A Non-photorealistic Rendering of Seurat's Pointillism

Hui-Lin Yang and Chuan-Kai Yang

National Taiwan University of Science and Technology, Taipei, 106, Taiwan, ROC

Abstract. In recent years, there has been a trend on simulating impressionism with computers. Among the various styles of impressionism, we are particularly interested in simulating the style of *pointillism*, especially the style presented by *Georges-Pierre Seurat*, as he was deemed the founder of pointillism. The reason that his style attracts us is twofold. First, the painting process of pointillism is extremely laborious, so simulating his painting style by computers is desired. Second, though several existing impressionism algorithms may approximate pointillism with point-like strokes, some delicate features frequently observed in Seurat's paintings are still not satisfactorily reflected by those general schemes. To achieve simulating Seurat's painting style, we made careful observations on all accessible Seurat's paintings and extract from them some important features, such as the few primitive colors, point sizes, and the effects of complementary colors and halos. These features have been successfully simulated and results are compared with not only Seurat's existing paintings, but also with previous attempted simulations.

1 Introduction

As one of the most famous non-photorealistic painting styles, *impressionism* concerns mainly the interaction between lights and shadows, or what the so called *impression* that a scene can bring to a person, without placing too much emphasis on the irrelevant details. Among the previously simulated impressionism styles, we are especially interested in *pointillism*, whose only painting primitive is *point*. As one of the most renowned painters in pointillism, Seurat, in particular, attracts our attention for two reasons. First, it is well known that a pointillism painting usually takes a very long time to finish, as it involves a very tedious process, which is potentially cut out for computers. Second, though generic techniques exist to approximate pointillism effects, such as [1,2,3,4], they are often too general to faithfully reflect all the features presented in Seurat's works. For example, there is a noticeable feature called *halo* that frequently appears in his paintings but has not been successfully simulated so far. To accurately emulate Seurat's painting style, we have made careful and detailed observations on numerous paintings of his in order to extract the common or important features that distinguish his work from others. To prove the success of our efforts, simulated images are presented at the end of the paper, together with their comparisons with the results produced by other existing techniques. Several Seurats' own paintings, after digitization, are also shown for serving as the comparison ground truth.

The rest of the paper is organized as follows. Section 2 reviews some existing works related to this study. Section 3 briefly describes what we observed from Seurat's paintings, i.e., the distinct features. Section 4 discusses how we simulate Seurat's painting

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 760–769, 2006.

styles based on our observed features. Section 5 presents our simulation results, which are also compared against previous works, accompanied by existing Seurat's paintings for being the ground truth. Section 6 concludes this paper and hints for potential future directions.

2 Related Work

It has been more than a decade since the name of *non-photorealistic rendering* was first brought to the computer graphics community by Winkenback et al. [5] in 1994, and a huge number of papers have been generated since then. Due to the limit of space, readers who are interested in learning more information regarding NPR are referred to the website at http://www.red3d.com/cwr/npr/, written by Craig W. Reynolds, who did an excellent and substantial survey on many aspects of NPR. Since our main goal is to convert existing images into pointillistic ones, we will pay attentions only to those similar studies.

Probably the first work to simulate pointillistic effect, Hertzmann proposed to mimic the hand-painted features by curved brush strokes, and in particular, he attempted the pointillistic effects by placing dense circles with perturbed hue and saturation values [2]. A layered structure is employed to process an image layer by layer, with progressively smaller brushes. Litwinowicz also proposed to simulate impressionism by varying the attributes of strokes [4]. Similarly, Hays et al. also presented an elaborated scheme to control various attributes of strokes, which lends itself to stylized images or videos [1]. Hertzmann proposed a rather different approach for simulating a stylized painting by performing *image analogies* [6]; however, many believe that his method cannot really produce better results than with a more domain-specific approach.

Most of the aforementioned approaches concentrate on simulating the strokes, while leaving color variation to random perturbation. Luong et al.'s work [3] adopted a different approach, where they pick colors in an *isoluminant* manner, so that it matches well with our perception system. Similarly, Jing et al. [7] divided an input image into tiles, within which colors are also perturbed in a an isoluminant way.

3 Seurat's Pointillism

Georges-Pierre Seurat (December 2, 1859 – March 29, 1891) was deemed as the founder of *neoimpressionism*, *pointillism*, or *divisionism*, and in fact each of these names has its own reason, which will be made clear later.

