Painterly Rendering Based on Piano Interface

Kyeong A Kwon and Hae Won Byun*

Dept. of Computer science, Sungshin Women's University {kyeonga, hyewon}@sungshin.ac.kr

Abstract

In this paper, we propose a new painterly rendering technique for visualizing sound, based on correlation between sight and auditory. We analyze the sound feature to create a painterly image which is dependent upon the input music that is playing by user using the piano interface. We analyze the tempo, the frequency, and the octave which are the basic characteristics of sound features. Sound features are used to determine strokes shapes, colors, and brightness which are the crucial elements to complete the pictorial rendering. In addition, during the rendering process, we use a multiple layer system to overlap the layers. From this study, we can create a non-photorealistic image which is suitable for the mood of the music that is playing by user.

Keywords: Non-Photorealistic Rendering, Painterly Rendering, Sound Feature, piano

1. Introduction

There have been plenty of researches tackling how to connect sight and sound. It is not a new study to connect sight and auditory in recent years, but there have been various attempting to connect sight and auditory by ancient philosophers, thinkers, and artists. With the development of digital technology, the sound visualization is expanding the area in the fields of media art, exhibitions, and performances. So it is easy to fine the sound visualization around us. For example, a musical fountain where the water moves by music and a media player which displays a wavelength based on the music. To visualize the sound, a musical fountain is focused on making the shape of the fountain by the rhythm and the pitch and a media player is focused on changing the shape of the wavelength.

In this paper, we use a painterly rendering technique to visualize the sound. To express the synesthesia that harmonizes with visual and auditory senses, we use sound features to determine the brush strokes. The input sound is the music that is played by a user using the piano interface. The tempo, the frequency, and the octaves which are the basic features used to categorize the music genre are used as sound features. Then each sound feature is used to select the shapes, the colors, and the brightness of the brush strokes. So we obtain the result which is suitable for the atmosphere of the input music.

^{*} Corresponding author: hyewon@sungshin.ac.kr

2. Related Work

2.1. Painterly Rendering

The ultimate objective of painterly rendering is to create non-photorealistic images by placing brush strokes according to some goals. Painterly rendering was developed in the early '90s by Paul Haeberli [1]. Haeberli proposed a painterly rendering by brushstrokes from user input. Litwinowicz [2] created the brush strokes by using direction and texture of the strokes. Later, the painterly rendering using the curved β spline strokes were proposed by Hertzmann [3]. He also presented a multi-layer system and used the different size of strokes in each layers. Hays [4] further arranged the brush strokes in motion layers. Hays used an interpolation using Radial Basic Function (RBFs) to make the direction of the strokes. Decarlo and Santella [5] provided an eyetracking-based user interface to allow for a user-guided specification of regional importance which enables adaptive control of the level of abstraction. Shugrina et al., [6] present an interactive painterly rendering whose appearance adapts in real time to reflect the perceived emotional state of the viewer. They analyze facial expressions of user through the detection of facial action units derived from the FACS scheme. Zeng et al., [7] presented a region-based painterly rendering algorithm that splits the scene into a tree of regions, each region being classified by its semantic content. This allowed the users to select different styles for human skin, buildings, vegetation or sky. Region based control over stroke orientation is also a feature of interactive painting environments such as AniPaint [8].

2.2. Audio Visualization

Attempting to connect sight and auditory has been studied by ancient philosophers, thinkers, and artists. Especially the ancient physicist Newton scientifically proved the similarity of sound and color as dividing the lights into seven colors and comparing each color to the wavelength. As a similar study, there are researches that attempt to express the sound in the color by one to one corresponding the sound and color [9, 10]. A painter Kandinsky [11] who studied in earnest about the correlation between the sound and the color reveals that each different colors have their own feelings as each different instrument has a unique tone. In addition, he said each instrument has different colors as trumpet has yellow color, viola-orange, cello-blue, bassoon-purple, organgreen, and tuba-red.

In this paper, we propose a new painterly rendering technique for visualizing sound and create the brush strokes using sound features from input music. Our goal is to create the suitable results for the mood of inputted music. Consequently we can express the synesthesia that is harmonized with visual and auditory.

3. System Overview

Figure 1 shows the configuration of our system. The proposed system can be roughly divided into the sound feature phase and the rendering process phase. First of all, the user plays the piano to make the input music using the piano interface. And then in the sound feature phase, we calculated the music tempo by using BPM (Beat Per Minute) and analyze the frequency to obtain the musical scales and the octaves of inputted music. The tempo, the frequency, and the octaves are used to determine strokes shape, color, and brightness in the rendering process phase. In the rendering process, first of

all we apply a Gaussian blur to reduce the noise in the input image and then determine the shape and the color of the strokes. After that, the strokes are rotated following the edge orientation by using ETF (Edge Tangent Flow) [12] and placed into a grid-based. Finally, using a multiple-layer system, we can obtain the final results.



