Artistic Emulation
- Filter Blending for Painterly Rendering -

Crystal Valente and Reinhard Klette
Department of Computer Science, CITR, Tamaki campus
The University of Auckland, New Zealand
crystal@serato.com & r.klette@auckland.ac.nz

Abstract

This paper looks at painterly rending using filter blending to create a novel range of artistic effects. We look into several techniques in the field of painterly rendering and combine these different rendering styles together in a user defined way to create a new filter. This creates a user defined painting style based on aspects of different painting styles and processes. The final application uses a triangular region for human-computer interaction where each corner represents an artistic filter and the middle of this triangle represents the original image. A point within the triangle is chosen to determine the filters that are used for blending and their contributing strengths. Artistic effects are based on three base filters that can be tailored by the user to suit the specific subject matter of the image.

1. The Computer as an Artist

The notion of the computer as an artist may seem like a strange one, but much of what a painter does is in fact a sequence of repetitive tasks. This is the area where computers have a strong advantage. Some kind of human interaction may be required to give the image its artistic flair, but there are many parts of the painting process that are well suited to computer simulation. Painterly rendering techniques are ways of simulating these parts of the painting process in order to create an *artistic effect* from a source photograph.

There has already been a lot of important work done in the field of painterly rending. Paul Haeberli in [1] describes an interactive painting tool that simulates impressionist brush strokes. Haeberli's application allows users to choose different types of strokes and use them to paint a source image to a blank canvas. This is not an automated process, the strokes are placed by the user interactively, but it shows some of the first experiments with applying different types of artistic strokes to a photograph.

Aaron Hertzmann in [3] describes an algorithm for creating an artistic image from a photograph by building up layers of curved brush strokes. Hertzmann's algorithm is made up of several layers of strokes where the strokes decrease in size at each layer. The color of each stroke is determined by the color of the reference image at the stroke's starting location. This process leaves areas with flat color represented by larger strokes while more detailed areas are represented by smaller strokes.

Hertzmann's curved strokes algorithm's simplicity makes it ideal for extending it in a number of ways. James Hays and Irfan Essa in [2] extend Hertzmann's algorithm to work with animation as well as still photographs. Hays and Essa base their stroke placement on the proximity of strokes to an edge point. To keep temporal consistency between animation frames, they add and remove brush strokes between frames based on optical flow information.

Not all painterly rendering is stroke based. There have been interesting experiments with other ways of altering an image for artistic effect. Hertzmann et al. in [4] use artificial intelligence techniques to create an algorithm that 'learns' a painting style from examples. Papari and Petkov in [5] take yet another approach and distort images based on a geometric transform. This algorithm uses a perceived geometric structure that is created using a Glass pattern related to the structure of the original image. This geometric structure is then translated to the image itself to give the impression of paint being shifted around a canvas in impressionist swirls.

2. Supporting Creativity

It is common to use individual methods for generating an artistic effect by applying it to a chosen photograph. Real artists of course do not stick so rigidly to a single painting style. Each of these filters attempts to mimic one specific painting style or aspect of a painting process. Even a single painting, however, will usually contain a mixture of different styles and techniques. Often what is more ascetically pleasing is a combination of different styles.

Parts of the painting process are well suited to computer

automation but creativity itself cannot be so easily simulated. Computers can be instructed to apply a specific style to an image but humans are much better at determining what visual styles have the best 'artistic' effect for different types of subject matter. Our contribution to this field is to combine different filters together to create a range of artistic effects. The filters that are used to create an image and the strength of these filters can be tailored to suite the subject of the image and the artistic intention of the user. The novelty of this approach is that results can be generated by specifying the strength of an artistic style rather than dealing with complex mathematical parameters.

Our final application will have three different filters that are combined together in a triangle interface. Each corner of the triangle represents a different filter. The user chooses a style by selecting a point within this triangle. Points closer to edges are stronger while points near the middle of the triangle are closer to the original image. This allows users to tailor a style to suite the subject matter of an image without needing to understand any complex filter parameters.

We implemented and use three different filters. The first is Hertzmann's algorithm that simulates an oil painting using layers of curved strokes as described in [3]. The second filter generates pointillistic images based on the works of Georges Seurat by breaking an image up into a series of color distorted dots. This algorithm is roughly based on the work done in [6]. Our final filter generates impressionist images using Glass patterns as described in [5].

