Chapter 4

Genetic Painting: A Salience Adaptive Relaxation Technique for Painterly Rendering

In this chapter we build on the single-pass salience adaptive painterly technique of the previous chapter to propose a novel, relaxation based iterative process which uses curved spline brush strokes to generate paintings[1]. We draw upon our previous observations of artistic practice to define the degree of optimality for a painting to be measured by the correlation between the salience map of the original image and the level of detail present in the corresponding painting. We describe a novel genetic algorithm based relaxation approach to search the space of possible paintings and so locate the optimal painting for a given photograph, subject to this criterion. In this work we make use of a more subjective, user trained measure of salience[2]. The quality of the rendering is further enhanced through the use of context dependent stroke rendering styles, and compensation for image noise; both are additional novel contributions to image-space AR.

4.1 Introduction

In the previous chapter we observed a relationship between the importance that artists assign to artifacts in a scene, and the level of detail and emphasis afforded to such artifacts in a piece of artwork. We also observed that automatic image-space AR techniques are at odds with this behaviour, emphasising all detail regardless of its salience.

[1] This work has previously appeared as [24].

[2] The trainable salience measure we use is described in Section 4.3 and will appear as [66]. The primary investigator of the measure was Peter Hall (University of Bath), with whom the author collaborated. For the purposes of examination, the areas of the measure to which the author has contributed are clearly indicated *in situ* within Section 4.3.

We rectified this discrepancy by developing a novel, single-pass painterly rendering process which modelled the artist's salience adaptive behaviour. The resulting paintings exhibited improved aesthetics; not only were salient regions painted precisely (strokes from non-salient regions did not encroach upon and distort regions of greater salience), but the rendering also exhibited a sense of focus around salient regions due to the abstraction of non-salient detail. We compared the edge maps of the resulting paintings with the salience maps of their original images, and observed the two to exhibit qualitatively closer correspondence than when using local frequency response filters to drive the rendering process (Figure 3-6).

We now build upon the success of this pilot, single-pass salience adaptive rendering technique (Section 3.3) to propose a novel relaxation based approach to salience adaptive painting. We build upon our previous observations to define the degree of optimality for a painting to be measured by the correlation between the salience map of the original image and level of detail within the corresponding painting. We describe novel salience based approach to painting which uses a genetic algorithm (GA) relaxation technique to search the space of possible paintings, and so locate the optimal painting for a given photograph. In doing so we use a more subjective 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, for example to detect edges, ridges and corners. A further contribution of our method is that differential rendering styles are possible by varying stroke style according to the classification of salient artifacts encountered. Uniquely, we also compensate for noise; a component of any real image.

There are several advantages that this novel technique holds over our previous single-pass technique of Section 3.3.

1. Single-pass rendering techniques (such as our pointillist-style algorithm and many existing image-space AR methods [58, 71, 103, 140, 159]) are highly susceptible to image noise. Some stroke attributes, such as orientation, are determined by local sampling of fields obtained through differentiation of the source image; for example Sobel intensity gradient fields. This differentiation process often serves to exaggerate noise present in the image, which manifests as numerical inaccuracies in the field leading to poor stroke placement or poor setting of visual attributes. In single-pass techniques, a stroke's location and visual attributes are determined once only, after which they remain fixed. One has no guarantee that the compositing of multiple, placed strokes will lead to an aesthetically optimal painting; strokes do not take into account the effect of other strokes in their vicinity. By contrast, a goal directed iterative painting strategy can approach a more optimal

