

International Journal on Artificial Intelligence Tools
© World Scientific Publishing Company

Saliency-adaptive Painterly Rendering using Genetic Search

J. P. COLLOMOSSE

*Department of Computer Science,
University of Bath,
Bath, BA2 7AY,
England.
jpc@cs.bath.ac.uk*

P. M. HALL

*Department of Computer Science,
University of Bath,
Bath, BA2 7AY,
England.
pmh@cs.bath.ac.uk*

Received (Day Month Year)

Revised (Day Month Year)

Accepted (Day Month Year)

We present a new non-photorealistic rendering (NPR) algorithm for rendering photographs in an impasto painterly style. We observe that most existing image-based NPR algorithms operate in a spatially local manner, typically as non-linear image filters seeking to preserve edges and other high-frequency content. By contrast, we argue that figurative artworks are *saliency maps*, and develop a novel painting algorithm that uses a genetic algorithm (GA) to search the space of possible paintings for a given image, so approaching an “optimal” artwork in which salient detail is conserved and non-salient detail is attenuated. Differential rendering styles are also possible by varying stroke style according to the classification of salient artifacts encountered, for example edges or ridges. We demonstrate the results of our technique on a wide range of images, illustrating both the improved control over level of detail due to our saliency adaptive painting approach, and the benefits gained by subsequent relaxation of the painting using the GA.

Keywords: Genetic Algorithms; NPR; Painterly Rendering; Saliency.

1. Introduction

Paintings are abstractions of photorealistic scenes in which salient elements are emphasised. In the words of art historian E.H. Gombrich, “works of art are not mirrors”¹ — artists commonly paint to capture the structure and elements of the scene that they consider to be important; remaining detail is abstracted away in some differential style. This *differential level of emphasis* is evident in all artwork, from the sketches of young children to works of historical importance.

Processing images into artwork remains an active area of research within the field of non-photorealistic rendering (NPR). We present a novel NPR technique for rendering images in an impasto painterly style. Our approach contrasts with those before us in that we seek to emulate the aforementioned *differential emphasis* practised by artists — automatically identifying salient regions in the image and concentrating painting detail there. This work builds upon a previous pilot study by the authors ² which demonstrates that ordering the placement of virtual brush strokes with respect to salience can enhance both accuracy and sense of composition within a painterly rendering. Here we build upon that simple, single-pass approach to propose two new technical contributions:

- A novel definition of salience that can be trained to select features interesting to an individual user, and which performs global analysis to simultaneously filter and classify low-level features of interest (for example to detect edges, ridges and corners).
- A novel salience-based approach to painting which uses a genetic algorithm (GA) to search the space of possible paintings, and so locate the optimal painting for a given photograph. A painting is deemed “better” if its level of detail coincides more closely with the salience magnitude of the original image, resulting in conservation of salient detail and abstraction of non-salient detail.

Although we are not the first to propose relaxation approaches to painting ^{3,4}, our approach is novel in that we converge toward a *globally* defined minimum distance between salience and corresponding detail in the painting. Our paintings are formed by compositing curved Catmull-Rom spline brush strokes via an adaptation of the multi-scale curved stroke painterly technique proposed by Hertzmann ⁵, modified to accommodate preferential rendering with regard to image salience. Strokes are more densely placed in salient regions, then ordered and modulated to prevent strokes from non-salient areas encroaching on more salient ones. Differential rendering styles are also possible by varying stroke style according to the classification of salient artifacts, for example edges or ridges. Furthermore, we use our novel relaxation scheme to iteratively converge the rendering toward the ‘optimal’ painting for a given image. In doing so, we adapt Hertzmann’s contour tracing algorithm to account for the influence of noise, present in any real image. As a consequence, post-relaxation strokes tightly match the contours of salient objects and non-salient details is attenuated. We demonstrate the results of our painterly technique on a range of images, illustrating the benefits of rendering with regard to salience and the improvements gained by subsequent relaxation of the painting.



Fig. 1. Detail of a painting generated using our saliency-based relaxation technique, sampled from the fittest individual within the 80th generation of paintings. A collection of paintings and their respective source images can be downloaded online at <http://www.cs.bath.ac.uk/vision/geneticpaint>

1.1. Related Work and Context

The development of automated painterly renderers arguably began to gain momentum with Haeberli's semi-automatic paint systems⁶. These allowed users to interactively generate impressionist style 'paintings' by creating brush strokes, the colour and orientation of which were determined by point-sampling a reference image. As applications increasingly demanded automation, the onus shifted away from user guidance toward automated heuristics to guide the placement of strokes. In Haeberli's system a user might choose a finer brush to render salient areas of an image; to mimic this behaviour automatically is more difficult. Early commercial attempts to derive automatic renderers from Haeberli's systems were based for the most part upon pseudo-randomness⁷. Data dependent approaches were later presented, driven by heuristics based on local image processing techniques which automatically estimated stroke attributes such as scale or orientation. Litwinowicz⁸ employed short, linear paint strokes, which were clipped to thresholded edges. Treavett and Chen⁹ proposed using local statistical measures to determine stroke parameters, aligning strokes along axes of minimum intensity variance. A similar approach taking into account chromatic variance was later described by Shirashi and Yamaguchi¹⁰. Hertzmann proposed a layered, coarse-to-fine approach to painting⁵ and was the first to automatically place curved (β -spline) strokes rather than dabs of paint. The stroke placement algorithm we describe here is most closely related to Hertzmann's work. Work by Gooch *et al.*¹¹ also uses curved strokes fitted to skeletons extracted from local connected regions of homogeneous luminance. Relaxation approaches to painting have also been examined, and although paint by relaxation systems were discussed in⁶, the first algorithmic solution is described in⁴. In this system active contours (snakes) are used to seek a painting preserving the maximum level of fine

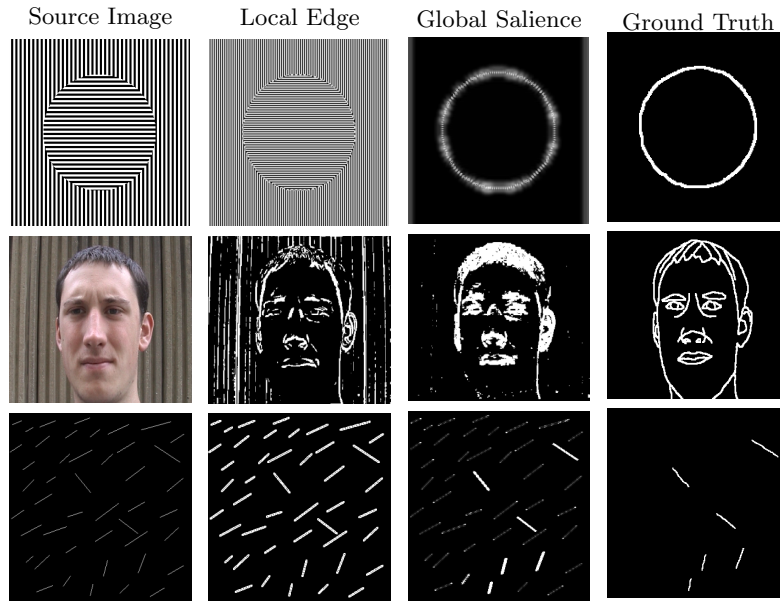


Fig. 2. Left: Examples of images edge detected, saliency mapped, and a hand-sketched ground truth. We observe that the global, rarity based saliency maps are qualitatively closer to sketches, and can “pick out” the circle and face where local methods such as edge detection fail.

detail from the original image. As with all active contour implementations success is sensitive to initial contour location, due to the susceptibility of snakes to local minima ¹².

A commonality exists between all of these algorithms; the attributes of each brush stroke are determined independently, by heuristics that analyse small pixel neighbourhoods local to that stroke’s position. Rendering is, in this sense, a *spatially local* process. The heuristics typically seek to convey the impression of an artistic style whilst preserving content such as edges, and other artifacts contributing to the upper frequencies of the Fourier spectrum. Indeed, existing relaxation-based painting algorithms ^{3,4} actively seek to maximally conserve high-frequency content from the original image. Measures of variance ^{9,10}, or more commonly, simple edge detectors (such as Sobel) ^{8,5} drive these heuristics. This results in a painting in which all fine detail is emphasised, rather than only the *salient* detail. Arguably this disparity contributes to the undesirable impression that such paintings are of machine rather than natural origin. In Fig. 2 (left) we demonstrate that not all fine scale artifacts are salient; indeed in these images, salient and non-salient artifacts are of similar scale (that is, we require windows of comparable size to detect them reliably). Such examples make the case for some other measure of saliency incontrovertible. When one speaks of the saliency of image regions, one implicitly speaks of the importance of those regions relative to the image as a whole. It follows that *global* image anal-

ysis is a prerequisite to saliency determination, rather than restricting attention to *spatially local* image properties.