The challenge was that those filters are defined quite differently, and the blending needs to combine different mathematical approaches.

3. Design

In this section we give a description of the user interface that controls the filter blending. Next we define a strength parameter $s \in [0, 1]$ for each filter. This controls the amount of abstraction from the original image that the final image will display. For example, when $s = 0$ the final result will just be the original image and when $s = 1$ the filter is at maximum strength.

For each combination of filters we describe the method used to blend the filters together. We combine the techniques that are used to produce images with each of two filters, and apply a combination of these techniques as our filter to produce the final image. The method used to produce the filter combination is therefore different in our implementation depending on the filters that are used. The methods used to implement each of the possible combinations are described in the final three subsections of this section.

Figure 1. Screenshot of the triangle interface of our application running on a sample image.

3.1. User Interface

Before going into the details of each filter combination, we look at the design of the interface for choosing these combinations. Our final application has a triangular interface where each corner of the triangle represents a different filter and the center of the triangle represents the original image. This concept can be seen in Figure 1.

The style we want to apply to our image is chosen by clicking somewhere within this triangle. For example, if we wanted an image that was 50% pointillism and 50% Glass patterns, we would click halfway along the line between these two corners. Points along the edges of the triangle have the strongest filters. Points close to the middle of the triangle are closer to the original image.

3.2. Defining Strength Parameters

For each filter a strength parameter $s \in [0, 1]$ is defined by the user. This parameter is used to alter the degree of abstraction from the reference image. The way the s is used to effect each filter is described in the following paragraphs.

For the curved strokes algorithm the strength of the filter is mainly determined by a threshold T, but the brush radius R also has a bearing on how closely we are able to approximate the reference image. We use s therefore to produce a new threshold T_1 and a new brush size R_1 at each layer. We set

$$T_1 = sT \tag{1}$$

We also alter the brush radius if s gets small enough. This new value is restricted by a minimum brush size b_{min} which is different depending on the layer we are currently painting in the curved strokes algorithm. We have used $b_{min} = 4$ for the first layer and $m = 1$ for the final layer. Our new brush

size R_1 at each layer is defined by

$$R_1 = \begin{cases} R & \text{if } s \geq 0.5 \\ R & \text{if } 2 \cdot sR < b_{min} \\ 2 \cdot sR & \text{if } s < 0.5 \text{ or } 2 \cdot sR \geq b_{min} \end{cases} \quad (2)$$

For the pointillistic filter the strength of the filter can be determined by the point size and the amount of color distortion used. Point size is decreased the same as for the brush radius of the curved strokes filter. So given a point size R and strength parameter s, the new point size R_1 is defined by Equation (2). In this case we have used $b_{min} = 3$ for the background layer and $b_{min} = 1$ for the other two layers.

There are several different types of color distortion which are effected by the strength parameter s in different ways. The parameter s determines the degree to which the hue and saturation are distorted at certain points and the also the chance of this distortion occurring.

For the Glass pattern filter the strength parameter affects the stroke length a and the amount of noise added to the image. For a we define our new length a_1 by

$$a_1 = as \quad (3)$$

For the noise that is added, we alter the strength of the noise that is added to each pixel. The noisy image is created by taking the original pixel value p at each point in the image and adding a Gaussian white noise value n which gives us our final pixel value q. We add the strength parameter to this by instead defining q at each point in the image by

$$q = p + ns \quad (4)$$

3.3. Curved Brush Strokes and Pointillism

The curved brush strokes and pointillistic filters share a lot of their underlying concepts. To combine the two, the first step is to look at the similarities between the two approaches. Both build up brush strokes in layers with a different brush radius at each layer. The differences are the way that the new stroke position is determined, the color of this stroke, and the shape of the stroke. Our combined filter will use a three-layered approach. Four decisions need to be made for each brush stroke that is painted:

- stroke radius
- stroke position
- stroke color
- maximum stroke length

The degree to which these decisions approximate each filter is determined by an influence parameter $\alpha \in [0, 1]$ which defines a scale between pointillism and curved brush strokes with 0.0 being pointillistic and 1.0 curved brush strokes.

