Painterly Rendering using Image Salience

J. P. Collomosse and P. M. Hall.
Department of Computer Science,
University of Bath,
Bath, BA2 7AY, England.
{jpc | pmh}@cs.bath.ac.uk

Abstract

The contribution of this paper is a novel non-photorealistic rendering (NPR) technique, capable of producing an artificial 'hand-painted' effect on 2D images, such as photographs. Our method requires no user interaction, and makes use of image salience and gradient information to determine the implicit ordering and attributes of individual brush strokes. The benefits of our technique are complete automation, and mitigation against the loss of image detail during painting. Strokes from lower salience regions of the image do not encroach upon higher salience regions; this can occur with some existing painting methods. We describe our algorithm in detail, and illustrate its application with a gallery of images.

1. Introduction

Non-photorealistic rendering (NPR) is an area of increasing interest within the field of computer graphics, and many techniques have been developed to render both two and three dimensional source data for the purposes of aesthetics and visualisation. We restrict our attention to a subset of these methods, specifically that which concerns the generation of a natural or 'hand-painted' effect upon 2D input images.

Many existing painterly rendering techniques share the similarity that they are concerned with the emulation of artistic media. A painterly effect is invariably produced by forming a set of virtual 'brush strokes' upon a canvas at predetermined spatial intervals. Attributes such as stroke colour and orientation are often derived from the original image, at the pixel locations corresponding to the origin of each stroke. These generated strokes are necessarily larger than the original image pixels, and thus introduce a spatial quantisation effect which can produce aesthetically pleasing renderings.

There are two major problems that arise using such an approach. First, temporally coherent animations are difficult to produce, as the positions of brush strokes must move with the reference image rather than adhere to the image plane. Second, detail within the rendered image can be lost as strokes are composited upon the canvas; strokes from unimportant scene background may encroach upon more important foreground objects. It is this latter problem that we address in our work.

We present a novel method of rendering 2D images to produce a painterly effect. Our method requires no user interaction, and uses statistical analysis to determine the importance or *salience* of pixels within the original image. Stroke attributes are derived from salience and gradient information in the image, producing an aesthetically pleasing painting whilst mitigating against loss of detail.

The remainder of this paper is organised as follows. In the next section we explain our research in the context of related literature. We then give a detailed description of our algorithm in Section 3, illustrating the application of our algorithm with a gallery of images in Section 4. We conclude the paper in Section 5, with a critical appraisal of our technique.

2. Background

Research into the field of NPR, specifically the development of painterly renderers, began to gain momentum with Haeberli's interactive paint systems[2]. These permitted a user to rapidly generate impressionist style "paintings" by creating brush strokes, the colour and orientation of which were determined by point-sampling a reference image. This process was labour intensive, and approaches toward automating the production of such paintings were later presented. Notably, Haeberli's concept of a simple particle system was extended to fit a 3D rendering

pipeline[9], and toward the automatic production of painterly animations[8, 5].

As automation increases, reliance upon human interpretation through interaction can decrease. In Haeberli's system, a user might choose a smaller brush to render more detailed areas of an image; to mimic this behaviour in an automated system is more difficult. Early commercial attempts[3] to derive automated renderers from Haeberli's systems were based for the most part upon pseudo-randomness, and paid little attention to emulating the interactive user's adaptation to detail. Such automated renderings often appeared unnatural or distorted.

Approaches were later presented which drew upon local image processing techniques to provide an interpretation of the image, and thus automatically estimate some stroke attributes such as scale. Litwinowicz[8] employed short linear paint strokes, which were clipped to edges detected within the image. Hertzmann[4, 5] used Gaussian convolution to blur the original image at various scales; larger strokes are painted where artifacts persist after larger scale blurring. These methods demonstrate some adaptation to detail over an image based upon small, localised sampling and so can be susceptible to noise. Treavett and Chen[12] propose a method of using local statistical measures to determine stroke parameters; specifically the intensity variance of the image. Strokes are oriented and scaled such that they lie upon contours of minimal intensity variance. A recent extension to this technique takes into account chromaticity by substituting colour variance[11].

Our method is unique in that it makes use of a *global* algorithm, which performs a robust statistical analysis upon the entire image to determine the importance or *salience* of image pixels. We use salience, supplemented with gradient information, to automatically determine stroke parameters, and so produce a painterly effect whilst retaining detail within the rendered image.