The aim of our work is to automatically control level of detail in NPR. Some recent techniques approach this problem by appealing to interaction. DeCarlo and Santella¹³ proposed an NPR system that enabled visual emphasis to be varied interactively using gaze trackers. Image masks or “weight maps”, specified manually or *a priori*, have also been used by Hertzmann⁴ and Bangham *et al*¹⁴ to interactively control level of detail. However the problem of adaptively controlling painting emphasis remains; this paper presents a solution.

2. Determining Image Saliency

Saliency is subjective; different faces photographed in a crowd will hold different levels of saliency to friends or strangers. User training is one way in which subjectivity can be conveyed to an automated measure, although current Computer Vision restricts general analysis to a lower level of abstraction than this example.

We wish to automatically estimate the perceptual saliency of images. That is, produce a mapping from a colour image to a scalar field in which the value of any point is directly proportional to the perceived saliency of the corresponding image point. We now describe a user trainable approach to estimating this mapping, comprising three operators which respectively compute the rarity, visibility, and classification of local image artifacts. These three operators are computed independently yielding three probabilities (P_{rare} , $P_{visible}$, P_{class}) which are combined to estimate the final probability of an image artifact being salient as: $P_{salient} = P_{rare}P_{visible}P_{class}$.

We begin by describing an operator which performs unsupervised global statistical analysis to evaluate the relative rarity (P_{rare}) of image artifacts, after Walker *et al.*¹⁵ who observe that salient features are uncommon in an image (subsection 2.1). This measure, whilst simplistic, can out-perform standard edge detection in many cases (Figure 2) principally because it is a global, rather than local, measure. However, not all rare artifacts should be considered ‘salient’. In particular, we assert that salient artifacts should also be visible, and propose a second perceptually trained operator which estimates the visibility ($P_{visible}$) of image artifacts (subsection 2.2). The user may perceive certain classes of artifact, for example edges or corners, to be more salient than others. We therefore propose a third operator which users train by highlighting salient artifacts in photographs. Signals corresponding to these artifacts are clustered to produce a classifier which may be applied to artifacts in novel images in order to estimate their potential saliency (P_{class}).

This trainable saliency measure is well suited to our NPR painting application for

two reasons. First, the salience maps produced have been shown to be measurably closer to human figurative sketches of scenes than edge maps and a number of other prescriptive salience measures¹⁶. Second, the ability to estimate both the salience and the classification of image artifacts simultaneously allows us to vary stroke style according to the class of artifact encountered (Fig. 5).

2.1. Determining pixel rarity

We first describe our unsupervised technique for determining pixel rarity. The basic technique is to model the statistical distribution of a set of measures locally associated with each pixel, and to isolate the outliers of this distribution.

For a given pixel $\mathbf{p} = (i, j)^T$ we consider a series of rings of radius ρ , each centred at $(i, j)^T$. We uniformly sample the image around each ring's circumference at angular positions θ , hence obtaining a discrete colour signal $\mathbf{x}(\mathbf{p}) = (\rho, \theta) \in \mathbb{R}^3$; colours are in RGB space. This signal is rewritten as a column vector. We have found a sampling rate of 16, and values of ρ ranging from 1 to 3 in increments of 0.5, to yield good results in subsequent processing. For an image of M pixels we have M vectors $\mathbf{x}(\cdot) \in \mathbb{R}^n$, where for us $n = 16 \times 3 \times 3$. We assume these points are Gaussian distributed, which we represent using an eigenmodel; a simple and convenient model that works acceptably well in practice. The eigenmodel is computed incrementally due to the large size of the data set. The eigenmodel provides a sample mean μ and a covariance matrix C ; the product of the set of eigenvectors and corresponding eigenvalues. An eigenmodel allows us to compute the squared Mahalanobis distance of any point $\mathbf{x}(\cdot) \in \mathbb{R}^n$:

$$d^2(\mathbf{x}(\cdot)) = (\mathbf{x}(\cdot) - \mu)^T \mathbf{C}^{-1} (\mathbf{x}(\cdot) - \mu) \quad (1)$$

We compute the Mahalanobis distance $d(\cdot)$ for all pixels \mathcal{P} in the image. The probability of an individual pixel $q \in \mathcal{P}$ being rare is then written as the quotient:

$$\mathcal{Q} = \{\mathbf{r} : \mathbf{d}(\mathbf{x}(\mathbf{r})) \leq \mathbf{d}(\mathbf{x}(\mathbf{q})) \wedge \mathbf{r}, \mathbf{q} \in \mathcal{P}\} \quad (2)$$

$$P_{rare}(\mathbf{q}) = \frac{\sum_{\mathbf{p} \in \mathcal{Q}} d(\mathbf{x}(\mathbf{p}))}{\sum_{\forall \mathbf{p} \in \mathcal{P}} d(\mathbf{x}(\mathbf{p}))} \quad (3)$$

This measures the fraction of the sample density which is less rare than the pixel \mathbf{q} .

2.2. Classification of image artifacts

We now introduce a degree of subjectivity by allowing users to train the system to identify certain classes of low-level artifact as potentially salient.

For a given pixel \mathbf{p} , we sample the image in an identical manner to that used for determining pixel rarity. However, we treat each ring separately, and so consider

the classification of the colour signal $\mathbf{c}(\theta)$ at constant ρ (this turns out to be more stable than considering the disc as a whole). We form a feature vector by first differentiating $\mathbf{c}(\theta)$, using Euclidean distance in RGB space, to obtain a periodic scalar signal $\mathbf{y}(\theta)$ (Figure 3).

We now take the absolute value of the Fourier components $|F[\mathbf{y}(\theta)]|$, normalise to unit power, and drop the d.c. (zeroth) component. Thus for a given $\mathbf{y}(\theta)$ we compute a feature as:

$$\mathbf{f}(\omega) = \frac{|F[\mathbf{y}(\theta)]|}{(\sum_{\theta} |\mathbf{y}(\theta)|^2)^{\frac{1}{2}}} \quad (4)$$

$$\mathbf{f}(\omega) \leftarrow \mathbf{f}(\omega) \setminus \mathbf{f}(\mathbf{0}) \quad (5)$$

Removing the d.c. component is equivalent to subtracting the mean, which makes this feature vector invariant to linear colour shifts. It is also invariant to orientation and mirroring. Thus $\mathbf{c}(\theta)$, $\mathbf{c}(\theta) + \alpha$, $\mathbf{c}(\theta + \beta)$ all map to same point in feature space. The system has proved to be robust to more general colour scalings, $\gamma\mathbf{c}(\theta)$, but cannot be invariant (suppose $\gamma = 0$). It is these properties that principally motivate our choice of circular sampling, since the classification of salient artifacts (for example, edges) should be invariant with respect to a cyclic shift of the signal. This contrasts with features based on standard derivative forms, in which edge signals, say, are thinly distributed across feature space.

2.2.1. Training and Classification

Training is a supervised process that occurs over several images, and requires the user to interactively highlight artifacts they regard as salient during a pre-processing step. Moreover, the user may choose a number of classes of artifact (such as edge, ridge, or corner), and identify a class label with each artifact they highlight. Training therefore results in multiple sets of artifacts, one set per class.