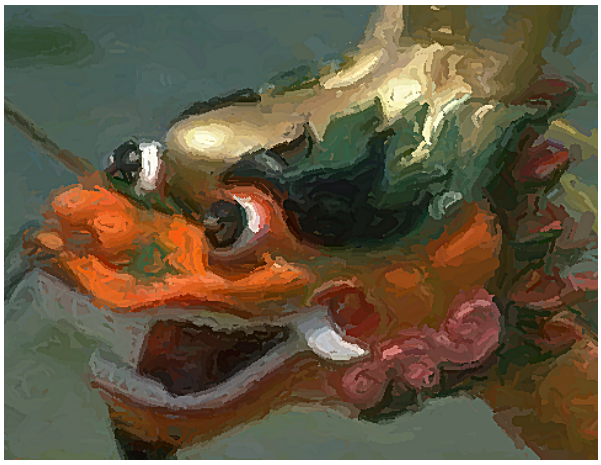


Figure 4-1 Section of a painting generated using our salience based relaxation technique, taken from the fittest individual within the 80th generation of paintings. High resolution versions of this and other paintings in this chapter are included in the electronic supplementary material in Appendix C.

solution. We describe a GA based relaxation process which iteratively varies the attributes of strokes, assessing the "fitness" of the entire painting at each iteration (using our salience based optimality criterion), with the aim of producing an optimal, target painting.

2. We make use of a user-trained salience measure, which users may teach to recognise artifacts which they deem to be perceptually important. As we mention in Chapter 3, salience is a task specific, subjective concept which can only be addressed by prescriptive measures at an early visual level. This trainable method provides a more subjective basis to the problem of determining the salience of artifacts within an image.

3. Our painterly technique not only drives emphasis in the painting via salience magnitude, but can also vary stroke rendering style according to the classification of salience artifact encountered (for example, edge or ridge).

4. Our algorithm composits multiple curved spline strokes to create paintings, rather than simple daubs of paint. This allows synthesis of paintings exhibiting elegant, flowing brush strokes, for example in the style of Van Gogh (Figure 4-17, right). This approach presents a more general painting solution, since the rendering parameters controlling maximum stroke length can be reduced, causing the output of the system to degenerate back toward daub based pointillist styles. Thus the proposed salience-driven system is capable of rendering photographs in a wider gamut of artistic styles than our previous salience-driven approach.

Our paintings are formed by compositing curved Catmull-Rom [14] spline brush strokes via an adaptation of the multi-scale curved stroke painterly technique proposed by

Hertzmann [71]. We build upon this work in two ways. First, we modify the technique to accommodate preferential rendering with regard to salience. Strokes are more densely placed in salient regions, then ordered and modulated to prevent strokes from non-salient areas encroaching upon 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. This context-dependent adaptation of stroke style is a novel contribution to AR. Second, we use our novel relaxation scheme to iteratively converge the rendering toward the "optimal" painting for a given image. 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 whilst non-salient high frequency detail (emphasised with other painterly methods) is attenuated. We demonstrate the results of our painterly technique on a wide range of images, illustrating the benefits of rendering with regard to salience and the improvements gained by subsequent relaxation of the painting using our GA based technique.

4.2 Background in Evolutionary Computing

Evolutionary Algorithms (EAs) seek to model the evolutionary processes found in biology, such as natural selection or mutation, to search for an optimal solution to a problem under a specific set of constraints. The early development of EAs dates back to the sixties, when the research of such algorithms fell under the encompassing title of "Evolutionary Computing". However the independent development of similar ideas by separate research groups ensured that, by the early eighties, EAs had diversified into three subtly distinct categories [70]: Genetic Algorithms (GAs) [36, 56, 77], Evolutionary Programming (EP) [50] and Evolutionary Strategies (ES) [131].