3   Painting Algorithm

In this section we present our rendering algorithm. Our method accepts a single 2D image, such as a photograph, and produces a 2D painterly rendering of identical size to that image. There are three stages to our algorithm, which in execution order are:

1. Calculation of gradient information.

2. Calculation of a salience map.
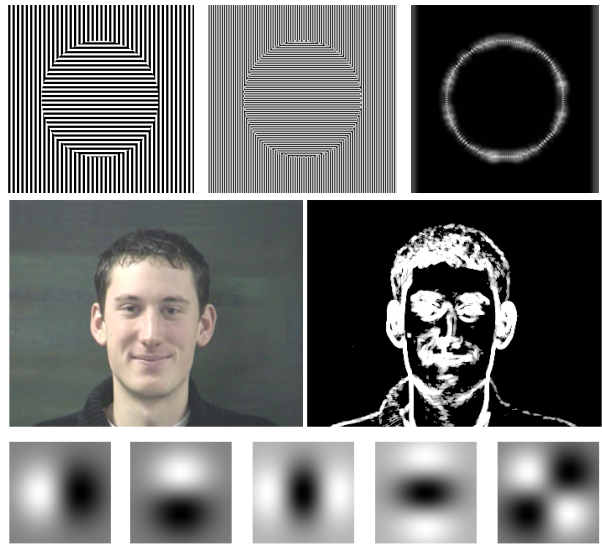
3. Generation of brush strokes.

Figure 1. Top: An illusory circle, edge detected, and salience mapped; showing salience mapping can "pick out" the circle where edge detection fails. Middle: a typical source image and corresponding salience map; lighter pixels are more salient. Below: the 5 differential convolution kernels, at one σ value, used to create measures $\vec{\phi}$.

We initially derive gradient information from the reference image, by convolving the intensity image of the input with a Gaussian derivative of first order, at a predetermined scale. Peaks in the first derivative of intensity often correspond to edges within an image, and although we do not detect edges *per se*, we use gradient information to partially determine the attributes of strokes.

The remaining two stages are of greater importance, and we explain each in turn in the following subsections.

3.1   Image Salience

We wish to evaluate the perceptual importance, or salience, of individual pixels within the source image. Reliably extracting such information from a general 2D image is a problem as yet unsolved by the vision community. However, we make progress by modifying a technique due to Walker *et al*[7], who observe that such salient pixels are uncommon in an image. The basic technique is to model the statistical distribution of a set of measures associated with each pixel, and to isolate the outliers of this distribution. The pixels corresponding to these outliers are regarded as salient.

To compute these measures, $\vec{\phi}$, over each pixel, we convolve the image with a set of origin-centred 2D Gaussian derivative filters, over various scales σ. Specifically we use 5 first and second order directional derivatives: $\partial G(x, y; \sigma)/\partial x$, $\partial G(x, y; \sigma)/\partial y$, $\partial^2 G(x, y; \sigma)/\partial x^2$, $\partial^2 G(x, y; \sigma)/\partial y^2$, and $\partial^2 G(x, y; \sigma)/\partial x \partial y$. These filters smooth the image before computing the derivative; they respond well to edge and other signals of characteristic scale σ, but as Figure 1 shows, our method is more general than edge detection. We filter using octave intervals of σ, as such intervals contain approximately equal spectral power. In our implementation we use σ values of 1, 2, 4 and 8; thus with each pixel we associate a vector $\vec{\phi}$ of $20 = 5 \times 4$ components.

For an image of M pixels we will have M vectors $\vec{\phi} \in \Re^n$, where for us $n = 20$. We assume these points are Gaussian distributed, which we represent using an eigenmodel[10]; a simple and convenient model that works acceptably well in practice. The eigenmodel provides a sample mean μ; a set of eigenvectors, each a column in orthonormal matrix $\mathbf{U}$; each eigenvector has a corresponding eigenvalue along the diagonal of Λ. An eigenmodel allows us to compute the squared Mahalanobis distance of any point $\vec{\phi} \in \Re^n$:

$$d^2(\vec{\phi}) = (\vec{\phi} - \mu)^T U \Lambda U^T (\vec{\phi} - \mu) \qquad (1)$$

The Mahalanobis distance measures the distance between the point and the sample mean, and does so using the standard deviation (in the direction $(\vec{\phi} - \mu)$) as a unit measure. The Mahalanobis distance therefore provides a convenient way of evaluating the relative salience of individual pixels.

