method achieved slightly higher scores than CycleGAN. We speculate that since CycleGAN was trained on diverse real-world image samples, the model can transfer Monet's art style properly on both still life and outdoor scenes. Our method substantially performed better as compared to the baseline and AdaptiveStyleNet, which further proves that the color palette selection scheme and three-pass brushstroke rendering are necessary.

In summary, our method works well when the input images contain one or more virtual 3D models that are placed using AR. However, further study is needed on why our method produces favorable results when virtual 3D models are present.

5.3 Mean Perception Error

Aside from analyzing the performance of our method through MPSNR and MSSIM, we explored how fea-

Table 1: Results of tabletop images in terms of MPSNR and MSSIM. Best values in bold.

Image Number: Tabletop Scenes	MPSNR Baseline	MPSNR CycleGAN	MPSNR AdaptiveStyleNet	MPSNR Ours	MSSIM Baseline	MSSIM CycleGAN	MSSIM AdaptiveStyleNet	MSSIM Ours
1	8.9262	9.4079	8.5507	**10.1829**	0.2131	**0.2737**	0.1943	0.2268
2	9.0150	**10.6156**	9.8140	9.6599	0.3253	0.3167	0.2120	**0.3466**
3	8.0774	**9.4321**	8.8698	9.2774	0.3026	0.2749	0.2145	**0.3401**
4	**10.0533**	8.9228	8.2585	9.0555	0.1689	0.2604	0.2123	**0.2872**
5	**9.0216**	8.6646	7.7644	8.7234	0.2384	0.2786	0.2120	**0.3191**
6	9.0979	**9.9304**	9.6984	9.8368	0.3501	0.3448	0.2207	**0.3685**
7	9.0687	**10.9157**	9.9291	9.4783	0.3150	0.3234	0.2161	**0.3332**
8	10.4475	10.9830	11.3076	**12.1283**	0.2243	0.2381	0.1956	**0.2978**
9	9.1286	8.6589	8.5850	**11.0010**	0.2641	0.2767	0.2247	**0.3247**
Average	9.2040	9.7257	9.1975	**9.9271**	0.2669	0.2875	0.2114	**0.3160**

Figure 8: Comparison of images produced by different methods. A: Input. B: Baseline. C: CycleGAN [ZPIE17]. D. AdaptiveStyleNet [SKLO18]. E: Ours.

ture representations in a convolutional neural network (CNN) behave when we use our rendered images as input. Specifically, we utilized the perceptual loss function, which is a measurement between image feature representations extracted from pre-trained CNN [JAL16]. The methodology for measuring MPSNR and MSSIM is similar wherein this time, the difference between feature representations of our images and feature representations of paintings by Monet is measured. If the value is close to zero, it implies that two images have almost identical feature representations. We used VGG-16 as our pre-trained network [SZ14] for extracting the feature maps for this metric, which we refer to as the mean perception error. Table 3 shows the results.

For tabletop scenes, our method consistently achieved the lowest mean perception errors on 7 out of 9 images, which plausibly means that our rendered images have close representations with Monet's paintings. For still life and outdoor images, our method only outperformed the other methods on four images, which further proves that our method generally works best for tabletop scenes with one or more 3D virtual models.

6 CONCLUSION

In this paper, we present an augmented reality application that simulates Claude Monet's Impressionistic art style, by using a multi-color palette selection scheme and brushstroke rendering with three passes. The user can place one or more virtual 3D models on an existing tabletop scene and render the final image. Results show that our method performs generally well on tabletop scenes and comparable to CycleGAN's performance while outperforming AdaptiveStyleNet. Future studies involve studying the user interaction aspect of our AR application and how it's effective in terms of showcasing an impressionistic art style. Other effective methods for clustering Monet's paintings can be explored, such as grouping the paintings by sceneries or color composition and further improving the multi-color palette selection scheme. One possible approach for further improving the performance is to explore how 3D models can be directly altered. Instead of using an

DOI:10.24132/CSRN.2021.3101.9

ISBN 978-80-86943-34-3

Table 2: MPSNR and MSSIM results of images with still life compositions or outdoor scenes. Best values in bold.

	MPSNR				MSSIM			
	Baseline	CycleGAN	AdaptiveStyleNet	Ours	Baseline	CycleGAN	AdaptiveStyleNet	Ours
Still Life Images								
Sunflowers	8.5312	**9.6476**	7.9276	8.5401	0.1586	**0.2073**	0.1673	0.1850
Violet flowers	7.0356	**10.2705**	8.5782	8.1070	0.3118	0.2888	0.2858	**0.3205**
Skull and lemonade	9.6906	10.1778	**10.8628**	9.7463	0.3450	0.3454	0.2542	**0.3586**
Flowers in metal vase, bags and kiwis	9.9569	9.7311	10.1624	**10.3814**	0.3579	**0.3599**	0.2889	0.3522
White flowers with a pples	10.3292	10.2516	10.9174	**11.4034**	0.3321	0.3413	0.2477	**0.3505**
Sunflowers with fruits	8.2634	8.4575	9.7994	**11.6888**	0.2368	0.2419	0.1718	**0.2536**
Dining table with fruits	10.1051	10.2556	**10.7580**	10.4910	0.3093	0.3054	0.2495	**0.3345**
Garden entrance	**11.3064**	9.4702	9.2243	10.1062	0.1887	**0.2521**	0.1741	0.2202
Women in the garden	10.9794	10.8531	10.4545	**11.6861**	0.1685	0.2334	0.1949	**0.2557**
Woman in kimono 1	8.6409	**11.1019**	8.3935	9.0217	0.2995	0.2838	0.2382	**0.3122**
Woman in kimono 2	8.1085	**9.8298**	8.6941	8.9167	0.2164	0.2351	0.1774	**0.2522**
Average	9.3588	10.0042	9.6156	**10.0081**	0.2659	0.2813	0.2227	**0.2905**