To build the classifier we convert each artifact in a given set, k say, into a feature vector as previously described. We then estimate the class conditional density $p(\mathbf{f}|\mathbf{k})$ for that set of features using a Gaussian Mixture Model (GMM), fitted using Expectation Maximisation¹⁷. We estimate a prior, $p(k)$, as the expected number of points — the ratio of the number elements in the given set to the number of points in all sets. This enables us to compute the posterior likelihood $p(k|\mathbf{f})$ by appeal to Bayes theorem:

$$p(k|\mathbf{y}) = \frac{\mathbf{p}(\mathbf{y}|\mathbf{k})\mathbf{p}(\mathbf{y})}{\sum_{\mathbf{j}=1}^N \mathbf{p}(\mathbf{y}|\mathbf{j})\mathbf{p}(\mathbf{j})} \quad (6)$$

During painting, classification of a pixel begins by sampling to obtain a new artifact. This is converted to a feature vector and the above probability vector is computed

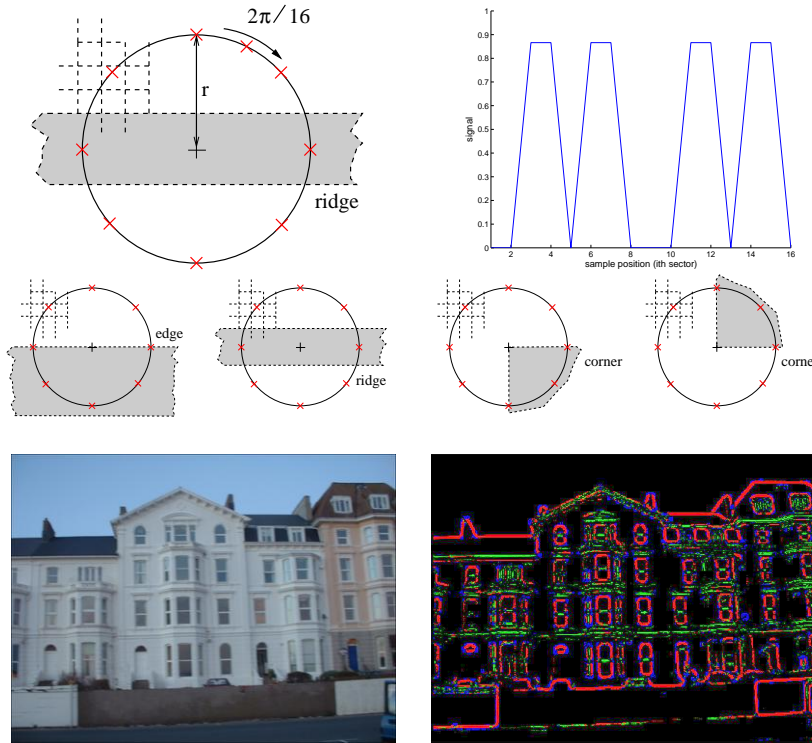


Fig. 3. Circular descriptors are used to create signals (top row) from points specified as salient by the user, which are then clustered in a feature space. Features such as ridges, edges and corners (second row) create distinctive spectral signals (third row) which may be used to determine not only the salience of a point, but also its classification type. Bottom row: a photograph and corresponding salience map; edges in red, ridges in green and corners in blue.

(one element per class). Assuming independence between classes is convenient, allowing us to simply add vector elements to estimate the probability that an artifact belongs to a subset of classes. For each classified pixel we therefore have a probability $p(k|\mathbf{y})$ of membership to each of the trained classes, and compute P_{class} as the maximum value over all $p(k|\mathbf{y})$.

2.2.2. Selection of Scale for Classification

The above approach classifies artifacts at a constant ρ , and so at constant scale. However classification can vary over scale. For example, an artifact classified as an edge at small scales might be classified a ridge at larger scales; in such cases one would arguably prefer the final classification to be ‘ridge’. By contrast corners remain relatively stable over scale, and it transpires that a range of heuristics exist for other such combinations. To opt for the most stable classification over scale is therefore insufficient, and only to hard code heuristics specific to edges, ridges etc.

is also a poor solution since these are but examples of more general features that users may identify.

Our strategy is to perform the classification of a given point at several values of ρ ; again using the range 1 to 3 at increments of 0.5. At each scale we obtain a posterior probability vector $p(k|\mathbf{y})$, and concatenate these to form a column vector (a point in a higher-dimensional space that now encapsulates scale information). Since we know the user supervised classification of each point we may again perform clustering of salient feature classes by fitting GMMs in this scale-dependent space. The aforementioned ‘heuristics’ for classification are thus implicitly learnt during the training process.

2.3. Determining Visibility

Our final operator estimates the probability that a local image window contains a perceptually visible signal. We have empirically measured the *just noticeable difference* (JND) between colours in RGB format. We assume that for each RGB colour \mathbf{c} there is distance $\tau(\mathbf{c})$, also in RGB space. Together the colour and the distance specify a sphere of RGB colours $(\mathbf{c}, \tau(\mathbf{c}))$. No colour interior to the surface of the sphere can be perceptually discriminated from the centre colour. The distance $\tau(\mathbf{c})$ is one JND at the colour \mathbf{c} . The sphere radius varies depending on experimental conditions, and after several experimental trials τ emerges as the mean radius accompanied by an associated standard deviation σ . Whilst this is a simple colour model (an ellipsoid might better model JND surfaces) we have found it to be satisfactory for our purposes. Similar distance metrics for luminance are also described in ¹⁹.

To measure the visibility of artifacts local to a point $\mathbf{p} = (\mathbf{i}, \mathbf{j})^T$, we sample in a manner identical to Section 2.1 and then obtain the total colour change $d\mathbf{c}(\rho, \theta)$ via the differential magnitude written as:

$$d\mathbf{c}(\rho, \theta) = \left| \left(\frac{\delta\mathbf{c}(\rho, \theta)}{\delta\rho} \right)^2 + \left(\frac{\delta\mathbf{c}(\rho, \theta)}{\rho\delta\theta} \right)^2 \right|^{1/2} \quad (7)$$

where $\mathbf{c}(\rho, \theta)$ returns the RGB value of the image at coordinates (ρ, θ) relative to \mathbf{p} . We compute the probability $\phi(\cdot)$ that this change is visible as:

$$\phi(\rho, \theta) = \text{erf}(d\mathbf{c}(\rho, \theta) - \tau(\mathbf{c}(\rho, \theta))/\sigma(\mathbf{c}(\rho, \theta))) \quad (8)$$

where $\tau(\cdot)$ and $\sigma(\cdot)$ are the JND and its deviation for the colour sample at $\mathbf{c}(\rho, \theta)$ in the local window. We reason that if a signal is visible in any ring, then the whole disc should be regarded as having at least that ring’s visibility. We write the probability of the disc being visible as:

$$P_{\text{visible}} = \sum_{\rho=1}^{\rho_{\text{max}}} \max(\phi(\rho, \theta)) \quad (9)$$

3. Painting as a Search

Our observations of artists lead us to assert that the level of detail in a painting should closely correlate with the salience map of its source image. In this sense, the *optimality criterion* for our paintings is a measure of the strength of this correlation (defined in equation 10). We treat the painting process as a search for the “optimal” painting under this definition. Our search strategy is genetic algorithm (GA) based. When one considers the abstraction of a painting as an ordered list of strokes ⁶ (comprising control points, thickness, etc. with colour as a data dependent function of these), the space of possible paintings for a given source image is very high dimensional, and our optimality criterion makes this space extremely turbulent. Stochastic searches that model evolutionary processes, such as GAs ²⁰, are often cited among the best search strategies in such situations; large regions of problem space can be covered quickly, and local minima more likely to be avoided ^{21,22}.

Our algorithm accepts as input a source image I ; paintings derived from I are points in our search space. We begin by applying the salience measure to I ; obtaining both a salience map and a classification probability for each pixel. An intensity gradient image is also computed using Gaussian derivatives, from which a gradient direction field is obtained. With this pre-preprocessing complete, we initialise a fixed size population of individuals. Each individual is single point in our search space, represented by an ordered list of strokes that, when rendered, produces a painting from I . Having initialised the population, the iterative search process begins. We now describe the initialisation and iteration stages of the search in detail.

3.1. Initialising the Painting Population

We initialise the search by creating an initial population of 50 paintings, each derived from the source image via a stochastic process. This population limit was determined empirically, and is discussed further in section 5. We now describe this derivation process for a single painting.

Our paintings are formed by compositing curved spline strokes on a virtual canvas. We choose piecewise Catmull-Rom splines for ease of control since, unlike β -splines (used in ^{5,11}), control points are interpolated. We begin by placing seed points on the canvas, from which strokes are subsequently ‘grown’ bidirectionally. As a heuristic we make provision for a stroke to be seeded at every other pixel; seeds are then scattered stochastically, with a bias toward placement of seeds in more salient regions. In practice we scatter 95% of strokes in this manner, the remaining 5% are scattered uniformly to fill holes appearing in areas of relatively low salience.