3.2 Stroke Generation

We now describe the method by which brush strokes are generated, to create a painterly rendering of the original image. We use a novel painting technique adapted from earlier work[2, 8], to generate a series of three-dimensional brush strokes; inverted cones with superquadric cross-section. The superquadric class of functions can be represented by the equation:

$$\left(\frac{x}{a}\right)^{\frac{2}{\alpha}} + \left(\frac{y}{b}\right)^{\frac{2}{\alpha}} = r^{\frac{2}{\alpha}} \qquad (2)$$

where a and b are normalised constants ($a + b = 1$) which influence the horizontal and vertical extent of the superquadric respectively, and r is an overall scaling factor. We observe that (2) reduces to the general equation
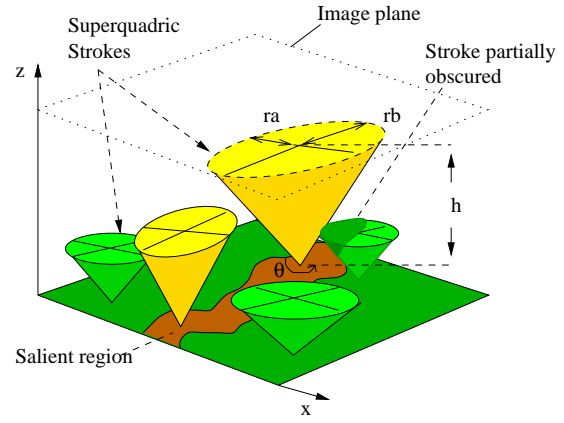

Figure 2. Strokes take the form of inverted cones with superquadric cross-section, and are z-buffered to produce the final image.

for a closed elliptic curve when $\alpha = 1$, tends toward a rectangular form as $\alpha = 0$, and toward a four-pointed star as $\alpha \rightarrow \infty$. Thus the superquadrics can express a wide variety of geometric forms, using a single set of parameters.

We sample the original image at regular spatial intervals to produce the three dimensional brush strokes, shading each stroke according to the colour of the original image at the point of sampling. Each stroke is z-buffered and the result is projected orthogonally onto the (2D) image plane to generate the final image (Figure 2).

There are seven parameters to each stroke; a, b, r, α (from equation 2), colour $j(c)$, orientation angle θ and height h. Parameter α determines the form of the stroke, and is preset by the user. Low values (< 1) of α create cross-sections of a rectangular form, giving the image a chiseled effect, whilst higher values of α produce jagged brush styles. Function $j(c)$ transforms, or 'jitters', the hue component of stroke colour c by some small uniformly distributed random quantity, limited by a user defined amplitude ϵ. By increasing ϵ, impressionist results similar to Haeberli's[2] can be produced. Further brush styles can also be generated by texturing the base of each cone with an intensity displacement map, cut at a random position from a sheet of texture; we find that this process greatly enhances the natural, 'hand-painted' look of the resulting image. The remaining five stroke parameters (a, b, r, θ, and h) are calculated by an automated process which we now describe.

The scale of the cone base r is set inversely proportional to salience magnitude. This causes small, definite strokes to be painted in the vicinity of salient detail in the image. Larger strokes are used to shade non-edge areas. This
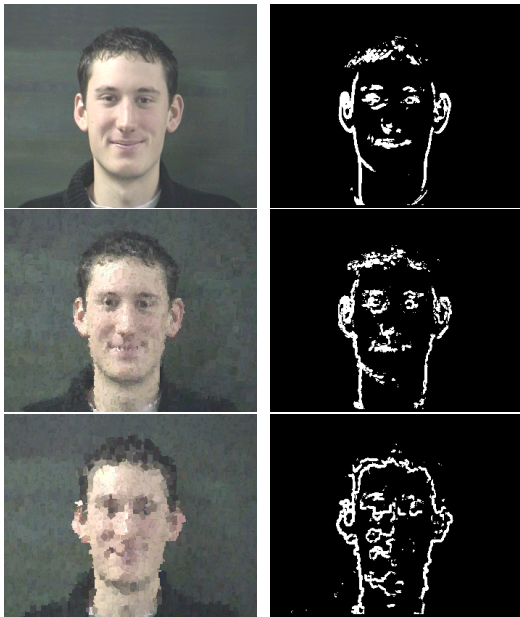
Figure 3. Three images of identical subject. Top: Original image (left) and associated, thresholded, salience map (right). Middle: painterly rendering with salience (middle) and associated edge map. Bottom: painterly rendering without salience (bottom) and associated edge map.

mimics the manner in which an artist may paint large strokes to abstract detail in non-salient areas, and sharply define the salient edges of an image.