Figure 9: Comparison of images produced by different methods. The input images are still life compositions and outdoor scenes with vibrant colors. A: Input. B: Baseline. C: CycleGAN [ZPIE17]. D. AdaptiveStyleNet [SKLO18]. E: Ours.

image that captures the whole tabletop scene, using programmable shaders or manipulating materials to make them look impressionistic are worth pursuing.

7 ACKNOWLEDGMENTS

The authors would like to acknowledge De La Salle University, DLSU Science Foundation, and Department of Science and Technology (DOST) for funding this research. Deep learning experiments were performed through the DOST-ASTI COARE facility.

8 REFERENCES

[BC20] Sukanya Bhattacharjee and Parag Chaudhuri. A survey on sketch based content creation: from the desktop to virtual and augmented reality. In *Computer Graphics Forum*, volume 39, pages 757–780. Wiley Online Library, 2020.

[BCL15] Mark Billinghurst, Adrian Clark, and Gun Lee. A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction*, 8(2-3):73–272, 2015.

[Bre74] Richard Brettell. The'first'exhibition of impressionist painters. *The New Painting: Impressionism*, 1886:189–202, 1874.

[Cap18] Ugo Capeto. Non photorealistic rendering software - the painter: http://3dstereophoto.blogspot.com/p/painting-software.html. 2018.

Table 3: Results of images in terms of Mean Perception Error. Best values in bold.

Image Number: Tabletop Scenes	**Baseline**	**CycleGAN**	**AdaptiveStyleNet**	**Ours**
1	4.1037	3.8229	4.1341	**3.6987**
2	3.5092	3.4368	3.8146	**3.3100**
3	3.6896	3.7835	3.9466	**3.2802**
4	3.9159	3.7494	4.0065	**3.4909**
5	3.8295	3.7966	4.0794	**3.3856**
6	3.5546	3.5202	3.8391	**3.3368**
7	3.5517	**3.3281**	3.7632	3.3641
8	3.9241	**3.4735**	3.6831	3.6214
9	3.6306	3.6265	3.8762	**3.2814**
Average	3.7454	3.6153	3.9047	**3.4188**

Still Life Images	**Baseline**	**CycleGAN**	**AdaptiveStyleNet**	**Ours**
Sunflowers	3.9625	**3.6334**	3.8588	3.6404
Violet flowers	3.2255	3.2992	3.3895	**3.1899**
Skull and lemonade	**3.0381**	3.1280	3.3324	3.0637
Flowers in metal vase, bags and kiwis	**3.0093**	3.1236	3.1688	3.0714
White flowers with a pples	3.2119	**3.0695**	3.2664	3.2040
Sunflowers with fruits	3.6379	**3.3804**	3.7759	3.5223
Dining table with fruits	3.2522	3.2444	3.2661	**3.1897**
Garden entrance	3.4800	**3.4130**	3.8250	3.5501
Women in the garden	3.6298	3.5177	3.5959	**3.3011**
Woman in kimono 1	3.3608	3.3371	3.7211	**3.2904**
Woman in kimono 2	4.0126	**3.5503**	3.8819	3.6180
Average	3.4382	3.3360	3.5529	**3.3310**

[CH03] John P Collomosse and Peter M Hall. Cubist style rendering from photographs. *IEEE Transactions on Visualization and Computer Graphics*, 9(4):443–453, 2003.

[CRH05] John P Collomosse, David Rowntree, and Peter M Hall. Stroke surfaces: Temporally coherent artistic animations from video. *IEEE transactions on visualization and computer graphics*, 11(5):540–549, 2005.

[CS07] Chi Chu and Zen-Chung Shih. Painterly rendering framework from composition. *Journal of WSCG*, 15:151–158, 2007.

[Dun76] Bernard Dunstan. Painting methods of the impressionists. 1976.

[FMSGA18] Zoe Falomir, Lledó Museros, Ismael Sanz, and Luis Gonzalez-Abril. Categorizing paintings in art styles based on qualitative color descriptors, quantitative global features and machine learning (qart-learn). *Expert Systems with Applications*, 97:83–94, 2018.

[For95] Andrew Forge. *Monet*. Art Institute of Chicago Chicago, 1995.