3.1.1. Bidirectional Stroke Growth

Strokes are grown to extend bidirectionally from seed points. Each end grows independently until it is halted by one or more preset criteria. Growth proceeds in a manner similar to Hertzmann’s algorithm⁵ in that we hop between pixels in the direction tangential to their intensity gradient. The history of visited pixels forms the control points for the spline stroke. Noise forms a component of any real image, and in particular any direction estimation is better regarded as being sampled from a stochastic distribution (Figure 4). This frustrates single-pass painterly NPR algorithms⁵, introducing inaccuracy into the stroke fittings process and resulting in “loose and sketchy” paintings. We have observed that this noise obeys the central limit theorem (see²³ for experimental details), and so model this distribution as a zero centred Gaussian, $G(0, \sigma)$; we determine σ empirically (subsection 3.1.2). Given a locally optimal direction estimate θ we select a hop direction by adding Gaussian noise $G(0, \sigma)$. The magnitude of the hop is also Gaussian distributed; on this occasion $G(\mu', \sigma')$, both parameters being inversely proportional to local saliency. The growth of a stroke end is halted when either the curvature between adjacent pixels, or the distance (in *JND* space) between the colour of the pixel to be appended and the mean colour of visited pixels, exceeds a threshold.

This method initially yields a sub-optimal trajectory for the stroke with respect to our measure in Section 1. For a ‘loose and sketchy’ painting this is often desirable (see Figure 5), but for tighter paintings stroke trajectories must be closer to the optimal. The degrees of freedom resulting from each of the many hops combine to create a range of stroke loci, at least one of which will result in the maximal conservation of salient detail. The combination of these optimally positioned strokes

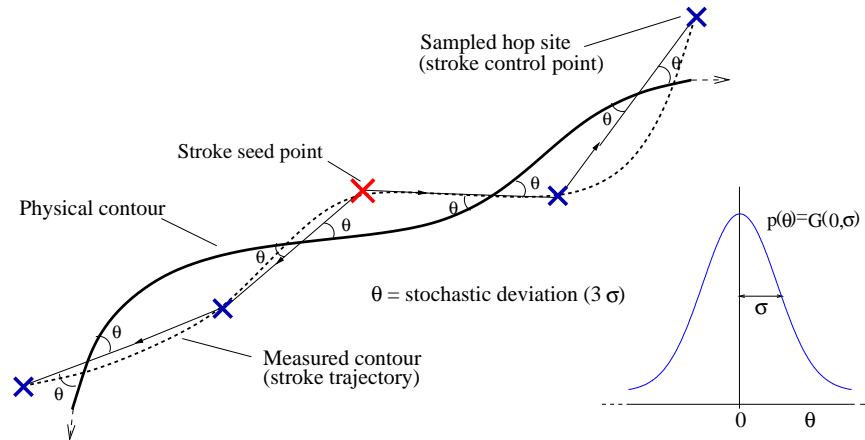


Fig. 4. Illustrating the stochastic growth of strokes from a seed pixel. We choose strokes with hop sites which minimise our objective function, under the constraint that hop angles are drawn from the distribution $p(\theta) = G(0, \sigma)$.

comprises the optimal painting, and it is by means of breeding the fittest paintings to create successively superior renderings, that we search for such a painting via GA relaxation in subsection 3.2. Our relaxation strategy is thus able to approach more globally optimal stroke trajectories, and these can out-perform trajectories based purely on local estimates of direction.

3.1.2. *Calibration for image noise*

The choice of σ significantly influences the stroke growth and relaxation process. A value of zero forces degeneration to a loose painterly system, as the degrees of freedom for variation in stroke placement are restricted. Similarly, if σ is too large, the relaxation process will be unnecessarily lengthened and also may introduce unnecessary local minima. We propose a one time user calibration process to select this σ , typically performed during the training step of the perceptual salience measure.

The user is asked to draw around sample image regions where direction of image gradient is perceived to be equal; i.e. along which they would paint strokes of similar orientation. This results in several samples of the image gradient from which we may compute angles. We have observed the natural distribution of these values to be Gaussian and take the mean angle $\mu(\cdot)$ as the common tangential angle. Similarly, we compute the unbiased standard deviation of the set of measured tangential angles which subsequently becomes the σ parameter for stroke growth. We assume σ to be equal for all angles.

We typically obtain very similar σ values for similar imaging devices, which allows us to perform this calibration very infrequently. A typical σ ranges from around 2 to 5 degrees, with the larger deviations being attributed to digital camera devices (possibly as artifacts of low CCD quality or JPEG compression). This variation allows between 12 and 30 degrees of variation per hop which, given the number of hops per stroke, is a wide range of stroke loci. This measurements add credence to our argument for the need of a relaxation process taking into account image noise; potentially large variations in stroke placement due to uncompensated image noise are likely to produce inaccurate stroke placements in single-pass painterly systems ^{8,5,10,11}.

3.1.3. *Rendering and Differential Styles*

At this stage we may render one of the paintings in our initial population to produce a “loose and sketchy” painting (Fig. 5). Alternatively we may proceed to the iterative search stage of subsection 3.2 to locate a more optimal painting — each iteration also requires paintings to be rendered to evaluate fitness. We now describe how paintings are formed from individuals in the population.

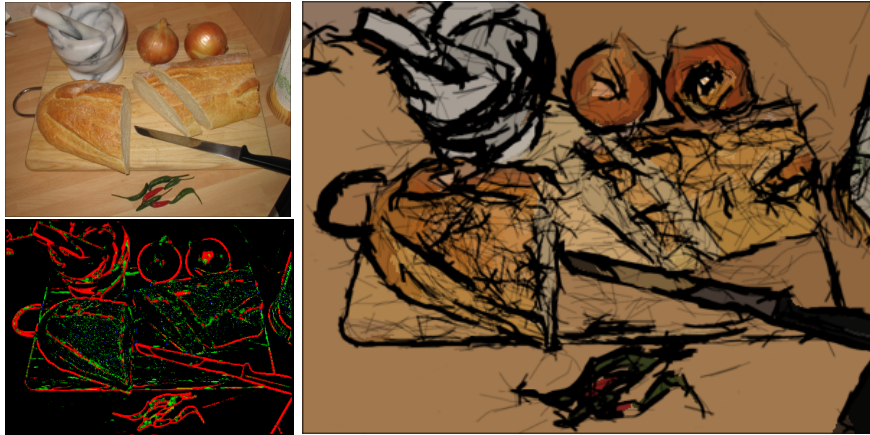


Fig. 5. Left: a still-life composition and corresponding salience map. Right: a loose and sketchy painting, exhibiting differential stroke style determined by local feature classification. Edges are drawn with hard, precise thick strokes; ridges with a multitude of light, inaccurate strokes. Rendered prior to the relaxation step of subsection 3.2.

Stroke rendering attributes are set automatically as a function of *stroke salience*, taken as the mean salience over each control point. By default, stroke thickness is set inversely proportional to salience. Stroke colour is uniform and set according to the mean of all pixels encompassed in the footprint of the thick paint stroke. During rendering, strokes of least salience are laid down first, with more salient strokes being painted later. This prevents strokes from non-salient regions encroaching upon salient areas of the painting.

The ability of our salience measure to differentiate between classes of salient feature also enables us to paint in context dependent styles. For example, we have described how we may discriminate between artifacts such as edges and ridges (Section 2.2). In Figure 5 we give an example of a painting generated by our system, in which the classification probability of a feature is used as a parameter to interpolate between three rendering styles (parameter presets) *flat*, *edge* and *ridge*. For the flat preset, rendering takes the default form described in the previous paragraph. For edges and ridges, the luminance of strokes is heavily weighted to create dark outline strokes. In the case of edges, thickness of strokes is also boosted to create thick outlines — whilst with ridges the thickness is greatly reduced to produce thin wispy strokes. The σ value for ridges is also boosted to reduce accuracy and produce ‘sketchy’ strokes. Since these preset rendering parameters (thickness, luminance decay, etc.) all vary by continuous multiplicative factors, interpolation between the presets according to the classification probability vector is straightforward. There is additional semantic value in that we render ridges as single strokes, rather than as two edge strokes. To the best of our knowledge, rendering in differential styles via an automated heuristic is a novel contribution to NPR.