Eleven Colors. Seurat was known for only using relatively few colors when composing his paintings. While some said that he used only eleven colors, so far to our knowledge, we cannot be sure of what these eleven colors are. However, the traditional rainbow colors are surely included as those colors were recognized long time ago.

Halo Effect. Figure 1 is an example for demonstrating the *halo* effect, while Figure 2 for the *crease edge* effect.

Fig. 1. Left: a partial image of Seurat's *Sunday afternoon on the island of La Grande Jatte* painting. Upper-right: an enlarged portion of the left image marked with red, where one can see how the halo (along the boundary of the lady's dress) and complementary colors (the red dots on the lady's dark dress) take places. Lower-right: another enlarged portion of the left image marked with green, where the green color is juxtaposed with its adjacent colors in Chevreul's color circle (see Figure 3), such as yellow and blue, to make the green color stick out.

Complementary Colors. Another important feature is to utilize complementary colors. The main purpose of such usage is to make the original colors more prominent after being contrasted with their complementary counterparts. Figure 1 gives a demonstration for this. Note that Seurat's concept of complementary colors may be different from ours, as shown by the Chevreul's color circle in Figure 3, which was believed to be used by Seurat.

Divisionism. As mentioned before, pointillism or divisionism, represents a desired color by putting two primitive colors together properly. Figure 2 also demonstrates an example of such.

4 Non-photorealistic Rendering of Seurat's Pointillism

4.1 Color Space Conversion

After careful and detailed examinations of many Seurat's paintings, now the task is to simulate these observed features by computers. Before illustrating our simulations step by step, one thing worth mentioning is the color space conversion. In this work we select the well-known $CIEL^*a^*b^*$ or $CIELAB$ color system as our internal color manipulation format, not only for its decoupling of hue, saturation and lightness, but also for its *perceptual uniformness*, which means that the same amount of value change would result the same amount of visual perception change.

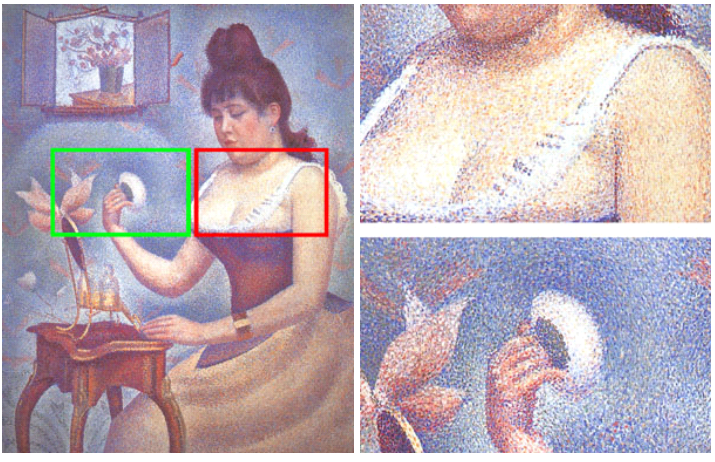

Fig. 2. Left: the image of Seurat's *young woman powdering herself*. Upper-right: an enlarged portion of the left image marked with red, where one can see the crease edge between the woman's body and her left arm being strengthened. Lower-right: another enlarged portion of the left image marked with green, where one can see the halo effect around the powder puff on the women's right hand.

4.2 Layering Structure of Points

Currently there are overall four layers in our implementation. The first, or the bottom layer is to serve as the background. Considering that a background color is often used to set the basic tone, we initially perform a *segmentation* process to partition a given image into regions of pixels with similar colors, then we locate the *brightest* point in each segment for using its color to color all the points falling within the same region.

The second layer, sitting on top of the first layer, is to add random but controlled variation, in terms of hues, to the previous points colored as backgrounds. The reason behind this randomness is apparent: it's natural that slight variation exists from stroke to stroke. The variation manifests itself in many ways, such as color, size, and shape, and we will discuss each of them in turn. In addition, to mimic the dense coverage observed from normal Seurat's paintings, this layer of points is independently added in by applying the Poisson disk method for another run.

We perform the operations of edge enhancements on the third layer. Through the use of *Canny edge detection* algorithm, we could identify essentially two kinds of edges: silhouette edge and crease edge. The distinction between the two is to be elaborated later. We then apply different edge enhancement schemes to strengthen these edges accordingly. Unlike what we did on the second layer, this enhancement process only replace the colors of the selected existing points from the second layer without generating new points.