[FYX17] L. Feng, X. Yang, and S. Xiao. Magictoon: A 2d-to-3d creative cartoon modeling system with mobile ar. In *2017 IEEE Virtual Reality (VR)*, pages 195–204, 2017.

[GEB16] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[Ger18] Vladimir Geroimenko. *Augmented Reality Painting and Sculpture: From Experimental Artworks to Art for Sale*, pages 211–225. Springer International Publishing, Cham, 2018.

[Hae90] Paul Haeberli. Paint by numbers: Abstract image representations. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 207–214, 1990.

[HE04] James Hays and Irfan Essa. Image and video based painterly animation. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, pages 113–120, 2004.

[Her98] 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.

[Hin13] Pieter Hintjens. *ZeroMQ: messaging for many applications.* " O'Reilly Media, Inc.", 2013.

[IZZE17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[JAL16] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

[Jul81] Bela Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, 1981.

[KCWI12] Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, and Tobias Isenberg. State of the" art": A taxonomy of artistic stylization techniques for images and video. *IEEE transactions on visualization and computer graphics*, 19(5):866–885, 2012.

[KD08] Jan Eric Kyprianidis and Jürgen Döllner. Image abstraction by structure adaptive filtering. In *TPCG*, pages 51–58, 2008.

[KHEK76] M. Kuwahara, K. Hachimura, S. Eiho, and M. Kinoshita. *Processing of RI-Angiocardiographic Images*, pages 187–202. Springer US, Boston, MA, 1976.

[KKD09a] Jan Eric Kyprianidis, Henry Kang, and Jürgen Döllner. Image and video abstraction by anisotropic kuwahara filtering. *Computer Graphics Forum*, 28(7):1955–1963, 2009.

[KKD09b] Jan Eric Kyprianidis, Henry Kang, and Jürgen Döllner. Image and video abstraction by anisotropic kuwahara filtering. In *Computer Graphics Forum*, volume 28, pages 1955–1963. Wiley Online Library, 2009.

[KS04] Levente Kovács and Tamás Szirányi. Painterly rendering controlled by multiscale image features. In *Proceedings of the 20th Spring Conference on Computer Graphics*, pages 177–184, 2004.

[KS06] Levente Kovács and Tamás Szirányi. 2d multilayer painterly rendering with automatic focus extraction. *WSCG: The 14-th international Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2006*, pages 141–146, 2006.

[KVB88] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988.

[LB17] Jonathan Linowes and Krystian Babilinski. *Augmented reality for developers: Build practical augmented reality applications with unity, ARCore, ARKit, and Vuforia*. Packt Publishing Ltd, 2017.

[Lit97] Peter Litwinowicz. Processing images and video for an impressionist effect. SIGGRAPH ’97, pages 407–414, 1997.

[Mar18] Todd Margolis. *Immersive Art in Augmented Reality*, pages 183–193. Springer International Publishing, Cham, 2018.

[MOT15] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. 2015.

[Mou03] David Mould. A stained glass image filter. In *Proceedings of the 14th Eurographics workshop on Rendering*, pages 20–25, 2003.

[MW78] Claude Monet and Daniel Wildenstein. *Monet’s Years at Giverny: Beyond Impressionism*. Metropolitan museum of art, 1978.

[Nak19] Reiichiro Nakano. Neural painters: A learned differentiable constraint for generating brushstroke paintings. *CoRR*, abs/1904.08410, 2019.

[OT01] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[RAGS01] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[Sam04] Margaret Samu. Impressionism: Art and modernity. *Heilbrunn Timeline of Art History*, 2004.

[SKLO18] Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Bjorn Ommer. A style-aware content loss for real-time hd style transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 698–714, 2018.

[SLKD16] Amir Semmo, Daniel Limberger, Jan Eric Kyprianidis, and Jürgen Döllner. Image stylization by interactive oil paint filtering. *Computers & Graphics*, 55:157–171, 2016.

[STD+16] Amir Semmo, Matthias Trapp, Tobias Dürschmid, Jürgen Döllner, and Sebastian Pasewaldt. Interactive multi-scale oil paint filtering on mobile devices. In *ACM SIGGRAPH 2016 Posters*, pages 1–2. 2016.

[SY00] Michio Shiraishi and Yasushi Yamaguchi. An algorithm for automatic painterly rendering based on local source image approximation. In *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 53–58, 2000.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[TZ99] Philip HS Torr and Andrew Zisserman. Feature based methods for structure and motion estimation. In *International workshop on vision algorithms*, pages 278–294. Springer, 1999.

[UVL17] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6924–6932, 2017.

[VC13] David Vanderhaeghe and John Collomosse. Stroke based painterly rendering. In *Image and Video-Based Artistic Stylisation*, pages 3–21. Springer, 2013.

[WXSC04] Jue Wang, Yingqing Xu, Heung-Yeung Shum, and Michael F Cohen. Video tooning. In *ACM SIGGRAPH 2004 Papers*, pages 574–583. 2004.

[ZJH18] Ningyuan Zheng, Yifan Jiang, and Dingjiang Huang. Strokenet: A neural painting environment. In *International Conference on Learning Representations*, 2018.

[ZPIE17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.