Figure 1. Overview

4. Piano Interface

In this part, we make the piano interface to allow the user to play the piano using the computer keyboard. The music that is playing by user is used as an input sound for extracting the sound feature. Figure 2 shows the piano interface proposed in this paper. The piano is generally made up of 88 keys that have the white keys of 52 pieces and the black keys of 36 pieces. And the piano has a range of one to seven octaves. Occasionally the piano with the keys of more than 88 is used in concert, but 88-key piano is typically used.



Figure 2. Piano Interface

International Journal of Multimedia and Ubiquitous Engineering Vol.8, No.6, (2013)

In this paper, we have let out all sounds of the piano by matching the piano keys to keyboard keys. Pressing a key on the keyboard, the key gets the sound of the white keys by default and pressing a key with the shift key, it gets the sound of the black keys. We get the piano sound by using MIDI library. Piano interface is made to be able to play by simultaneously using the left and right hands like playing the real piano and can sound out the seven octaves. Figure 3 shows the correspondence between the keys of the piano, with piano interface in this system.



Figure 3. Piano Octaves

5. Sound Feature Extraction

5.1. Tempo extraction

To calculate the tempo from the input sound, we use Beats per minute (BPM). BPM is a unit typically used as a measure of tempo in music and heart rate. In musical terminology, tempo is the one of the most basic feature of the music configuration and it usually means fast of the music. However, a lexical meaning of the tempo is an atmosphere of the music. In general classic music has a slow tempo and rock music has a fast tempo. So tempo is a crucial element to categorize a music genre. And in this paper, tempo is used to determine the shape of the stroke.

5.2. Frequency and Octave analysis

We use Fast Fourier Transform (FFT) to obtain the frequency from the sound. FFT is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse. The frequency of the sound decides the pitch. High frequency makes a high pitch and low frequency makes a low pitch. Using FFT, we can extract the musical scales and the octaves. The musical scales are used to determine the colors of the strokes. And the brightness of the strokes is determined by extracted octaves. In music, an octave is the interval between one musical pitch and another with half or double its frequency. Figure 4 shows the octaves and the frequency of the A note corresponding to the octave. International Journal of Multimedia and Ubiquitous Engineering Vol.8, No.6, (2013)



Figure 4. Octaves and Frequency

6. Painterly Rendering

6.1. Stroke Shape

The shape and the color of the strokes are used as elements in painterly rendering. The shapes of the strokes are chosen by matching the extracted tempo from the source music. The atmosphere generated with a slow tempo matches the sound of its kind, giving it a simple and smooth stoke. Vice versa with music that has a fast and upbeat tempo the atmosphere rendered will be seemingly suitable for that particular type. We adopted some brush examples from the brush dictionary [7] (see Figure 5).





Figure 6. RGB color model and HSI color model

6.2. Stroke Color and Brightness

In this study, we use a HSI color space rather than a RGB color space; because in RGB color space there will be correlations between the different channels' regarding the color displacement, values (see Figure 6). The HSI color model represents every color accurately using three components. H stands for the Hue, S for the Saturation and I for the Intensity. H can be chosen by frequency of the sound. We converted an RGB color image to HSI color space by using Equations 1. The original colors of the brush

strokes are chosen from blurred source image and can be transferred by frequency of the input sound.

$$H = \cos^{-1} \left[\frac{\frac{1}{2} [(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right]$$

$$S = 1 - \frac{3}{(R + G + B)} [\min (R, G, B)]$$
(1)

$$I = \frac{(R + G + B)}{3}$$

(If B > G, H = 360° - H)

In this paper, we adopt the sound to color conversion to determine the color of strokes [8]. The sound to color conversion is a conversion method to convert sound into color, based on the likelihood in the physical frequency information between sound and color. Human can recognize the various colors according to the energy distribution of light 390THz (= $3.9x10^{12}Hz$)~750THz(= $7.5x10^{14}Hz$). Sound manifested the frequency (Hz) and the range of hearing which human can hear is approximately 20Hz~20kHz. Figure 7 shows the frequency mapping between audible and the visible frequency bands.



Figure 7. The frequency mapping between audible and visible frequency bands

We extract the sound feature for the frequency mapping between sound and color. The color and the sound physically share the attribution of a wave motion as sympathetic vibration, amplification, interference and offset. And a ratio of Do, Mi, Sol and a ratio of red, green, blue are conformable to each other. Using this principle, the audible frequency range is converted to visible frequency (see Figure 8).



Figure 8. The process of converting musical scale elements into color

The brightness of the strokes is corresponded to the octave of sound features. If the octave of input music is high, raise the brightness of the strokes and if the octave is low, lower the brightness of the strokes. So the atmosphere of the image is determined. Figure 9 shows the steps of brightness and the color changes of the brightness of the color (red).



