



HAL
open science

Interactive Painterly Rendering for Mobile Devices

Dongwann Kang, Kyunghyun Yoon

► **To cite this version:**

Dongwann Kang, Kyunghyun Yoon. Interactive Painterly Rendering for Mobile Devices. 14th International Conference on Entertainment Computing (ICEC), Sep 2015, Trondheim, Norway. pp.445-450, 10.1007/978-3-319-24589-8_38 . hal-01758450

HAL Id: hal-01758450

<https://inria.hal.science/hal-01758450>

Submitted on 4 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Interactive Painterly Rendering for Mobile Devices

Dongwann Kang and Kyunghyun Yoon

School of Computer Science & Engineering,
Chung-Ang University, Seoul, Korea
{dongwann, khyoon}@cg1lab.cau.ac.kr

Abstract. Painterly rendering is among the most popular non-photorealistic rendering techniques and has been employed in many applications. Research on painterly rendering mainly focuses on automatically generating artistic results. In this study, we aim to develop an entertaining and interactive application for painterly rendering with touchscreen mobile devices. The proposed application provides user interaction for added enjoyment and ensures high-quality painterly results. We provide a method for finding the appropriate position of the brush stroke around the points touched by the user and for generating and rendering the brush stroke. With the proposed method, users can quickly generate high-quality painterly results.

Keywords: Non-photorealistic rendering, Painterly rendering, Stylization

1 Introduction

Non-photorealistic rendering (NPR) [5, 11, 14] is an area of research in the field of computer graphics concerned with imitating the rules and practices of aesthetics and fine arts. In this field, research has attempted to simulate physical phenomena through the medium of the painting process and to mimic artwork styles. As a result of this research, a number of stylization methods are available that can plausibly imitate artistic styles. Moreover, with the increased use of mobile devices such as smartphones and tablet PCs, users can generate their own artwork using various mobile applications that employ these methods[2, 1].

One common approach to NPR requires users manual control of the finished product in order to imitate artwork styles. Because this approach merely simulates physical phenomena[4, 3], users must have a well-developed artistic sense or skill for drawing (or painting) akin to artists. However, it is difficult for most users to learn such artistic skills and develop an aesthetic sense.

Another NPR approach is to facilitate the style imitated by generating artistic results without the need for user expertise[7, 6]. Most methods that adopt this approach automatically draw according to a particular painting style. Consequently, untrained users can obtain desirable results. However, users are almost completely uninvolved in the drawing process with this approach.

In order to design a popular NPR application, the latter approach is more suitable, insofar as a wider range of users will be able to utilize it. However, if users can only obtain results that are automatically generated, they will quickly lose interest in the application owing to limited user control. Therefore, an NPR approach in which users are involved in drawing process is needed, and this can be accomplished by providing the appropriate level of interaction without requiring pretrained skills.

In this paper, we propose an interactive algorithm for painterly rendering that runs on mobile devices. The proposed algorithm is based on previous methods[12], and it allows users to generate plausible painterly artwork easily. In Section 2, we describe our enhanced algorithm for painterly rendering. We present the results from evaluating our method in Section 3. Finally, we conclude with a summary of our method and discuss the directions for future development in Section 4.

2 Proposed Algorithm

Our study is based on [12]'s algorithm for painterly rendering. This algorithm automatically generates painterly images using a brush-stroke texture database and a photomosaic method[13, 10, 9]. However, users cannot participate in the painting process with [12]'s method, apart from choosing the input image and a limited number of rendering settings. To enhance user participation, we propose an interactive algorithm for painterly rendering in order to facilitate the direct involvement of the user during the painting process.

2.1 Basic Painterly Rendering algorithm

This study follows the multiple-layered-grids approach proposed by [7]. First, we generate several grid layers. For each grid on each layer, a brush stroke is rendered. Smaller-sized grids correspond to layers at the upper levels. Consequently, only tiny grids are generated for the top layer.

According to the method in [7], a brush stroke is generated for each grid along the gradient of the input image. For our proposed algorithm, however, we employ [12]'s method to determine the most suitable brush stroke from a stroke-texture database and map it to the canvas. First, we sample the color of the input image at a point selected by the user through a touch interface in order to allow the user to choose the color of the brush stroke. Next, we select a brush-stroke texture from the database and map it to the canvas with the sampled color. Then, we resize the size of the texture to 1.5 times that of the grid in order to cover the area of the grid. Finally, we calculate the color distance between the grid area of the canvas and that of the input image. By repeating this process for all of the strokes in the database and selecting the one with the smallest distance, we can find the most suitable brush stroke. We then render this brush stroke to the canvas according to the texture-mapping process described above.