GAs are generally considered to originate from the cellular automata work of Holland *et al* [77]. The GA operates upon a population of individuals; each individual represents a point in the problem space and is uniquely characterised by its associated genome containing a "genetic code"; this code often takes the form of a binary string but this is not strictly necessary. Each individual can be evaluated via a "fitness function" to yield a scalar value corresponding to the optimality of the solution it represents. The GA operates by breeding successive generations of these individuals. Selection of individuals for breeding is via a stochastic process biased toward selecting fitter individuals; so exhibiting a Darwinian "survival of the fittest" behaviour. The breeding process itself involves the swapping of genetic code (genome fragments) between parents, to produce a novel individual. This exchange of code is termed the "cross-over" process. In addition, each element of the genome may be perturbed by some random amount

— however the probability of large scale perturbation is slight. This process is termed "mutation". Self propagation (analogous to asexual reproduction) of a single parent may also be permitted in some implementations. This survival of the fittest methodology can evolve populations of individuals which tend to approach global maxima even for complex problems exhibiting turbulent, high dimensional problem spaces. The GA is a methodology, rather than an algorithm, and there are five principal issues that must be resolved to successfully tailor the GA to a specific application:

1. How should a solution be represented by an individual's genome?
2. What are the mechanics of the cross-over and mutation processes?
3. How should the initial population be generated?
4. How should fitness be defined?
5. What should the population size be? Should it vary or remain static?

We address these issues *in situ*, during the explanation of our GA based relaxation process (Section 4.4.2).

The EP methodology is subtly different to that of the GA, in that no cross-over is performed. Offspring for successive generations are copied from a single parent, selected stochastically with a bias to fitness, and subjected to mutation; major mutations have a much lower probability than minor mutations. The genome tends not to be explicitly represented as a string, but as a point in the problem space. Mutation in many cases takes the form of a translation in the problem space of stochastically chosen direction and magnitude.

The ES methodology is very similar to that of EP, but again differs slightly. The selection process in EP is often "tournament based"; a pair of parents are picked at random to compete in a tournament — the outcome being decided by their fitness and the winner being allocated a "win point". Many individual tournaments take place when producing a successive generation, and the highest aggregate scoring parents are allowed to propagate. By contrast, with ES the weaker parents are deterministically culled from the population prior to propagation.

4.2.1 Genetic Algorithms in Computer Graphics

EA based techniques have been successfully applied to a number of areas within Computer Graphics. Early uses of EAs include Reynolds' distributed behavioural models [129]. These simulated many individual automata interacting via simple rules, from

which emerge more complex collective behaviours such as flocking and herding. A realistic model of fish locomotion was presented by Tu and Terzopoulos [160], which simulated perception, learning and, like Reynolds' automata, also exhibited group behaviour. A well known application of GAs to Computer Graphics is in the virtual creatures of Sims [141]. Each of Sims' creatures is defined genetically, and generations of creatures can evolve novel behavioural and graphical characteristics over time. Sims posited that creatures which evolve via simulated genetic processes should be capable of exhibiting more a complex and realistic biological evolutionary response than possible through explicit, procedural modelling. GAs have also been applied to content based image retrieval (CBIR) [2].

The majority of applications for GAs in the field of Computer Graphics address the problem of goal directed animation. GAs have been successfully applied to animate realistic flower growth [105], and also to develop stimulus response systems for human articulated motion [115]. Tang and Wan [157] described a GA based system which allows character motions to evolve in virtual environments, for example learning the optimal way to perform a jump to reach a goal. Motion planning and character animation techniques were also driven by GAs in [126] and [179] respectively.

Our paint by relaxation technique is GA based. To justify this, consider Haeberli's [62] abstraction of a painting as an ordered list of strokes (comprising control points, thickness, etc. with colour as a data dependent function of these); the space of possible paintings for a given image is clearly very high dimensional, and our optimality criterion makes this space extremely turbulent. Stochastic searches that model evolutionary processes, such as genetic algorithms, are often cited as among the best search strategies in situations of similar complexity [36]. This is due to the fact that GAs search from a population of points, not a single point, and that the mutation and cross-over processes integral to propagation cause jumps in the problem space which can mitigate against the attraction of local minima; these cause difficulty to other strategies such as gradient descent or simulated annealing. Furthermore, whilst it is difficult to explicitly model the complex relationships between stroke parameters during the creation of a painting, goal driven stochastic optimisers such as GAs are known to perform acceptably is the absence of such models.