3.2. Iterative Relaxation by GA

Genetic algorithms simulate the process of natural selection by breeding successive generations of individuals through the processes of cross-over, fitness-proportionate reproduction and mutation. In our implementation such individuals are paintings; their genomes being ordered lists of strokes and their associated attributes. We now describe a single iteration of the GA search, which is repeated until the improvements gained over the previous few generations are marginal (the change in both average and maximum population fitness over a sliding time window fall below a threshold). A schematic of the iterative process is given in Figure 6.

3.2.1. Fitness and Selection

The entire population is rendered, and edge maps of each painting are produced using by convolution with Gaussian derivatives, which serve as a quantitative measure of local fine detail. The generated maps are then compared to a precomputed salience map of the source image. The mean squared error (MSE) between maps is used as the basis for determining the fitness $F(\cdot)$ of a particular painting; the lower the MSE, the better the painting:

$$F(I, \psi) = 1 - \frac{1}{N} \sum |S(I) - E(\Psi(I, \psi))|^2 \quad (10)$$

The summation is over all N pixels in source image I . Ψ is our painterly process, which produces a rendering from I and a particular ordered list of strokes ψ corresponding to an individual in the population. Function $S(\cdot)$ signifies the salience mapping process of Section 2, and $E(\cdot)$ the process of convolution with Gaussian derivatives.

In this manner, individuals in the population are ranked according to fitness. The bottom 10% are culled, and the top 10% pass to the next generation. The latter heuristic promotes convergence; the fittest individual in successive generations must be at least as fit as those in the past. The middle 80% are used to produce the remainder of the next generation. Two individuals are selected stochastically with a bias to fitness, and bred via cross-over to produce a novel offspring for the successive generation. This process repeats until the population count of the new generation equals that of the current.

3.2.2. Cross-over

Two difference images, A and B , are produced by subtracting the edge maps of the parents from the salience map of the original image, then taking the absolute value of the result. By computing the binary image $A > B$, and likewise $B > A$, we are able to determine which pixels in one parent contribute toward the fitness criterion to a greater degree than those in the other. Since the primitives of our paintings are thick brush strokes rather than single pixels, we perform several binary dilations

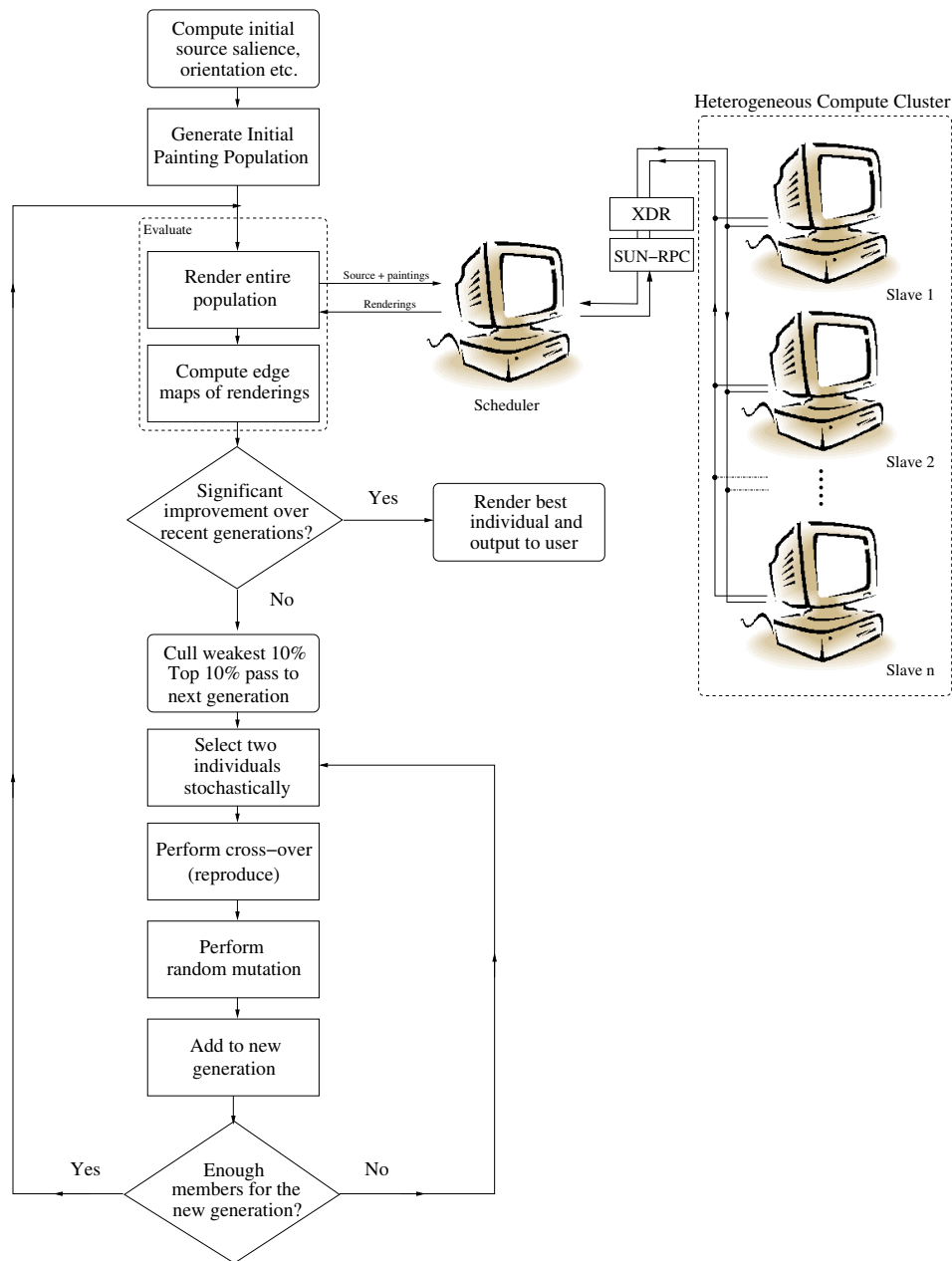


Fig. 6. Illustrating flow of control in the genetic algorithm. The population evaluation stage is inherently parallel and rendering is farmed out to a distributed compute cluster.

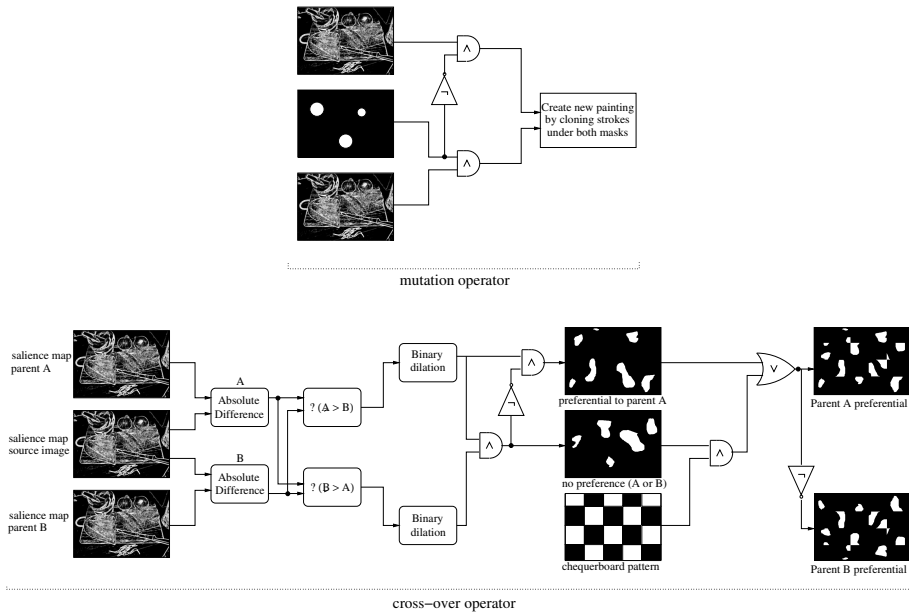


Fig. 7. Genetic operators: The cross-over and mutation operators used during the relaxation process (images are mock-ups for illustrative purposes only)

to both images to mark small regions local to these “fitter” pixels as desirable. A binary AND operation between the dilated images yields mutually preferred regions (i.e. where $A = B$). We mask these conflicting regions with a coarse chequerboard texture (of random scale and phase offset) to decide between parents in an arbitrary fashion. Finally, strokes seeded within the set regions in each parent’s mask are cloned to create a new offspring. Figure 7 summaries this process.