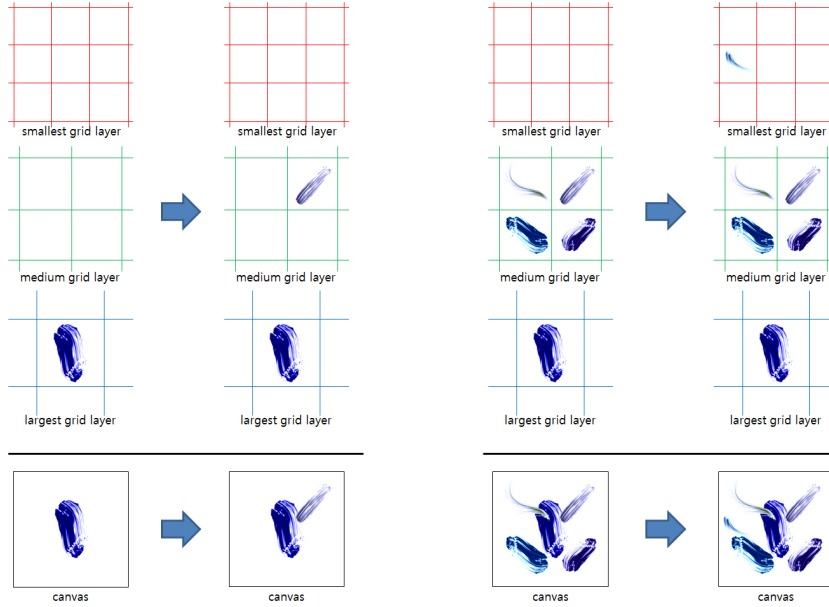


Fig. 1. The proposed interactive painting method. We select the largest grid layer among the grid candidates and paint a brush on it.

2.2 Interactive Painterly Rendering algorithm

By using the method described in the previous section, we can obtain painterly rendering that is similar to the results from previous works[7, 6, 12]. Rather than performing the process automatically, we offer an interactive approach to the rendering. With our method, users can select the position of each brush stroke. If users are required to select all of the properties for the brush, such as the color, position, size, and shape, the process would be overly tedious for most untrained users. To alleviate such complexity, the size, texture, and color of the brush stroke are adaptively determined according to the position of the stroke selected by the user.

With our method, users input a point on the canvas by using a touch interface. The application then generates a circle with a predefined radius around that point and locates grids whose central points fall within this circle. Then, we eliminate the grids within the brush strokes that have already been generated. The remaining grids are candidates for the brush stroke that will be generated. By following the multilayered approach for rendering brush strokes according to the size of the strokes, we select the largest grid among the candidates (Figure 1). If there are other grids that are equivalent in size to the largest grid, they are selected as well. In order to prevent the regularity of the brush location, we obtain the jittered point for the center of the selected grids and generate a brush stroke for that point using the method described in Section 2.1. When multiple

grids are selected, we choose the most suitable brush stroke for rendering (i.e., the stroke with the minimum color distance).

If there are no more candidates within the circle generated at the point touched by the user, we then obtain the jittered point from the user-selected point and generate a grid with its center at the jittered point. Then, a brush stroke is generated using the method described above. Thus, the size of the grid is equivalent to the size of the grids at the top layer.

By iteratively performing the above process until the canvas is appropriately covered with a number of brush strokes, painterly results are generated interactively.

3 Results

Our experiments were conducted on an iPad 3. Figure 2 shows a comparison of the results from an automatic rendering approach, a manual approach, and our interactive approach. Table 1 summarizes the elapsed time for generating the results shown in Figure 2. For the automatic approach, we employed the basic method described in Section 2.1 and applied it in the order of each layer. With the manual approach, a user selected the size and orientation of the brush for each brush-stroke texture and rendered the brush stroke on a user-selected point using the iPads touch interface. For a given image, the user paints brush strokes by touching the screen until he/she wants to finish painting. The experiments demonstrate that the automatic approach rendered the image more quickly to obtain a complete painterly result. The quality of the result appeared to be satisfactory. However, the user does not participate in the rendering process with the automatic approach, apart from selecting a few parameters such as the grid size and the number of layers. In contrast, the manual approach requires a considerable amount of time to complete the painting. Moreover, the quality of the result appeared to be the worst, probably because the user was unfamiliar with the interface and the painting method. Our interactive approach offered a balance between these approaches in terms of the elapsed time. Moreover, it is evident in Figure 2 that the user was able to focus on several interesting points. Because the proposed interactive approach allowed the user to be involved in the painting process, the users preferences and intent were appropriately reflected in the result. Figure 3 shows various results from the proposed interactive method by untrained users.