4.3 Determining Image Salience

For the purposes of examination, please note that the primary investigator of the salience measure described in this section was Peter Hall of the University of Bath, with whom the author collaborated. Specifically, Hall developed the basic measure which operates at a single scale (Sections 4.3.1– 4.3.3). The multi-scale extensions to the measure were investigated and developed by the author (Section 4.3.4).

Our painterly process requires a method to automatically estimate the perceptual salience 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 salience of the corresponding image point. We now describe an 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}). These are combined to estimate the final probability of an image artifact being salient as:

$$P_{salient} = P_{rare}P_{visible}P_{class} \tag{4.1}$$

Each of the three operators makes use of circular signals generated by sampling from concentric rings centred upon the pixel whose salience is to be determined. The first operator performs unsupervised global statistical analysis to evaluate the relative rarity (P_{rare}) of image artifacts. This process is similar to our original rarity based approach of Section 3.2, and the motivation for this operator is principally to adapt that rarity measure to be consistent with the circular sampling strategy. However the rarity based measure is augmented with two further operators. Not all rare artifacts should be considered salient; for example, normally invisible JPEG image compression artifacts can sometimes be regarded as salient using rarity alone. A prerequisite for salient artifacts is therefore that they should also be visible, motivating a second perceptually trained operator which estimates the visibility ($P_{visible}$) of image artifacts. The measure is refined by asserting that certain classes of artifact, for example edges or corners, may be more salient than others. This motivates use of a third operator, which users train the system by highlighting artifacts in photographs they regard as salient. 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 salience (P_{class}).

This definition allows for a more subjective measure of salience, and holds further advantages in that *classes* of salient features may be trained and classified independently. This allows stroke parameters to vary not only as a function of salience magnitude, but

also allows differentiation of rendering style according to the classification of salient regions (see Figure 4-8).

4.3.1 Determining Pixel Rarity

The first operator is an unsupervised technique for determining pixel rarity. The technique is very similar to that of Section 3.2 in that a model is constructed which encodes the statistical distribution of a set of measures locally associated with each pixel, and the outliers of this distribution are isolated.

For a given pixel $\underline{p} = (i, j)^T$ the operator examines a series of rings of radius ρ, each centred at $(i, j)^T$. The image is uniformly sampled around each ring's circumference at angular positions θ, hence obtaining a discrete signal $\underline{x}(\underline{p}) = (\rho, \theta) \in \Re^3$; 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 pixels in increments of 0.5, to yield good results in subsequent processing. As before, an eigenmodel is created from the collection of vectors $\underline{x}(.)$ resulting from each pixel within the image. The Mahalanobis distance $d(.)$ is then computed for all pixels $\mathcal{P}$ in the image.

$$d^2(\underline{x}(.)) = (\underline{x}(.) - \underline{\mu})^T \underline{\underline{U \Lambda U}}^T (\underline{x}(.) - \underline{\mu}) \tag{4.2}$$

The probability of an individual pixel $\underline{q} \in \mathcal{P}$ being rare is then written as a quotient measuring the fraction of the sample density which is less rare than the pixel $\underline{q}$:

$$\mathcal{Q} = \{\underline{r} : d(\underline{x}(\underline{r})) \leq d(\underline{x}(\underline{q})) \wedge \underline{r}, \underline{q} \in \mathcal{P}\} \tag{4.3}$$

$$P_{rare}(\underline{q}) = \frac{\sum_{\underline{p} \in \mathcal{Q}} d(\underline{x}(\underline{p}))}{\sum_{\forall \underline{p} \in \mathcal{P}} d(\underline{x}(\underline{p}))} \tag{4.4}$$