3.2.3. Mutation

When a bred individual passes to a successive generation it is subjected to a random mutation. A new “temporary” painting is synthesised (though never rendered), and a binary mask produced containing several small discs scattered within it. The number, location and radius of the discs are governed by random variates. Strokes seeded within set regions of the binary mask are substituted for those in the temporary painting; the temporary painting is then discarded. In our implementation large areas of mutation are relatively rare, averaging around 4% of the canvas area.

3.2.4. Termination

The relaxation process runs until the improvements gained over the previous few generations become marginal (the change in both average and maximum population

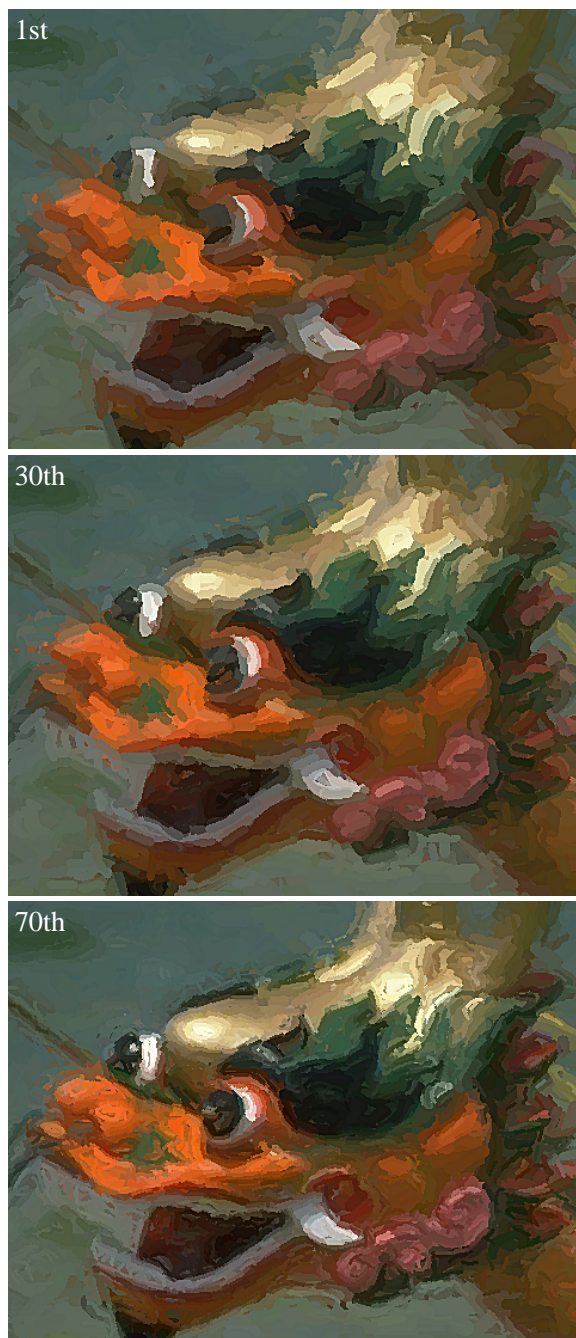


Fig. 8. Relaxation by genetic algorithm. Detail in the salient region of the 'dragon' painting sampled from the fittest individual in the 1st, 30th and 70th generation of the relaxation process. Strokes converge to tightly match contours in salient regions of the image thus conserving salient detail.

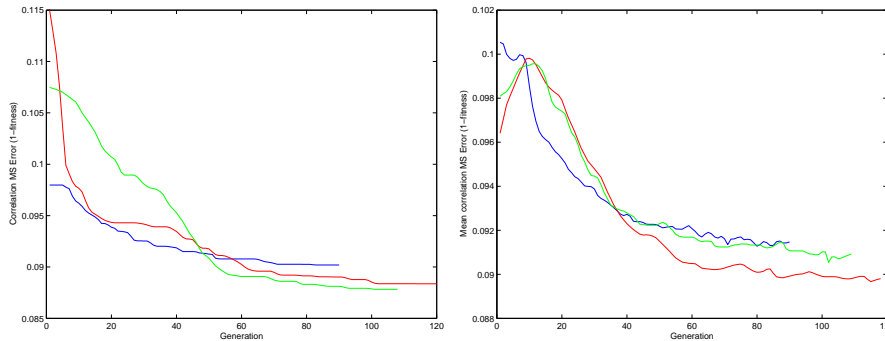


Fig. 9. Left: Three runs of the relaxation process; blue corresponds to the model (Fig. 10), red the dragon (Fig. 1) and green the truck (Fig. 12). MSE of the fittest individual is plotted against time. Middle: MSE averaged over each generation

fitness over sliding time window fall below a threshold Δ), at which point the search has settled into a minima of sufficient extent in the problem space that escape is unlikely. The fittest individual in the current population is then rendered and output to the user. Typically executions run for around one to two hundred iterations for values of σ between two and five degrees, which we have found to be the range of standard deviations for image noise (see Section 3.1.2). Forcing larger values of σ can result in convergence but, we observe, at the cost of an exponential increase in execution time.

3.2.5. Implementation Notes

In practice, evaluation is the most lengthy part of the process and the rendering step is farmed out to several machines concurrently. In our implementation we distribute and receive paintings via the Sun RPC interface, using XDR to communicate over a small heterogeneous (Pentium III/UltraSPARC) compute cluster. The typical time to render a 50 painting generation at high (1024×768) resolution is approximately 15 minutes over 6 workstations. Relaxation of the painting can therefore take in the order of hours, but significant improvements in stroke placement can be achieved as can be seen in Figure 8. The overhead of our task scheduler is low, and processing time falls approximately linearly as machines of similar specification are added to the cluster.

4. Results

We have rendered a number of paintings to demonstrate application of our algorithm in Figures 10, 12-15; the reader is also referred back to the dragon (Figure 1) and sketchy still-life (Figure 5) paintings, presented *in situ*. As a note, we have found that running the paintings through a standard sharpening filter can assist presentation of our paintings on the printed page, and have applied such a filter to

