International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 13 (2015) pp 33611-33613
© Research India Publications. http://www.ripublication.com

A Modified Point Placement algorithm for Video Stippling

Seulgi Oh
School of Information Technology
Sungshin Women's University, Korea
osg571@naver.com
Haewon Byun*
School of Information Technology
Sungshin Women's University, Korea
hyewon@sungshin.ac.kr
Corresponding Author*

Abstract- This paper presents a modified point placement algorithm for video stippling, following the style of professional headcut illustrations. For the effective display of some moving object features, a new novel generation method of initial point set, point distribution, addition and deletion is proposed. The main idea is to improve the dots placement algorithm to calculate the dot position by considering brightness, edge and feature lines of an image. The contribution of our algorithm is the video stippling with enhanced object features and the automatic generation of various video stippling with input parameters. Experimental results show that our method leads to the generation of feature exaggerated stipple video for various type of input video including human and objects.

Keywords- Non-Photorealistic Rendering, Video Stippling, Initial Point Generation, Point Distribution, Point Addition, Point Deletion

1 Introduction

Stippling is a painting executed by means of dots or small spots with various shade and size effects. Typically, several tons of thousands of dots are manually arranged to generate a single drawing. For a long time, many researchers have focused on producing a stippled image. Recently, some researchers want to make stippled image sequences from an input video. The previous researches for video stippling have some limits that do not express the special features of moving objects in a video because the stippled video is displayed by controlling the dots intensity according to just image tone[1].

In the previous paper, we proposed a new video stippling method to enhance object features in a video[2]. We exaggerated the features of moving objects with dots placement algorithm that places a number of dots intensively in the region close to objects boundary in a video. In this paper, we improve the dots placement algorithm to calculate the dot position by considering brightness, edge and feature line of an image. By focusing on the style of headcut illustration like a Figure 1, we make a feature line image and then refine the dots position with the feature line image.

The other important thing in non-photorealistic video research is to keep time coherency between video frames. To keep the time coherency, we employ an image stippling algorithm for the first frame and then refine the dots position from the first frame for the second and following frames. The main contribution of our algorithm is that we express a stippled video with enhanced object features by exaggerating the features of each object in an image. The other contribution is the automatic generation of stippled video so as to produce various style of stippled video with an input of some parameters.

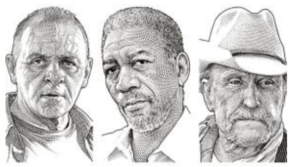


Fig 1. Headcut illustrations created by Randy Glass

2 Related Works

In a pointillism research of non-photorealistic rendering, many people have studied to express the artistic technique of a neo-impressionism artist, Seurat[3][4]. As well as reproducing the traditional pointillism, new pointillism technique using various styles of points are proposed recently. There are many image stippling methods. A semi-automatic image stippling system with Voronoi Diagram[5], an image stippling method reflecting tone of input image[6], edge feature enhanced image stippling with a graph[7], image stippling with directional feature[2][8], an artist's example based technique[9] are proposed. In contrast to these researches focusing on image stippling, video stippling researches are starting to be appeared recently[1][11].

3 Overall Process

Our system consists of three stages which are image preprocessing, point placement and stippled video rendering. In the image preprocessing stage, we extract image sequences from a video and transform the image into gray image and generate feature line image and frame difference image. In the point placement stage, we employ Roid method[1] that deals with image intensity and then adapt the method so as to handle feature lines of an image. In the last stage of the stippled video rendering, we produce series of stippled image and convert the image sequence into a video. The image preprocessing and the stippled video rendering method is same as to the previous Oh's research[2]. Our contribution is on the modified point placement algorithm for the feature enhanced stippled video.

4 A Point Placement Algorithm

This algorithm includes initial points set generation, point distribution, point deletion and addition. To distribute points addressing object features, we modify the Houit algorithm[1]. The original method focuses on making Centroid Voronoi Diagram according to image intensity. For the first frame, we modify the method to decide the image intensity by using image edge and intensity and set the position of circular dots by using feature lines of object. For the following frames, we produce the initial points set by deleting and adding some points to keep time coherency between a video frames.

4.1 Initial Point Set Generation

we use both a point set 'V' which is in the region of Voronoi Diagram and a point set 'S' which is site points for stippling rendering. We control the number of elements in the set S according to user input. We use the number of elements in the set V as the number of the elements of S multiplying 100. If the pixel value from the gray image is larger than a random number between 0 and 255, we add the point as an element into the set. That is, if the brightness of pixel is larger than the other pixels, the bright pixel has greater chance that it belongs to the set. In the process of initial point set generation, we reference the previous initial point set in order to prevent setting a point into the same position as to that of the previous point.

4.2 Point Distribution

The points are distributed evenly by rearranging the initial points according to feature lines. At the early stage, the initial points are distributed according to the image edge and intensity. The points are distributed affected by the white feature lines of feature line image produced in the previous stage.

4.3 Point Deletion

By considering all the points in both Site point set and Voronoi point set, we decide if we delete some points. After converting a brightness value into a range value from 0 to 100, we generate a random integer value in the range of 0 to 99 and compare the two values and then decide if we delete the point. It means that we want to delete if the area in the current frame is darker than the area in the previous frame or the area does not include edge of some objects any more.