The fourth layer, or the top layer is used to deal with the complementary colors. As mentioned before, complementary colors, commonly used in Seurat's paintings, serve as contrasts to consequently enhance the original colors. We selectively choose points from the second layer and substitute their original colors by their complementary

counterparts. As for how to find the complementary color of a given color will be explained shortly. And it could be inferred that the distribution of points in this layer should be sparser than previous layers.

4.3 Attributes of Points

As many classified Seurat's works into pointillism, it is self-evident that how to deal with points is of utmost importance.

Locations of Points. Among the many attributes of a point, such as position, color, size, and shape, position is no doubt one of the most important properties of a point. More concretely speaking, each point should be arranged in a proper location in the simulated painting so that the entire set of points as a whole are distributed uniformly with desired spacing among them. To have a simple and efficient solution, we adopt the idea of *Poisson disks* [8] to place points evenly within the entire image, while the pairwise distance between any two points is at least with some prespecified amount. Note here the size of a point in general is larger than a pixel, as we will associate a mask with each such point for simulating the real point strokes used during the painting process.

Colors of Points. Following the description of the four-layer structure of points, here comes the most important aspect of a point, i.e., the color. We first explain how color perturbation is realized on the second layer of points. The central idea is to perform the modulation in the $CIELAB$ domain, as the same amount of change in values leads to the same amount of alteration in visual perception in this color system. For a given color, represented as (L^*, a^*, b^*) in its $CIELAB$ representation, we define three thresholds, one for each of the three channels in $CIELAB$, respectively, to limit the range of variations, thus forming a 3D region of variation. Furthermore, such a variation distribution should have its peak centered at the original color (L^*, a^*, b^*) to be perturbed, therefore we enforce a *tri-normal* or *3D Gaussian* distribution on top of the 3D variation range. In practice, this tri-normal distribution could be implemented by the *Box-Muller* method [9].

Sizes of Points. To have a faithful simulation, we begin our study by making careful observations on Seurat's numerous paintings. In particular, we have to infer the point size used in his paintings in the real world, and calculate the corresponding point size on the image to be simulated and displayed on our screen with a desired resolution. This inference could be easily done by measuring the point sizes from the downsized reproductions of Seurat's paintings whose real size information is available. In our implementation, the side length of a point size is set to be 9 to 17 pixels.

Shapes of Points. Once the location and size of a point are settled, the next property comes to play is shape. For each point to have a natural look, like what a painter would do with his point strokes, each point should not always bear the same look. A naive approach is to manually scribble a dozen of "pre-defined" shapes for being the *selection pool*, while each time a shape randomly selected from this pool is randomly perturbed to form the final shape. Another way is to start with a solid

circle and then randomly and gradually grow from its boundary to "attach" more pixels. Currently we adopt the naive approach for its simplicity and acceptable results.

4.4 Edge Enhancement

To simulate the edge enhancement effect regularly presented in Seurat's paintings, we first apply Canny edge detection [10] to identify potential edges. Two kinds of edges are further distinguished as *silhouette edges* and *crease edges*. The former usually represents the edges along the boundary, thus being more salient, whereas the latter mainly refers to the internal edges. In practice, we differentiate them by the length of edges, although more sophisticated schemes or user assistance could be adopted for a more precise classification.

Silhouette Edges. Once an edge is classed as a silhouette edge, we proceed by strengthening it to generate the so called *halo effect* mentioned previously. To do this, we first need to determine the affected region, which is in turns determined by an associated radius, in terms of pixels. We could identify such regions by "rolling" a circle with a specific radius along all silhouette edges. Next, we need to further distinguish the *bright side* and *dark side* from an edge, as we will enforce different enhancing policies for them.

To enhance an edge, the basic idea is to enlarge the contrast; or more specifically, to make the bright side brighter and at the same time the dark side darker. We also hope such a contrast enhancement gradually decay as we move farther away from the corresponding edge. This is achieved by applying the following non-linear functions:

$$y = \frac{a}{2}[2(x - \frac{1}{2})]^{b} + 1 - \frac{a}{2} \tag{1}$$

to the dark side and

$$y = \frac{a}{2}[-2(x - \frac{1}{2})]^{b} + 1 + \frac{a}{2} \tag{2}$$

to the bright side, where x and y represent the distance (normalized to be between 0 and 1) from the current point to its corresponding edge, and the adjusting ratio, respectively. This ratio, defined to be the new lightness value divided by the old lightness value, governs how the lightness should be changed for edge enhancement. The two constants, a and b, are used to control the non-linear behavior and the blending with other non-enhanced areas, and typical values for them are 0.5 and 0.8, respectively.