For our stroke radius, we want a stroke size between the stroke sizes of the two filters at each layer. We want this value to have a random appearance but for these random numbers to be centered around an area defined by the influence parameter. We determine this size by getting a normally distributed random number z where the mean and standard deviation vary according to the influence parameter. For brush sizes b_p and b_c at the current layer, where z is generated using mean m and standard deviation σ, we determine our final stroke radius b_{final} by

$$b_{final} = b_p + z(b_c - b_p) \quad (5)$$

If $b_p = b_c$ then we can skip this stage. Parameters m and σ are determined by the formulas

$$m = \begin{cases} 1.0 - \alpha & \text{if } b_p \leq b_c \\ \alpha & \text{if } b_p > b_c \end{cases} \quad (6)$$

$$\sigma = \begin{cases} 1.0 - |m - \alpha| & \text{if } b_p \leq b_c \\ 1.0 - |2 \cdot m - 1| & \text{if } b_p > b_c \end{cases} \quad (7)$$

For determining the position of the brush strokes at each layer, we first determine two positions $p_p = (x_p, y_p)$ and $p_c = (x_c, y_c)$. We divide the image into a grid of size b_{final} and choose one stroke point for each grid point. For p_p we take a point at a random location in the neighborhood. For p_c we find the point with maximum color distance from the reference image. Our final point p_{final} is a point between p_p and p_c where the influence parameter α determines how close p_{final} is to the point chosen by the pointillistic filter. This is determined by the formula

$$p_{final} = (\alpha x_p + (1.0 - \alpha)x_c, \alpha y_p + (1.0 - \alpha)y_c) \quad (8)$$

The stroke color will vary between the pointillistic filter, where a number of color distortions are performed, and the curved stroke filter where we simply take the value of the reference image at this point without distorting the color in any way. (When we implemented the curved stroke filter we already defined a way to effect the strength of the color distortions.) We apply the same theory here but instead of using the strength parameter s on its own we use

$$s_{final} = s\alpha \quad (9)$$

The maximum stroke length l_{final} is determined in a similar way to the stroke radius. Given a user-defined stroke length l_{max}, l_{final} is determined by the formula

$$l_{final} = 1 + z(l_{max} - 1) \quad (10)$$

where z is again a normally distributed random number with mean $m = I(f_c)$ and standard deviation $\sigma = 1 - |m - \alpha|$.

3.4. Curved Brush Strokes and Glass Patterns

The curved brush strokes and Glass pattern filters have quite different approaches to the way they alter the image so there is no obvious scale between them like with the curved brush strokes and pointillism. Instead we will find a way to mix the concepts of the two filters together. Again we use an influence scale $\alpha \in [0, 1]$, but this time we define $\alpha = 0.5$ as the point where both filters are at full strength. As α tends toward 0 the Glass pattern filter loses strength and as α tends toward 1 the curved brush stroke filter loses strength.

When $\alpha = 0.5$ we follow the usual method of the curved brush stroke filter, but instead of strokes following the normal of the image gradient, our brush strokes follow the vector field $\mathbf{v}$ as defined in [5]. As α tends toward 0 we want the filter to gradually stop following this vector field and instead revert to following the normal of the image gradient. We do this by calculating the result of both methods for determining the new stroke control point and taking the appropriate method between them depending on the value of α. For each new control point, we calculate the normal of the image gradient $\mathbf{g} = (x_g, y_g)$ and the value of the vector field $\mathbf{v} = (x_v, y_v)$. The next control point is placed at the point $\mathbf{p} = (x, y)$, which is calculated by

$$\mathbf{p} = 2 \cdot \alpha\mathbf{v} + 2(0.5 - \alpha)\mathbf{g} \tag{11}$$

We also set the maximum length l of the brush strokes based on the Glass pattern length parameter a. The influence of a over the length of the brush strokes decreases as α tends toward 0. For a maximum stroke radius R_{max} and a minimum stroke radius R_{min} (i.e., the stroke radius at the first and final layer respectively of the curved stroke algorithm), l is defined by

$$l = \begin{cases} 2 \cdot aR_{min} & \text{if } \alpha \geq 0.5 \\ 4 \cdot \alpha aR_{min} + 8(0.5 - \alpha)R_{max} & \text{if } \alpha < 0.5 \end{cases} \tag{12}$$

When $\alpha > 0.5$ we start to decrease the influence of the curved brush stroke filter. This is done by gradually reducing the strokes radius parameters R_{max} and R_{min}, and the threshold T. For each of these parameters, the original value v_1 is transformed to the new value v_2 as follows:

$$v_2 = 2(1 - \alpha)v_1 \tag{13}$$

We also add noise to the reference image as α increases to make the impressionist whirls more visible. We generate a small random number c and add $2(\alpha - 0.5)c$ to each RGB color channel of the image. When $R_{min} \leq 1$ we discard the curved stroke algorithm and simply run the Glass pattern algorithm with the noise level set according to α as above.