4.4 Point Addition

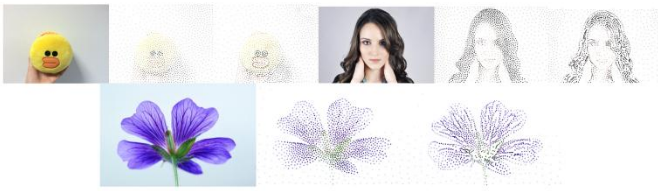
Like the process of point deletion, we add both Site points and Voronoi points in proportion to the brightness of the image difference between two frames. We repeat the process until the number of points in the current frame is same as to the number of points after deleting each point.

5 Results

At first, we experiments on an image prior to experiments on a video. In Figure 2, we can confirm that our results have the clearer edge lines of all the sample images than Thomas's results. In Figure 2 (a), we can see the exaggerated line features in the area of eyes and mouth of the character on the photo. Figure 2 (b) is the result of a human picture. We can see the exaggerated features around hair and eyebrows to keep the characteristics of the original photo. In Figure 2 (c), the input image is about nature and our result reflects the special features on leaf vines. In Figure 3, we produce different styles of stippled image with 3 different point sizes.

To decide the proper number of initial Site points, we formulate an equation by using the global intensity and the number of pixels in image, as follows. *N* is the number of pixels in image and *density* is the global density of image. Through the many experiments, we can calculate the number of pixels by setting the *C* as 0.03. Table 1 shows parameters for the various results of stippled image.

$$\text{The number of initial site points} = N \times density \times C \qquad (1)$$



(a) A Character (number of initial points: 1200)
(b) A human (number of initial points: 1500)
(c) nature (number of initial points: 1200)
Figure 2. (left) input image (middle) Thomas's result (right) our result

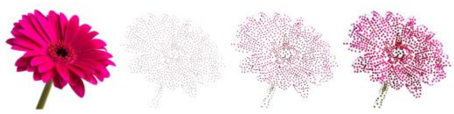


Figure 3. Results according to various point sizes

Table 1. Parameters for various stippled image

	The number of pixels	Image intensity	The initial number of Site points
1	139,600	0.293870	1230
2	166,000	0.268799	1338
3	240,000	0.206804	1488

Figure 4 and Figure 5 shows the stippled animation video from the Disney style input animation.

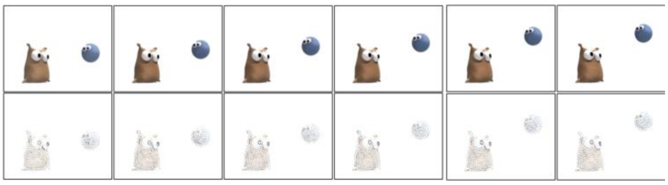


Figure 4. Stippled video for animation

Figure 5. Stippled video including characters

6 Conclusion and Future Work

We presented a new video stippling method to enhance the special features of moving objects in a video by the feature analysis of image sequences in the previous paper. In this paper, we proposed a modified point placement algorithm in order to exaggerate the features of moving objects and keep time coherency between video frames. A new algorithm for initial point set, point distribution, point deletion and addition are proposed. We can produce the various styles of video stippling by controlling input parameters. Our approach is automatic, so any user can use our system easily and conveniently. We expect that our system can be used in a various area such as an advertisement and exhibition. For a future work, speed improvement of the algorithm is needed by using GPU programming.

Acknowledgements

This work was supported by the Sungshin Women's University Research Grant of 2013.

References

[1] T. Houit, F. Nielsen, Video Stippling, *Advances Concepts for Intelligent Vision Systems. Springer Berlin Heidelberg*, (2011), 384 - 395.
[2] D. Kim, M. Son, Y. Lee, H. Kang and S. Lee, Feature-guided Image Stippling, *Computer Graphics Forum*, Vol. 27, No. 4 (2008), 1209 -1216.
[3] C.K. Yang, & H.L. Yang, Realization of Seurat’s pointillism via non-photorealistic rendering, *The Visual Computer*, **24** (2008), 303 - 322.
[4] Y.C. Wu, Y.T. Tsai, W.C. Lin, W.H. Li, Generating pointillism paintings based on Seurat's color composition, *Computer Graphics Forum*, **Vol. 32, No. 4** (2013), 153 – 162.
[5] O. Deussen, S. Hiller, C. Van Overveld, Floating points: A method for computing stipple drawings, *Computer Graphics Forum*, **Vol. 19, No. 3** (2000), 41 - 50.
[6] A. Secord, Weighted voronoi stippling, *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, (2002), 37 - 43.
[7] D. Mould, Stipple placement using distance in a weighted graph, *Proceedings of the Third Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, (2007), 45 – 52.
[8] M. Son, Y. Lee, H. Kang, S. Lee, Structure grid for directional stippling, *Graphical Models*, **73.3** (2011), 74 - 87.
[9] S.Y. Kim, R. Maciejewski, T. Isenberg, W.M. Andrews, W. Chen, M.C. Sousa, and D. S. Ebert, Stippling by example, *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*, (2009), 41 - 50.
[10] S.G. Oh and H. W. Byun, Feature Enhanced Video Stippling, to be appeared (2015)