Figure 9. The steps of the brightness

6.3. Stroke placement

We use edge tangent flow (ETF) to obtain the flow field and rotate the strokes which are guided by ETF. During the rendering process, we use a multiple layer system. We start painting with a large brush for the first layer and paint again over the foreground layer with a smaller brush to add detail.

7. Results

In this part, we discuss rendering results obtained from the source image in Figure 10 (a). Figure 10 (b) is the basic rendering result that has no input music. We used the default brush and color. It was taken from blurred source image. Figure 11 shows the results for the comparison of the stroke shape according to the music tempo. The input music was the same but had different tempo. (Figure 11 (a) – using simple and smooth stroke, (b) - using rough stroke)

Figure 12 shows the results for the change in brightness according to the octave. The same music was played but the octave was changed. Other rendering results are shown in figure 13. (Figure 13 (a) - Mary Had A Little Lamb, (b) - Volevo un gatto nero, (c) - My heart will go on)









Figure 11. Stroke shapes comparing



Figure 12. Brightness comparing



Figure 13. Other results

8. Conclusion and Future Work

We have presented an interactive system for painterly rendering that visualize the sound playing by user using piano interface. To choose the shapes, the colors, and the brightness of the strokes, sound features were extracted from the input music playing by user. Through the connection between sound features and stroke features, we obtain the results of the various styles.

Currently, the user can play only piano sound, but in the future, it will be possible to assemble by using various instruments. We only extract the tempo, the frequency, and the octaves as sound features from input sound, but the painterly rendering will be considered the meaning of the lyrics of the song to apply to the rendering process in the future. So that painterly rendering system will be able to create a result which is not only suitable for the mood of the song but also the meaning of the lyrics of the song. Our system is based on PC. However, in the future, if we extend the system for mobile phones, it can be more convenient for user.

Acknowledgements

This work was supported by the Sungshin Women's University Un-Jeong Global Project Research Grant of 2013.

References

- [1] P. Haeberli, "Paint by numbers: abstract image representations", In Proceedings of ACM SIG-GRAPH, vol. 4, (1990), pp. 207–214.
- [2] P. Litwinowicz, "Processing images and video for an impressionist effect", In Proceedings of ACM SIGGRAPH, (1997), pp. 407–414.
- [3] A. Hertzmann, "Painterly rendering with curved brush strokes of multiple sizes", In Proceed-ings of ACM SIGGRAPH, (**1998**), pp. 453–460.
- [4] J. Hays and I. Essa, "Image and video based painterly animation", In Proceedings of 3rd ACM Sympos. on NPAR, (2004), pp. 113–120.
- [5] D. Decarlo and A. Santella, "Abstracted painterly renderings using eye-tracking data", In Proceedings of ACM SIGGRAPH, (2002), pp. 769–776.
- [6] M. Shugrina, M. Betke and J. Collomosse, "Empathic painting: interactive stylization through observed emotional state", In Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering (NPAR 2006) (Annecy, France, 2006), ACM Press, (2006).

- [7] K. Zeng, M. Zhao, C. Xiong and S. -C. Zhu, "From image parsing to painterly rendering", ACM Transactions on Graphics (TOG), vol. 29 no. 1, (2009) December, pp. 1-11.
- [8] P. O'Donovan and A. Hertzmann, "AniPaint: interactive painterly animation from video", IEEE Trans. Vis. Comput. Graph., vol. 18, no. 3, (2012), pp. 475–487.
- [9] G. H. Kim and J. G. Baek, (Eds.), "Sound Color Harmonism", Impress Publishers, Seoul, (2003).
- [10] J. I. Kim and J. S. Jung, "A basic study on the system of convering color image into sound", In Proceedings of KIIS Spring Conference, vol. 20, no. 1, (2010), pp. 251n–256.
- [11] W. Kandinsky, Über das Geistige in der Kunst, (1911).
- [12] H. Kang, S. Lee and C. K. Chui, "Flow-based image abstraction", IEEE Transactions on Visualization and Computer Graphics, vol. 15, no. 1, (2009), pp. 62-76.

Authors



Kyeong A Kwon

She received the BS degree in computer science from Sungshin Women's University, Korea, in 2010. She is currently working toward the MS in computer science at the Sungshin Women's University. Her research interests include non-photorealistic rendering, computer graphics, and mobile application.



Hae Won Byun

She received the BS degree in computer science from Yonsei University, Korea, in 1990 and MS and PhD degrees in computer science from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1992 and 2004, respectively. She is currently a professor in the School of Information Technology at Sungshin Women's University, Seoul, Korea. She's primary research interests and facial motion capture and expression retargeting, performance based animation, nonphotorealistic rendering, and texture synthesis.