Stroke orientation θ is derived from gradient orientation; the larger axis of the superquadric is aligned tangential to the edge direction. In areas where gradient magnitude is low, orientation derived in this manner becomes less reliable. We therefore vary the eccentricity of the superquadric (a, b) in relation to the magnitude of the image gradient at the position sampled. If the gradient is low, then $a \approx b$, and orientation becomes less important as the superquadric is not greatly expressed in either horizontal or vertical directions. Where image gradient is high, then $a > b$ and the superquadric stretches out. One emergent property of this approach is that strokes typically stretch along an edge, tending to merge, which often causes edge highlights to appear as though produced by fewer, longer strokes. This is typical of the manner in which an artist might manually render such highlights, and adds aesthetic quality to the image.

Stroke height h, is set proportional to image salience at the point of sampling. Higher salience image pixels tend to correspond to the features and detail within the image, and so produce strokes of greater height to protrude over the lower salience strokes in the z-buffer.

In effect, the least salient strokes are laid down first, much as an artist might use a wash to generate wide expanses of colour in an image, and fill in the details later. Without this sensitivity to salience, the rendering procedure can obscure regions of high salience with strokes of lower salience, demonstrated by Figure 3. By setting the conic height h proportional to salience, detail is conserved within the painting - especially about the eyes and nose (Figure 3, middle). Ignoring the implicit ordering of strokes can still produce a painterly effect, but without the adaptive sensitivity to salient detail that our method provides (Figure 3, bottom). By inspection we make the qualitative observation that the majority of salient detail in the original has been conserved in the salience-adaptive painting; this is not true for the lower painting.

Typically salience in an image correlates with edges and detail (although this may not always be the case, see Figure 1). Tall, thin, highly-elongated strokes tend to be painted tangential to edges, while short, fat strokes are painted in regions of low detail (see, for example, the phone-box in figure 4). Taller strokes rise above the canopy of shorter strokes, lending an implicit ordering to painting. The result is that salient areas are conserved within the final rendering.

4 Gallery of Images

We present the results of applying our algorithm to several 2D images in Figure 4. Source material was captured using a scanner or digital video camera; video images were acquired by sampling alternate scan-lines of a chosen frame of video – this suppresses edge artifacts that would otherwise appear due to hardware interlacing of the image.

The images of the truck and gondola were rendered with superquadric shape parameter $\alpha = 1$. Small, random inflections were made to stroke colour in the gondola image, by setting the jitter parameter $\epsilon = 10$; the effects are particularly noticeable in the vicinity of the water and give an impressionist effect to the image. The image of the truck demonstrates the wide sweeping strokes, covering low salience areas of the image such as the pavement. This is in contrast to the detailed strokes used to paint detailed areas such as the tree, truck bumper, or owner. The image of Rialto bridge was rendered with $\alpha = 2$ giving a more jagged brush effect. The phone-box has been rendered with $\alpha = 0.5$; in particular we draw attention to the detail

Figure 4. A gallery of images illustrating the application of our rendering algorithm. A section of the phone-box rendering has been magnified, demonstrating the alignment of strokes tangential to the salient window frames.

retained in the window frames. Strokes have been aligned tangential to the edges of each frame, merging to create sweeping brush strokes. The strokes rendering the window glass do not encroach upon the window frames, which are more salient, and so conserve detail in the painting.

Our gallery also includes images of a face and guitar, rendered in a compositional style reminiscent of the Cubists, such as Picasso or Braque. A full explanation of the compostion method is beyond the scope of this paper; an account appears elsewhere [1]. Briefly, we use our salience measure to identify *high-level features* such as eyes, or ears in a set of input images. A subset of such features across all input images is selected. Features in the subset undergo geometrical distortion, and are then composited into a target image, creating "islands" of distorted features set against a blank background. The background is then filled in using distorted versions of the original input images as a reference, this is a non-trivial procedure. The importance of the Cubist examples lies in the algorithm's use of salience to detect high-level features; it shows that methods for producing non-photorealistic images predicated on some measure of salience are of interest.