4.3.2 Determining Visibility

The second operator estimates the probability that a local image window contains a perceptually visible signal. The *just noticeable difference* (JND) between colours in RGB format is empirically measured. It is assumed that for each RGB colour $\underline{r}$ there is distance $\tau(\underline{r})$, also in RGB space. Together the colour and the distance specify a sphere of RGB colours $(\underline{r}, \tau(\underline{r}))$. No colour interior to the surface of the sphere can be perceptually discriminated from the centre colour, whilst all exterior colours can be so discriminated. The distance $\tau(\underline{r})$ is one JND at the colour $\underline{r}$. The sphere radius can vary depending on experimental conditions, and after several experimental trials τ emerges as the mean radius accompanied by an associated standard deviation σ. Although this is a simple colour model (an ellipsoid might better model JND surfaces) it has been found to perform satisfactorily, and the reader is referred to [66] for dis-

cussion and experimental details. Similar distance metrics are also described in [175] for luminance. The advantage of the this approach and [175], over other perceptually based colour spaces (such as CIELAB), is that unit distances in JND space correspond to colour distances that are only just discernible by the user.

To evaluate the visibility of artifacts local to a point $\underline{p} = (i, j)^T$, the image is sampled in a manner identical to Section 4.3.1 to obtain a signal (ρ, θ), the differential magnitude of which may be written as:

$$d(\rho, \theta; \underline{p}) = \left| \left| \frac{d\underline{c}(\rho, \theta; \underline{p})}{d\rho} \right|^2 + \left| \frac{d\underline{c}(\rho, \theta; \underline{p})}{d\theta} \right|^2 \right|^{1/2} \tag{4.5}$$

where $\underline{c}(\rho, \theta; \underline{p})$ returns the RGB value of the image at coordinates (ρ, θ) relative to $\underline{p}$. This is, however, not a perceptual distance and the probability $\phi(.)$ that this change is visible is computed as:

$$\phi(\rho, \theta) = erf((d(\rho, \theta) - \tau)/\sigma) \tag{4.6}$$

where τ and σ are the JND and its deviation for the colour sample at $\underline{c}(\rho, \theta)$ in the local window. The reasoning is that if a signal is visible in any ring, then it is visible for the whole ring but not for the whole disc, and so write:

$$P_{visible} = \sum_{\rho=1}^{\rho_{max}} \max(\phi(\rho, \theta)) \tag{4.7}$$

as the probability of the disc being visible. This definition ensures that if a signal grazes the edge of the disc it will register as visible, but not strongly because it will not pass through every ring. If, on the other hand, a signal pases through the centre of the disc then it passes through every ring, and a high visibility is obtained.

4.3.3 Classification of Image Artifacts

The final operator introduces 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 $\underline{p}$, the image is sampled in an identical manner to that used for determining pixel rarity. However, each ring is treated separately, and so considers the classification of the colour signal $\underline{c}(\theta)$ at constant ρ (this transpires to be more stable than considering the disc as a whole). A feature vector is formed by first differentiating $\underline{c}(\theta)$, using Euclidean distance in RGB space, to obtain a periodic scalar signal $\underline{y}(\theta)$ (Figure 4-2).