Crease Edges. The enhancement of crease edges is relatively simple. We do not distinguish between the dark side and bright side of an crease edge, but instead just apply a non-linear contrast enhancement scheme as the following:

$$y = \frac{1}{2}[2(x - \frac{1}{2})]^{d} + \frac{1}{2} \tag{3}$$

where x and y represent the original lightness value and adjusted lightness value, respectively. The term d is again used to control the non-linearity. The affected

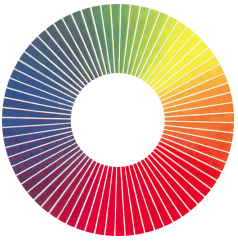

Fig. 3. The Chevreul's color circle

region, compared with that of the silhouette edge, is relatively small. A typical value of d is 0.5.

4.5 Complementary Colors and Color Dithering

As mentioned in the previous section, the existence of complementary colors is to enhance the original colors under contrast, thus being very important for numerous painting styles. One might surmise that finding the complementary color of a given color is trivial while it is not for at least the following two reasons. First, although modern color theory has enjoyed its success in many fields, it is in fact quite different from what traditional painters were using. For example, nowadays the color space of Hue, Saturation, and Value, or HSV color space for short, shows significant difference when compared with *Chevreul's color circle*, which is very likely the reference palette that Seurat adopted for his paintings. Therefore, it also indicates that to simulate the effect of complementary colors, instead of applying modern color theory such as the HSV model to find the color with the opposite hue, we should make use of Chevreul's color circle. Note that Chevreul's color circle, as shown in Figure 3, an extension of the eleven color model to contain 72 non-overlapping color ranges, is the fundamental color model that we use throughout for final color representation. Therefore a color, no matter coming from RGB or $CIELAB$, is converted to a representation by Chevreul's color circle before being put onto the simulated image. Such a conversion is done by first matching the color against all 72 color ranges, in terms of their *hue* values of the HSV model. Note that such a procedure is not only for finding a 72 color representation for a color, but also for locating the complementary color of a color in Chevreul's color circle. Also notice that complementary colors should appear with a relatively lower frequency so that they will not offset the original tones. And this could be achieved by selectively replacing a color in the second layer by its complementary counterpart, together with the checks to make sure such a color inversion will happen uniformly within an image.

However, as the gist of pointillism is not through color mixture but color juxtaposition, we thereby borrow the old technique of *halftoning* or *dithering* for simulating this effect. For example, assuming color a is to be approximated by a weighted average of color b and c, with the corresponding weighting factors to be 1 and 1, respectively. Instead of mixing the color before applying it, we approximate the original color with a 2×2 pattern by placing color b in two of the four regions while color c in the rest two regions.

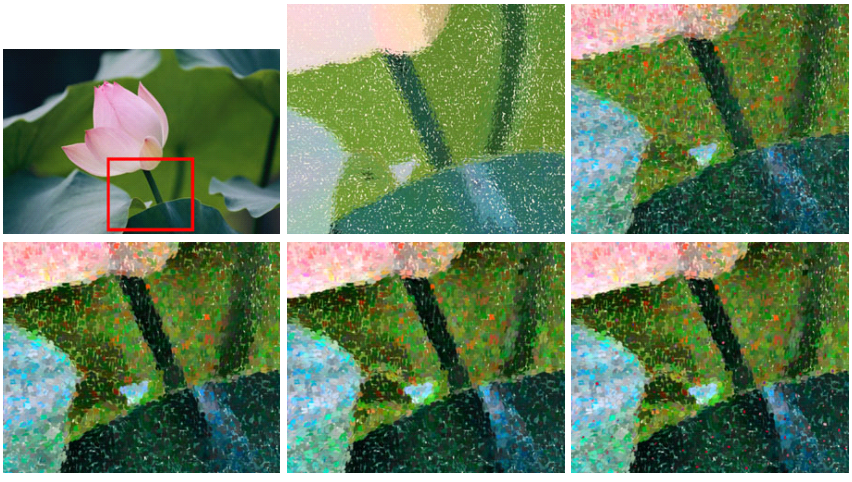

Fig. 4. From left to right and from top to bottom. 1. The source image of a pink flower. 2. The corresponding background layer image for an enlarged portion marked by red in the source image. 3. The previous image added with points of randomized colors. 4. The previous image added with silhouette edge enhancement. 5. The previous image added with crease edge enhancement. 6. The previous image added with complementary colors, thus the final image.