4 Conclusion and Future Work

In this paper, we proposed an interactive method for painterly rendering with an enhanced mobile application based on previous methods[12]. To facilitate user interaction, we developed a method that finds the appropriate position of the brush stroke around a point selected by the user, after which the brush stroke is generated, and the image is rendered. With our method, users can generate high-quality painterly results quickly and easily.

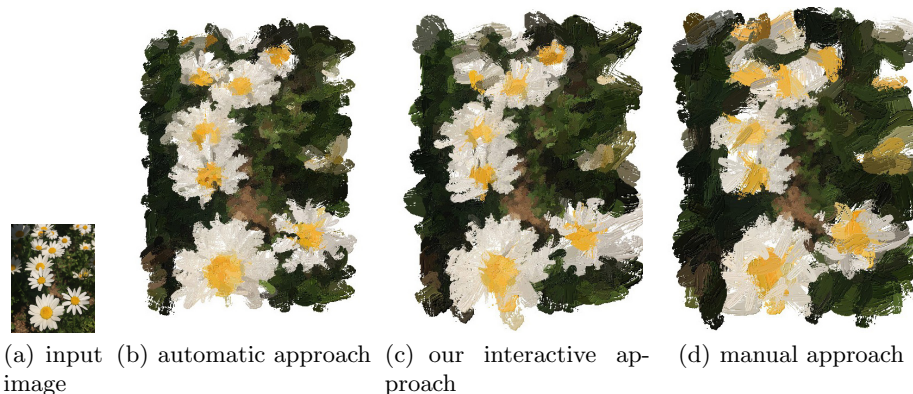


Fig. 2. A comparison of the results from several approaches. These results are generated by user1 in Table 1.

Table 1. The elapsed time for generating the results shown in Figure 2.

approach	elapsed time (s)		
	user1	user2	user3
automatic approach	51	50	55
our interactive approach	123	359	108
manual approach	549	786	451

In future research, we plan to implement various styles of stroke-based rendering[8], such as watercolor, pen and ink, and pastel. To do so, we must reinforce the database to accommodate these various strokes and modify our rendering method to apply them appropriately.

Acknowledgment

This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korea government(MEST) (NRF-2013R1A2A2A01069032).

References

1. Mobilemonet, <http://itunes.apple.com/us/app/mobilemonet/id395563116>, accessed 21 Sept 1998
2. Toonpaint, <http://toonpaint.toon-fx.com>, accessed 21 Sept 1998
3. Baxter, W., Wendt, J., Lin, M.C.: Impasto: A realistic, interactive model for paint. In: Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering. pp. 45–148. NPAR '04, ACM, New York, NY, USA (2004)
4. Curtis, C.J., Anderson, S.E., Seims, J.E., Fleischer, K.W., Salesin, D.H.: Computer-generated watercolor. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. pp. 421–430. SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (1997)



Fig. 3. Various results from the proposed interactive method.

5. Gooch, B., Gooch, A.: *Non-Photorealistic Rendering*. A. K. Peters, Ltd., Natick, MA, USA (2001)
6. Hays, J., Essa, I.: Image and video based painterly animation. In: *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering*. pp. 113–120. NPAR '04, ACM, New York, NY, USA (2004)
7. Hertzmann, A.: Painterly rendering with curved brush strokes of multiple sizes. In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. pp. 453–460. SIGGRAPH '98, ACM, New York, NY, USA (1998)
8. Hertzmann, A.: Tutorial: A survey of stroke-based rendering. *IEEE Comput. Graph. Appl.* 23(4), 70–81 (Jul 2003)
9. Kang, D., Seo, S.H., Ryoo, S.T., Yoon, K.H.: A parallel framework for fast photomosaics. *IEICE Transactions on Information and Systems* 94-D(10), 2036–2042 (2011)
10. Kang, D., Seo, S., Ryoo, S., Yoon, K.: A study on stackable mosaic generation for mobile devices. *Multimedia Tools and Applications* 63(1), 145–159 (2013)
11. Kyprianidis, J.E., Collomosse, J., Wang, T., Isenberg, T.: 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 (May 2013)
12. Seo, S., Ryoo, S., Park, J.: Interactive painterly rendering with artistic error correction. *Multimedia Tools Appl.* 65(2), 221–237 (Jul 2013)
13. Silvers, R.: *Photomosaics*. Henry Holt and Co., Inc., New York, NY, USA (1997)
14. Strothotte, T., Schlechtweg, S.: *Non-photorealistic Computer Graphics: Modeling, Rendering, and Animation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2002)