Figure 2. The effect of the combination of the curved brush stroke and Glass pattern filters for varying values of α. *Upper left:* Original image. Original photograph by Luan You. *Upper right:* $\alpha = 0.2$. *Lower left:* $\alpha = 0.5$. *Lower right:* $\alpha = 0.8$.

3.5. Pointillism and Glass Patterns

Like the previous two filters, the pointillistic and Glass pattern filters have quite different approaches to image manipulation. We adopt a similar approach to Section 3.4 and use an influence scale $\alpha \in [0, 1]$ where $\alpha = 0.5$ is the point where both filters are at full strength. As α tends toward 0 the Glass patterns filter loses strength and as α tends toward 1 the pointillistic filter loses strength.

The painting process is divided into three layers. At the background layer larger points are placed by a sampling of poisson disks in order to cover the canvas without much color distortion. At the middle layer, color distorted points are placed by dividing the image into a grid as in the curved brush strokes filter. For each grid point, if total error of the pixels in the neighborhood is above a threshold then a point is placed at the location of maximum error. The final layer adds some edge enhancement.

The background layer paints points as usual, but the other two layers mix their point placement with the Glass patterns method. For each point that is placed by the pointillistic part of the filter, we take the color of this point and use the Euler algorithm to paint more points of this color along the arc of a streamline defined by the vector field $\mathbf{v}$. We want the combined filter to retain the look of being made up of points, so these extra points that are generated have a probability of not being painted to the canvas to prevent the filter having the look of smooth strokes.

When $\alpha \leq 0.5$ then this is the full process of the filter. The variable $prob$ ensures that the filter tends toward a pointillistic look as α decreases since gradually less of the points along the arc are actually painted.

When $\alpha > 0.5$ we need to take a slightly different approach. We want the distortion by the Glass pattern to

Figure 3. Image using a combination of the curved strokes and pointillistic filters with $\alpha = 0.5$. Scene from Buckinghamshire, England. Original photograph by Angela Palmer.

increase, so we set $n = (1 + \alpha)n$. We also decrease the radius of the points so that we tend toward manipulating pixels rather than larger areas. Our radius R is set to $R = (1 - \alpha)R$. We also want to get rid of the color distortion of the pointillistic filter after a point. We do this by altering our strength parameter in regard to the color distortion. When $\alpha > 0.75$, for strength parameter s, we determine our new strength parameter s_{final} by

$$s_{final} = 4(1 - \alpha)s \tag{14}$$

Finally, as in Section 3.4, we add random noise to the reference image by generating Gaussian noise c and adding $2(\alpha - 0.5)c$ to each RGB color channel of the image.

4. Results and Evaluation

This section presents and evaluates results of running our application on some sample images. All the filters are run at full strength $s = 1.0$. Our results can be seen in Figures 3, 4 and 5.

Since we are trying to approximate art, any evaluation will of course be very subjective. At a glance, our results are interesting and visually pleasing. Each combination creates a nice artistic effect that looks inspired by but distinct from the filters it is made up of. We will look at each image separately to give a more thorough evaluation of the effects that are achieved in each of these images.

4.1. Effects

Figure 3 shows a combination of the curved stroke and pointillistic filters. This gives us a variety of brush shapes and sizes. With the influence parameter at this level, less of the smaller points can be seen, but pointillism's interesting color distortions prove very effective when used with larger strokes as well as small points. We get an effect similar to the curved strokes filter but the variety of stroke sizes and color distortions makes for an effect that is more 'artistic'.

Figure 4 shows a combination of the pointillistic and Glass pattern filters. The effect of color distortion is particularly striking in images like this where the saturation distortion brings out colors that are not normally noticeable in the scene. The color distortion mixes well with the geometric distortion as both implement different ideas of impressionism, departing from realism to give a nice 'impression' of the scene.

Using the points gives the geometric structure created by the Glass patterns an interesting texture that works well for some images. It also puts more emphasis on the objects in the scene as points flow around contours rather than being randomly placed. This effect is not suited to all images, but it works very effectively for particular scenes, giving a nice artistic effect.