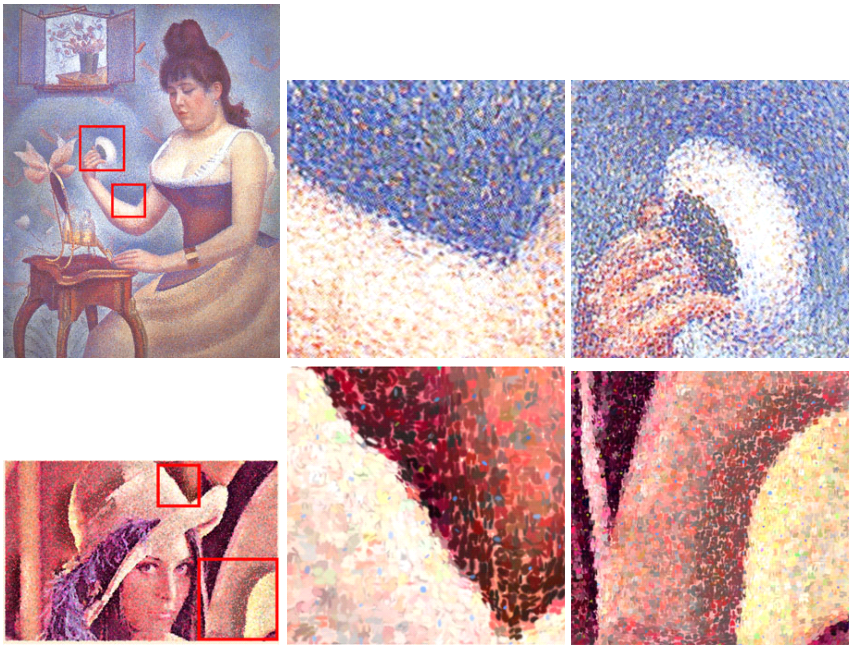

Fig. 5. From left to right and from top to bottom. 1. Seurat's painting. 2. An enlarged portion of the painting marked by red. 3. Another enlarged portion. 4. A simulated result of our system. 5. An enlarged portion of of our result marked by red. 6. Another enlarged portion.

5 Results

We have conducted our experiments on a Pentium IV 2.8GHz machine with 512MBytes memory running on a Windows XP operating system.