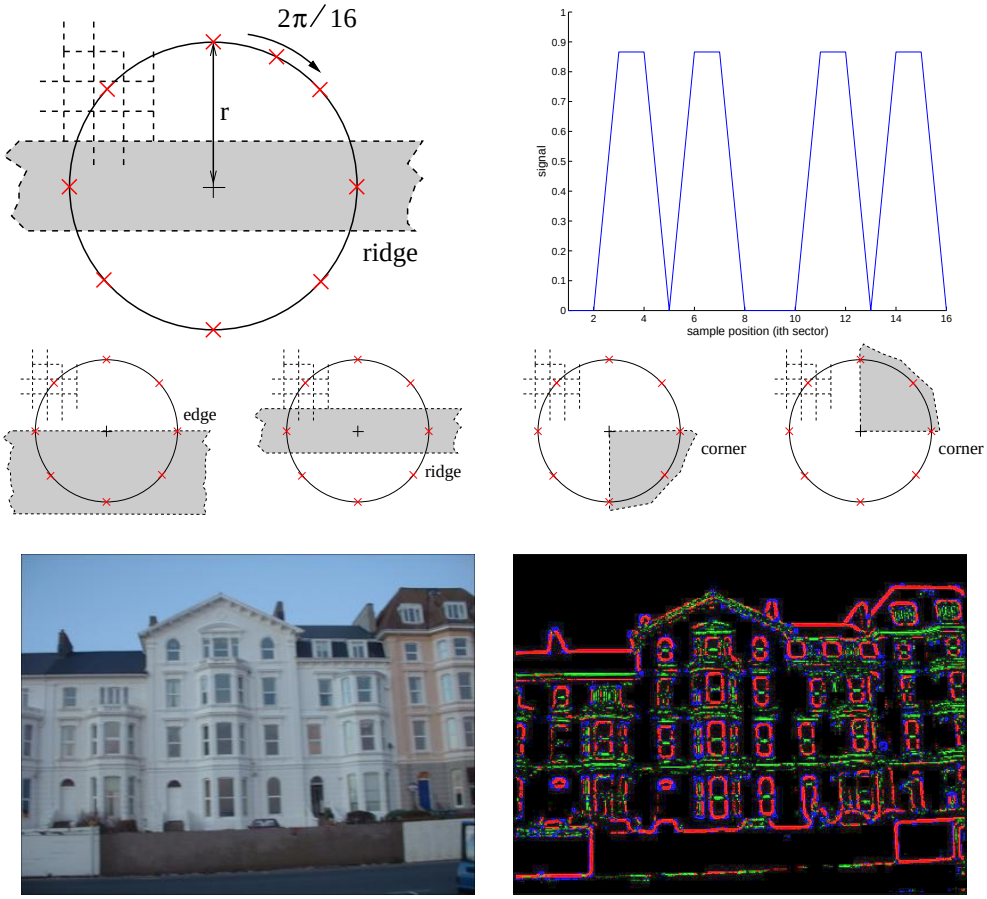


Figure 4-2 Circular descriptors (top left) are used to create signals (top right) from points specified as salient by the user, which are then processed and clustered in a high dimensional space. Features such as ridges, edges and corners (second row) create distinctive spectral signals (third row), which may be used to determine not only the salience of a point, but also its classification type. Bottom row: a photograph and its corresponding salience map with edges in red, ridges in green and corners in blue.

The absolute value of the Fourier components $|F[\underline{y}(\theta)]|$ are computed, normalised to unit power, and the d.c. (zeroth) component dropped. Thus for a given $\underline{y}(\theta)$ a feature is computed as follows:

$$\underline{f}(\omega) = \frac{|F[\underline{y}(\theta)]|}{\left(\sum_\theta |\underline{y}(\theta)|^2\right)^{\frac{1}{2}}} \tag{4.8}$$

$$\underline{f}(\omega) \leftarrow \underline{f}(\omega) \setminus \underline{f}(0) \tag{4.9}$$

by appeal to Parseval's theorem to compute power. 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. Thus $\underline{c}(\theta)$, $\underline{c}(\theta)+\underline{\alpha}$, $\underline{c}(\theta+\beta)$ all map to the same point in feature space. The system has proven to be robust to more general

colour scalings, $\gamma\underline{c}(\theta)$, but cannot be invariant (suppose $\gamma = 0$).

It is these properties that principally motivated choice of circular sampling (after Smith and Brady who advocated the use of circular sampling [144] in their SUSAN system), 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 (forming a closed one-dimensional manifold).

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 artifacts (such as edge, ridge, or corner), and identify a class label with each artifact they highlight. Training therefore results in multiple sets of artifacts, each set containing artifacts of identical class.