Figure 5 shows a combination of the curved stroke and Glass pattern filters. This combination creates a nice artistic effect that uses aspects of both filters to create an image that is more analogous to the painting process than any of the standalone filters. It incorporates the layers of the curved strokes filters and also uses the vector field from the Glass pattern filter to add the idea of motion to the image. The result is an image with a variety of stroke sizes and nice layered effect but with strokes that have a nice flowing movement around object contours. This remedies the one of the problems with the Glass pattern filter. We get the same movement but with a user definable stroke size rather than strokes with a set 1 pixel width. The filter combination

Figure 4. Image using a combination of the pointillistic and Glass pattern filters with $\alpha = 0.6$. Scene from a remote village near Laguna Colorada, Bolivia. Original photograph by Marian Arnold.

Figure 5. Image using a combination of the curved strokes and Glass pattern filters with $\alpha = 0.5$. Original photograph by Koria Stevens.

is very effective and brings many images closer to approximating a real artwork.

All three filter combinations successfully create a novel artistic effect that incorporates different aspects of the painting process. Every image is different of course, so not all of these filters will be appropriate for every scene. Our results are successful because they create a range of artistic effects that are distinct and achieve success in different areas. This allows us to pick an artistic style that suits the subject matter of the image.

4.2. Runtimes

Finally, we take a look at the runtimes of each of our filters on images of different sizes. The results here average the time taken over 50 sample images of the same size. They are run on a MacBook 2 GHz Intel Core Duo with 1.5 GB memory running Mac OS X 10.5.8. The filter combinations are all run with $\alpha = 0.5$. We expect times to be slightly different if the influence parameters are altered. The results can be seen in Figure 6.

As seen in Figure 6, the pointillistic filter is the fastest and runs close to real time for all image sizes. Curved strokes runs slightly slower, but still within a reasonable wait time even for larger images. The Glass patterns filter, although it gives us some of the most interesting results visually, is a lot slower than the other two filters. It will need to be optimized to be run on larger images.

The filter combinations are not any slower than the standalone filters. The curved strokes and pointillism combination runs somewhere between the runtimes of the two filters. Both combinations involving Glass patterns run much slower than this. They are actually faster than the standalone Glass patterns filter however. This is because although the combined filters are still slowed down by generating the vector field $\mathbf{v}$, the overall painting process is faster because these filters work with larger points or strokes rather than individual pixels. The effect of combining filters does not add noticeably to the runtime of the filters.

	Image size (pixels)	
	700 x 525	1000 x 750
Curved strokes	2.1 s	4.0 s
Pointillism	0.8 s	1.6 s
Glass patterns	12.1 s	24.8 s
Curved strokes and pointillism	1.6 s	3.1 s
Pointillism and Glass patterns	6.8 s	13.9 s
Curved strokes and Glass patterns	7.6 s	15.4 s

Figure 6. Runtimes of the different filters used in our application on a variety of image sizes.

5. Conclusions

In this paper we have designed a method of creating an artistic effect using filter blending. Our application has a triangular interface where each corner of the triangle corresponds to a different artistic filter. The user defines the type of style that they want their image to have by clicking somewhere within this triangle. The final image will have an artistic style that is a blend of these filters, where the influence of each filter is determined by the point chosen in the triangle. This allows users to create their own artistic style by mixing these preset filters to suite the subject matter of the photograph. This produces a range of novel artistic effects that are created by specifying a desired artistic style.

References

[1] Haeberli, P.: Paint by numbers: Abstract image representations, in Proc. *SIGGRAPH*, pages 207–214, 1990. 1

[2] Hays, J., and Essa, I.: Image and video based painterly animation, in Proc. *Non-Photorealistic Animation Rendering*, pages 133–120, 2004. 1

[3] Hertzmann, A.: Painterly rendering with curved brush strokes of multiple sizes, in Proc. *SIGGRAPH*, pages 453–460, 1998. 1, 2

[4] Hertzmann, A., Jacobs, C., Oliver, N., Curless, B., and Salesin, D.: Image analogies, in Proc. *SIGGRAPH*, pages 327–340, 2001. 1

[5] Papari, G., and Petkov, N.: Continuous glass patterns for painterly rendering, *IEEE Trans. Image Processing*, **18**:652–664, 2009. 1, 2, 4

[6] Yang, C.-K., and Yang, H.-L.: Realization of Seurat's pointillism via non-photorealistic rendering, *The Visual Computer*, **24**:303–322, 2008. 2