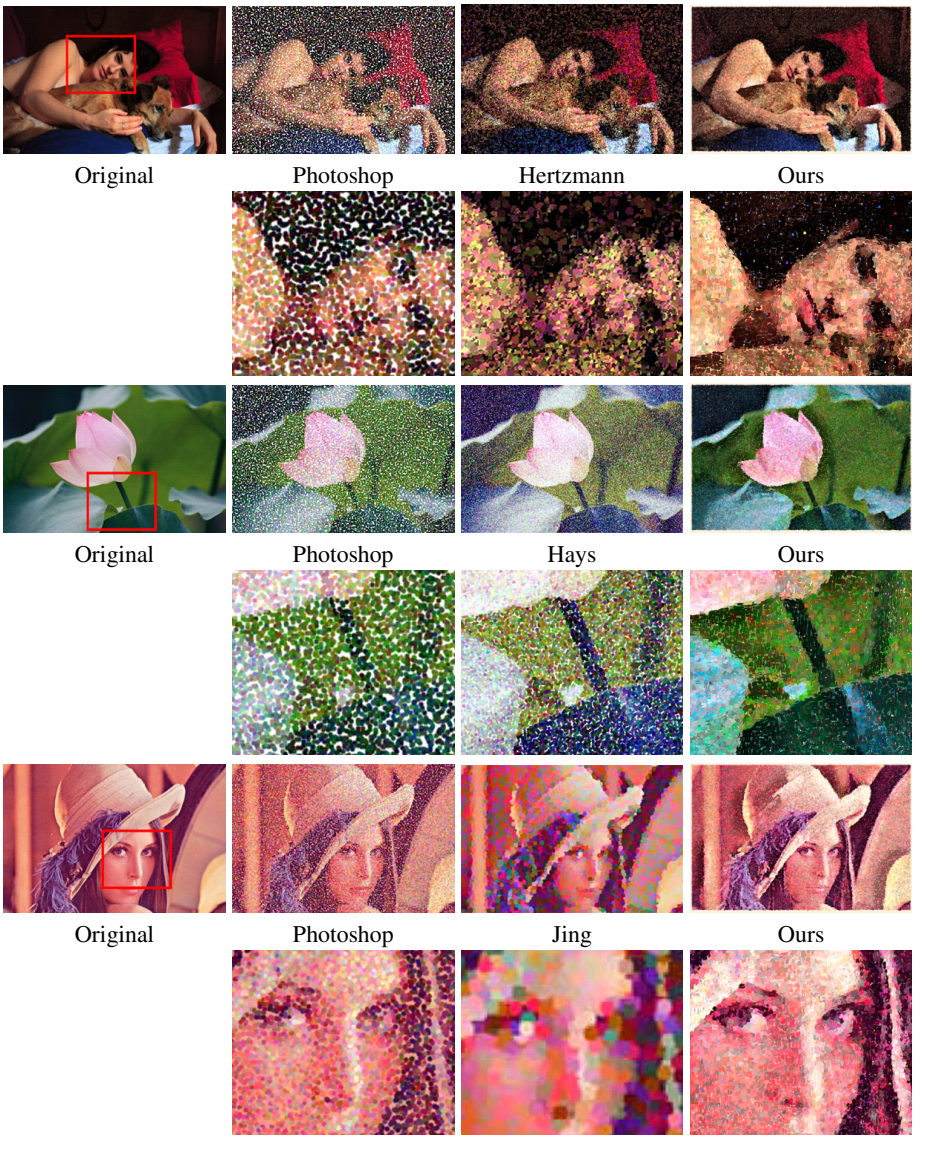

Fig. 6. Comparisons with Photoshop and three previous works. In each group, the original image is shown at the upper-left corner, and the rest of the images are divided into three columns, with each column demonstrating the results (and their corresponding enlarged portions) by using different approaches.

Figure 4 demonstrates our system through a "step-by-step" simulation of Seurat's painting styles, as described in the previous section.

Figure 5 compares the results of our system with one of Seurat's paintings, and as can be seen from this figure, our system does successfully simulate some important features generally presented in Seurat's work.

Finally, Figure 6 compare our results with the ones generated by Photoshop, Hertzmann et al. [2], Hays et al. [1], and Jing et al. [7]. Noticeable improvement on halo, complementary colors, as well as the overall appearance can be easily observed.

6 Conclusions and Future Work

We have implemented a system that could faithfully perform a non-photorealistic rendering of Seurat pointillism. Our success lies in simulating those features that we commonly observed from Seurat's paintings, such as halos and complementary colors, which have not yet been properly simulated by others. Results are demonstrated and compared with both Seurat's existing paintings and previous attempted simulations. In the future, we plan to automate our simulation process by reducing the involved parameter tuning as much as possible. We would also like to faithfully simulate the styles of other impressionism painters as well.

References

1. Hays, J., Essa, I.: Image and Video Based Painterly Animation. In: NPAR '2004. (2004) 113–120
2. Hertzmann, A.: Painterly Rendering with Curved Brush Strokes of Multiple Sizes. In: SIGGRAPH '1998. (1998)
3. Luong, T., Seth, A., Klein, A., Lawrence, J.: Isoluminant Color Picking for Non-Photorealistic Rendering. In: Graphics Inteface '2005. (2005) 233–240
4. Litwinowicz, P.: Processing Images and Video for An Impressionist Effect. In: SIGGRAPH '97. (1997) 407–414
5. Winkenbach, G., Salesin, D.: Computer-Generated Pen-and-Ink Illustration. In: SIGGRAPH '1994. (1994) 91–100
6. Hertzmann, A.: Image Analogies. In: SIGGRAPH '2001. (2001) 327–340
7. Jing, L., Inoue, K., Urahama, K.: An npr technique for pointllistic and mosaic images with impressionist color arrangement. In: International Symposium on Visual Computing 2005. (2005) 1–8
8. Yellot, J., I, J.: Spectral consequences of photoreceptor sampling in the rhcsus retina. Science **221** (1983) 382–385
9. Box, G.E.P., Muller, M.E.: A note on the generation of random normal deviates. The Annuals of Mathematical Statistics **29** (1958) 610–611
10. Canny, J.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **8** (1986) 679–698