To build the classifier each artifact in a given set, k say, is converted into a feature vector as previously described. An estimate of the class conditional density $p(\underline{f}|k)$ for that set of features is then obtained using a Gaussian Mixture Model (GMM), fitted using Expectation Maximisation [41]. A prior, $p(k)$, is also estimated 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 computation of the posterior likelihood $p(k|\underline{f})$ by appeal to Bayes theorem:

$$p(k|\underline{y}) = \frac{p(\underline{y}|k)p(\underline{y})}{\sum_{j=1}^{N} p(\underline{y}|j)p(j)} \tag{4.10}$$

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 (one element per class). The L_1 norm of this vector is unity, and in fact we can simply add elements to estimate the probability that an artifact belongs to a subset of classes. For each classified pixel we therefore have a probability $p(k|\underline{y})$ of membership to each of the trained classes, and compute P_{class} as the maximum value over all $p(k|\underline{y})$. Later, this classification allows us to vary the stroke rendering style according to the class of salient artifact encountered — see Figure 4-8.

4.3.4 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

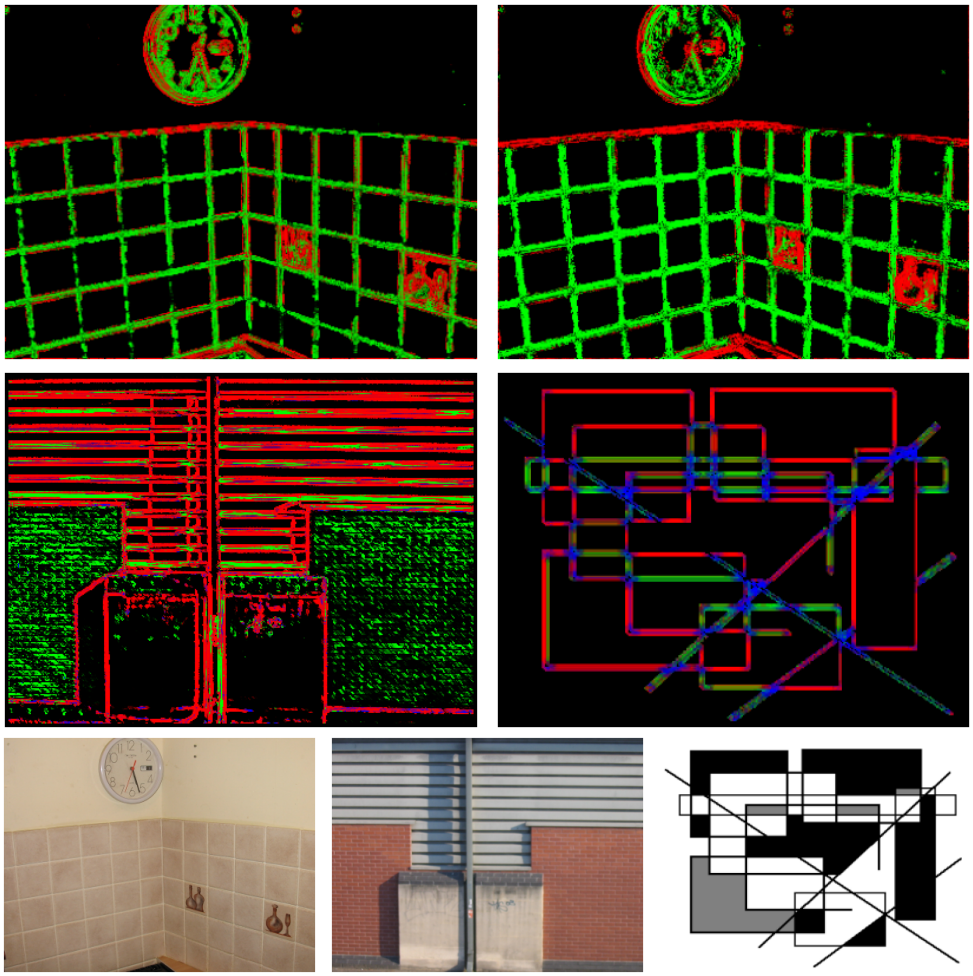


Figure 4-3 Top: Classifier output using a single scale $\rho = 3$ (left) and multiple scales (right). Note that ridges and edges around the tiles are more clearly discriminated using the multi-scale technique. Middle: Further multi-scale classified output on real and synthetic source images. Bottom: Source images.