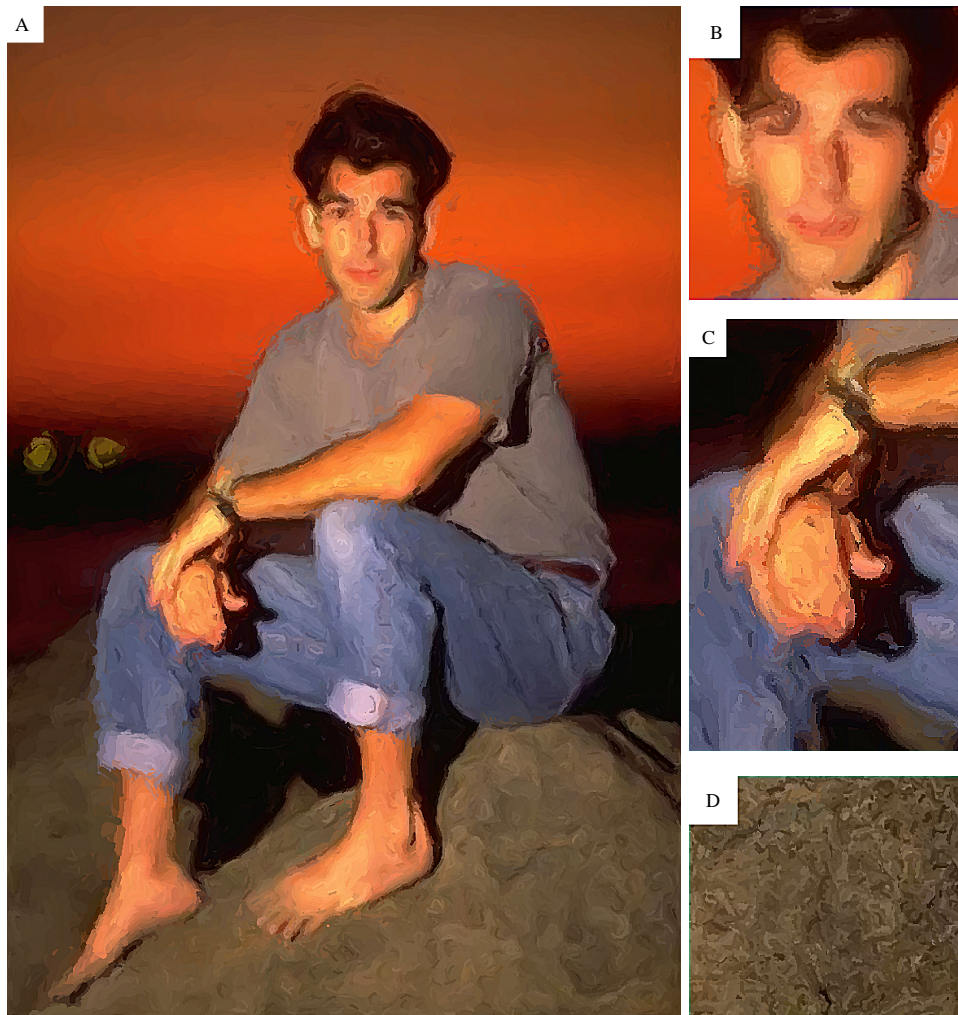


Fig. 10. Man on rock: (a) final painting after convergence using our proposed method, close-up of hands in (c). (b) example of the face rendered with insufficient emphasis (d) example of rock texture rendered with too great an emphasis. Please refer to the text of Section 4 for full explanation of (b) and (d), and how our salience adaptive painting avoids such difficulties.

all paintings presented in this paper.

The painting of the model in Figure 10a converged after 92 generations. Thin precise strokes have been painted along salient edges, while ridges and flats have been painted with coarser strokes. We can see that non-salient high-frequency texture on the rock has been abstracted away, yet tight precise strokes have been used to emphasise salient contours of the face. In the original image the high frequency detail in both regions is of similar scale and edge magnitude; existing painterly

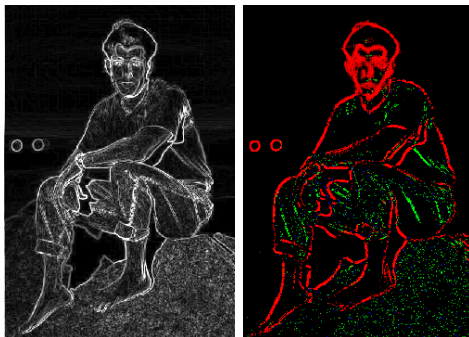


Fig. 11. Comparison of the Sobel edge magnitude field (left) and our saliency map (right), corresponding to the painting of Figure 10. Salient edges (red) are discriminated from non-salient high frequency detail (ridges in green, corners in blue) guiding emphasis in the final painting. Such distinction can not be made locally using variance or Sobel measures.

techniques would, by contrast, assign both regions equal emphasis. With current techniques, one might globally increase the kernel scale of a low-pass filter⁵ or raise thresholds on Sobel edge magnitude⁸ to reduce emphasis on the rock (Figure 10c). However this would cause a similar drop in the level of detail on the face (Figure 10b). Conversely, by admitting detail on the face one would unduly emphasise the rock (Figure 10d). In our method, we automatically differentiate between such regions using a perceptual saliency map (Figure 11) — contrast this with the Sobel edge field in the same figure, in which no distinction between the aforementioned regions can be made.

We present a still-life in Figure 13 which achieved convergence after 110 generations. Inset within this figure we present a similar painting prior to relaxation, demonstrating differential rendering style as strokes with a high probability of being edges are darkened to give the effect of a holding line. Further examples of level of detail adaptation to saliency are given in Figure 12. In region *A*, observe that the salient 'phone sign is emphasised whilst non-salient texture of the background shrubbery is not (also see Figure 14 for an enlarged, comparative example). For the purposes of demonstration we have manually altered a portion of saliency map in region *B*, causing all detail to be regarded as non-salient. Contrast stroke placement within this region with that on the remainder of the car body. Variations in style may be achieved by altering the constants of proportionality, and also thresholds on curvature and colour during stroke placement. Paintings may be afforded a more loose and sketchy feel by increasing the halting threshold Δ and so decreasing the number of relaxation iterations; essentially trading stroke placement precision for execution time. A similar trade-off could be achieved by manually decreasing the noise estimate σ .

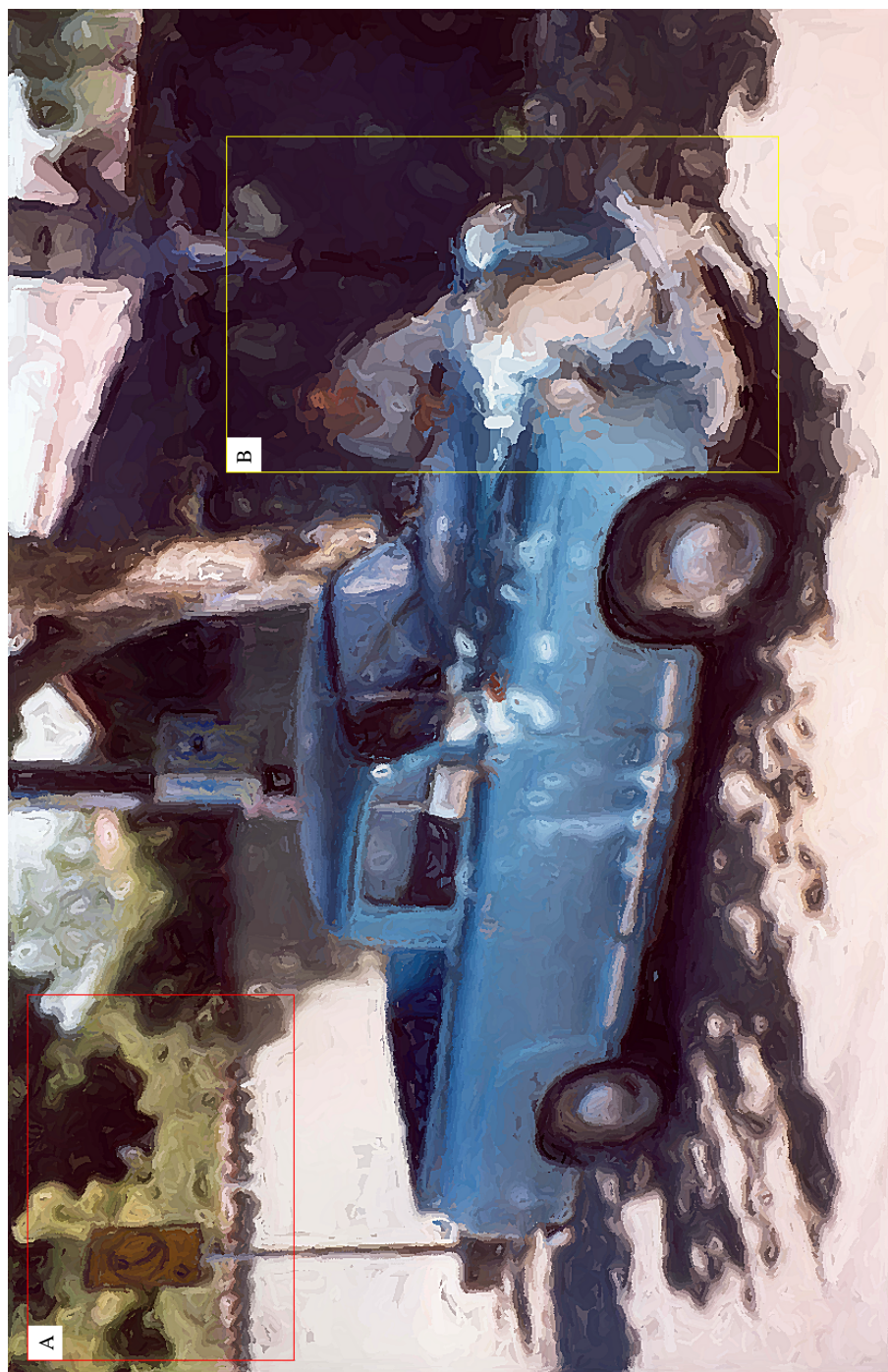


Fig. 12. Pickup truck after convergence. Observe saliency adaptive emphasis of sign against background in (a). We have manually dampened the saliency map in (b) to cause greater abstraction of detail; compare stroke placement here with the remainder of the car body. Original photo courtesy Adam Batenin.