5 Discussion and Conclusion

We have presented an automatic NPR technique capable of rendering 2D images in a painterly style, using three-dimensional conic strokes of superquadric cross section. The implicit ordering of cones is governed by a statistical measure of salience, which mitigates against low salience strokes encroaching upon high salience areas. We argue that our approach increases the conservation of detail within the painted image, and have compared paintings generated with and without the use of salience information.

The derivation of implicit stroke ordering from image salience, is a unique contribution of our method. Many existing 2D techniques use either user interaction[2], or a combination of *a priori* knowledge and non-determinism[8] to decide stroke ordering. Whilst the use of non-determinism can contribute to the 'natural' look of an image, it can introduce problems when extending a technique to produce temporally coherent painterly animations. Preliminary tests have shown our definition of salience to be exhibit high temporal coherence across frames of moving video. Intensity gradient information also retains high coherence from frame to frame. Thus a possible direction of future work could be to use these properties to extend our technique to the generation of coherent painterly animations.

We have found our measure of salience to be surpris-

ingly general, and suitable for our purpose of painterly rendering. However, our measure is simple and enhancements to its definition form an interesting direction for future work. Improved results might also accrue were brush strokes to explicitly follow edges, rather than rely on emergent properties for the coherence of painted edges. Additionally, Elder and Zucker[6] propose a method of edge detection, which can determine the location of edges and the optimal scale at which those edges exist. It may be possible to couple this more sophisticated edge information with salience, for like Elder and Zucker we use Gaussian derivatives to compute measures over the image (recall vector $\vec{\phi}$). It may also be possible to set the superquadric shape parameter α, according to the derivative that responds most strongly. This could help to scale and orient strokes, so that they follow important edges more closely, and change their shape in response to local image charactersitics. For example, we may force $\alpha > 1$ for responses where the measure $\partial^2 G(x, y; \sigma)/\partial x \partial y$ is dominant, this would tend to produce a star-like stroke.

We have briefly described alternate uses for salience, specifically a further NPR algorithm capable of creating Cubist style compositions. This algorithm relies upon salience to identify high-level features in an image, such as eyes, to create novel abstract compositions. We believe that image salience holds an important role in the development of future NPR techniques, and warrants further study.

6 Acknowledgements

References

[1] J. P. Collomosse and P. M. Hall. Cubist-style rendering from photographs. *IEEE Transactions on Visualization and Computer Graphics*, to appear.

[2] P. Haeberli. Paint by numbers: abstract image representations. In *Computer Graphics (Proc. ACM SIGGRAPH)*, volume 24(4), pages 207–214, 1990.

[3] M. Haggerty. Almost automatic computer painting. *IEEE Computer Graphics and Applications*, pages 11–12, 1991.

[4] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. *Computer Graphics (Proc. SIGGRAPH)*, pages 453–460, 1998.

[5] A. Hertzmann and K. Perlin. Painterly rendering for video and interaction. *First International Symposium on Non-Photorealistic Animation and Rendering*, pages 7–12, June 2000.

[6] S. W. Z. J. Elder. Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 20(7):699–704, 1996.

[7] C. J. T. K. N. Walker, T. F. Cootes. Locating salient object features. In *BMVC Proceedings*, volume 2, pages 557–567, 1998.

[8] P. Litwinowicz. Processing images and video for an impressionist effect. In *Proc. SIGGRAPH, Annual Conference Series*, pages 407–414, Los Angeles, USA, 1997.

[9] B. J. Meier. Painterly rendering for animation. In *Proc. SIGGRAPH, Annual Conference Series*, pages 477–484, New Orleans, USA, 1996.

[10] R. M. P. Hall, D. Marshall. Merging and splitting eigenspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(22):1024–1049, September 2000.

[11] M. Shiraishi and Y. Yamaguchi. An algorithm for automatic painterly rendering based on local source image approximation. In *First International Symposium on Non-Photorealistic Animation and Rendering*, Anncey, France, 2000.

[12] S. Treavett and M. Chen. Statistical techniques for the automated synthesis of non-photorealistic images. In *Proc. 15th Eurographics Conference*, March 1997.