Fig. 13. Sunflowers after convergence. Inset: a sketchy version of the sunflowers in the style of Figure 5, prior to relaxation.



Fig. 14. Detail from Figure 12, region A. Left: Section of the original photograph exhibiting non-salient background texture (shrubby) and salient foreground (sign-post). Middle: All fine detail is emphasised using an existing automatic approaches (due to Litwinowicz) which places strokes using only spatially local information. In this image, the high frequency detail of the background leaf texture has caused strokes to be clipped at edges, tending the process back toward photorealism. However it is clear that mitigating this effect, by reducing the edge threshold for clipping, will further degrade salient detail on the sign. Right: Using our adaptive approach, salient detail is conserved, and non-salient detail is abstracted away.



Fig. 15. Bath Abbey after 110 generations of the relaxation process. The darker strokes outlining the arch and other salient edges are generated by interpolating between a default and ‘edge’ preset according to the probability of salient artifacts being edges (see Section 3.1.3).

5. Conclusion

We have presented a novel automatic algorithm for creating impasto style painterly renderings from photographs. The contributions of our technique are twofold. First, we propose a novel relaxation process using a genetic algorithm which results in tight accurate placement of thick curved brush strokes. Stroke placement takes into account the influence of noise, present in any real image. Second, we propose use of a novel, perceptual salience measure to drive the rendering process. This results in a painting in which salient regions are emphasised in tight precise strokes, but non-salient detail is abstracted away. This contrasts with existing painterly methods which aim to conserve all high frequency detail in the painting; arguably our approach is more in line with traditional artistic practice. Furthermore our salience measure allows us to classify the type of salient artifacts encountered and to differentiate stroke rendering according to that classification. To the best of our knowledge the variation of strokes rendering style via an automated heuristic is also a novel contribution to image-based NPR.

All of our experiments have used populations of 50 paintings per generation. We initially speculated that population level should be set in order of hundreds to create the diversity needed to relax the painting. However it transpires that although convergence still occurs with such population limits, it requires, on average, 2 to 3 times as many iterations to achieve. Such interactions are often observed in complex optimisation problems employing genetic algorithms²². We conclude that the diversity introduced by our mutation operator is sufficient to warrant the lower population limit.

We also experimented with a number of alternative GA propagation strategies. Originally we did not carry the best individuals from the previous generation directly through to the next. Instead, the search was allowed to diverge, and a record of the “best painting so far” was maintained separately. This resulted in a more lengthy relaxation process, which sometimes produced marginally fitter paintings than the current method. However the marginal aesthetic benefit that resulted did not seem to warrant the large increase in run-time. Similar results were observed using another early strategy; if, after a number of generations, we observe no change in fitness, then we may have reached a plateau in the problem space. In such circumstances the probability of large scale mutation occurring was gradually increased until the search escaped the plateau. Again, this caused lengthy execution times for which the pay off in terms of quantitative change in the fitness function, and qualitative improvement in aesthetics, was marginal.

Simplifying assumptions have been made in our salience measure. For example, the decision to use spherical JND surfaces in our visibility operator, and the use of a single Gaussian for clustering during rarity were made on the grounds of unattrac-

tive computational complexity during clustering.

As regards rendering, we might choose to texture strokes to produce more realistic brush patterns, although this should be a post-processing step so as not to introduce undue error in the comparison of saliency maps. Many techniques apply texture mapping to strokes^{8,5,10}, and a bump mapping technique was also proposed in²⁴. Highly realistic volume-based hairy brush models have recently been proposed²⁵ which could be swept along the Catmull-Rom spline trajectories generated by our algorithm. However, we have concentrated on stroke placement rather than media emulation, and we leave such implementation issues open. We believe the most productive avenues for future research will not be in incremental refinements to the system, but rather will examine alternative uses for saliency measures in the production of image-based non-photorealistic renderings.

References

1. E. H. Gombrich, *Art and Illusion*, Phaidon Press Ltd., Oxford, 1960.
2. J. P. Collomosse and P. M. Hall, "Painterly rendering using image saliency," in *Proc. 20th Eurographics UK Conference*, June 2002, pp. 122–128.
3. T. Sziranyi and Z. Tath, "Random paintbrush transformation," in *Proc. 15th Intl. Conference on Pattern Recognition (ICPR)*, Barcelona, 2000, vol. 3, pp. 155–158.
4. A. Hertzmann, "Paint by relaxation," in *Proc. Computer Graphics Intl. (CGI)*, July 2001, pp. 47–54.
5. A. Hertzmann, "Painterly rendering with curved brush strokes of multiple sizes," in *Proc. 25th Intl. Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH)*, 1998, pp. 453–460.
6. P. Haeblerli, "Paint by numbers: abstract image representations," in *Proc. 17th Intl. Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH)*, 1990, vol. 4, pp. 207–214.
7. M. Haggerty, "Almost automatic computer painting," *IEEE Computer Graphics and Applications*, vol. 11, no. 6, pp. 11–12, Nov. 1991.
8. P. Litwinowicz, "Processing images and video for an impressionist effect," in *Proc. 24th Intl. Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH)*, Los Angeles, USA, 1997, pp. 407–414.
9. S. Treavett and M. Chen, "Statistical techniques for the automated synthesis of non-photorealistic images," in *Proc. 15th Eurographics UK Conference*, Mar. 1997, pp. 201–210.
10. M. Shiraishi and Y. Yamaguchi, "An algorithm for automatic painterly rendering based on local source image approximation," in *Proc. 1st ACM Symposium on Non-photorealistic Animation and Rendering*, 2000, pp. 53–58.
11. B. Gooch, G. Coombe, and P. Shirley, "Artistic vision: Painterly rendering using computer vision techniques," in *Proc. 2nd ACM Symposium on Non-photorealistic Animation and Rendering*, June 2002, pp. 83–90.
12. M. Kass, A. Witkin, and D. Terzopoulos, "Active contour models," *Intl. Journal of Computer Vision (IJCV)*, vol. 1, no. 4, pp. 321–331, 1987.
13. D. DeCarlo and A. Santella, "Abstracted painterly renderings using eye-tracking data.," in *Proc. 29th Intl. Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH)*, 2002, pp. 769–776.

14. J. A. Bangham, S. E. Gibson, and R. Harvey, "The art of scale-space," in *Proc. 14th British Machine Vision Conference (BMVC)*, Sept. 2003, vol. 1, pp. 569–578.
15. K. N. Walker, T. F. Cootes, and C. J. Taylor, "Locating salient object features," in *Proc. 9th British Machine Vision Conference (BMVC)*, 1998, vol. 2, pp. 557–567.
16. P. M. Hall, M. Owen, and J. P. Collomosse, "Learning to detect low-level features," in *Proc. 15th British Machine Vision Conference (BMVC)*, 2004, vol. 1, pp. 337–346.
17. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Royal Statistical Society Series B*, vol. 39, pp. 1–38, 1997.
18. P. M. Hall, M. Owen, and J. P. Collomosse, "A trainable low-level feature detector," in *Proc. Intl. Conf. on Pattern Recognition (ICPR)*, 2004, vol. 1, pp. 708–711.
19. G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, John Wiley and Sons, 2nd edition, 1982, ISBN: 0-471-02106-7.
20. J. Holland, *Adaptation in Natural and Artificial Systems. An introductory analysis with applications to biology, control, and artificial intelligence*, Univ. Michigan Press, 1st edition, 1975, ISBN: 0-472-08460-7.
21. K. de Jong, "Learning with genetic algorithms," *Machine Learning*, vol. 3, pp. 121–138, 1988.
22. D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989, ISBN: 0-201-15767-5.
23. J. P. Collomosse, *Higher Level Techniques for the Artistic Rendering of Images and Video*, Ph.D. thesis, University of Bath, U.K., May 2004.
24. A. Hertzmann, "Fast paint texture," in *Proc. 2nd ACM Symposium on Non-photorealistic Animation and Rendering*, June 2002, pp. 91–96.
25. S. Xu, F. C. M. Lau, F. Tang, and Y. Pan, "Advanced design for a realistic virtual brush," in *Proc. Computer Graphics Forum (Eurographics)*, Grenada, Spain, Sept. 2003, vol. 3, pp. 533–542.