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 variation, 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, but 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 as salient.

Our strategy is to perform the classification of a given point at several values of ρ; again using the range 1 to 3 pixels at increments of 0.5. At each scale we obtain a posterior probability vector $p(k|\underline{y})$, and concatenate these to form a column vector (in effect, 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

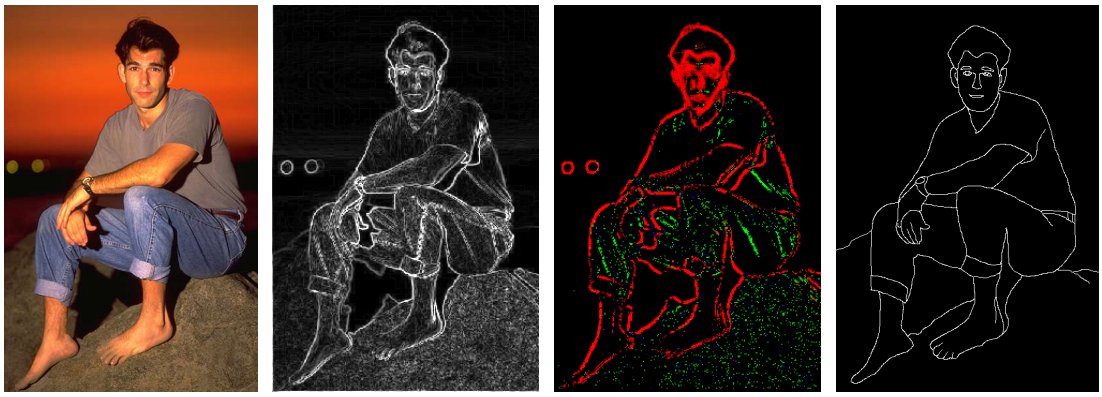


Figure 4-4 Applying the salience measure to an image from test set β, from left to right: original image, Sobel filtered image, multi-scale salience map (red lines indicate salient edges), and a manually specified ground truth of salient edges. The salience measure allows us to discriminate between salient edges, and the ridges and non-salient artifacts that comprise the remainder of the image. Such distinction is not possible using local measures. We later present a painterly rendering of this image in Figure 4-13, in which salient edges (for example the face) are emphasised, and other details (such as the rock texture) are abstracted away.

of salient feature classes, this time by fitting GMMs in this scale-dependent space. The advantage of our approach is that the aforementioned "heuristics" for classification are now implicitly learnt by example.

The extension of the feature classifier to operate at multiple scales impacts both a) the ability to determine salience magnitude, and b) the ability to classify the salient artifacts encountered — both are relevant to our painting process. We performed two experiments to measure each of these impacts respectively.

Experiment 1: Salience magnitude

We trained both the single-scale and multi-scale versions of the salience measure using the image set α (see Figure 4-5). The classes of feature trained on were edges, ridges and corners. In the case of the single-scale measure, we used a disc radius value of $\rho = 3$ for both the training and classification processes; this value has been found to work well over many images (see [66]). Once trained, we applied both measures to a further image set β (distinct from α), to determine salience magnitude within those images ($P_{salient}$, see equation 4.1). We also manually obtained a ground truth salience map for each image in β manually, from a human participant instructed to draw over the important features in the image. The ground truth, and the training, were supplied by the same participant in our experiment (a sample ground truth map is given in Figure 4-4). We compared the output of the salience measures with the ground truth salience map, to determine the performance of each salience measure.

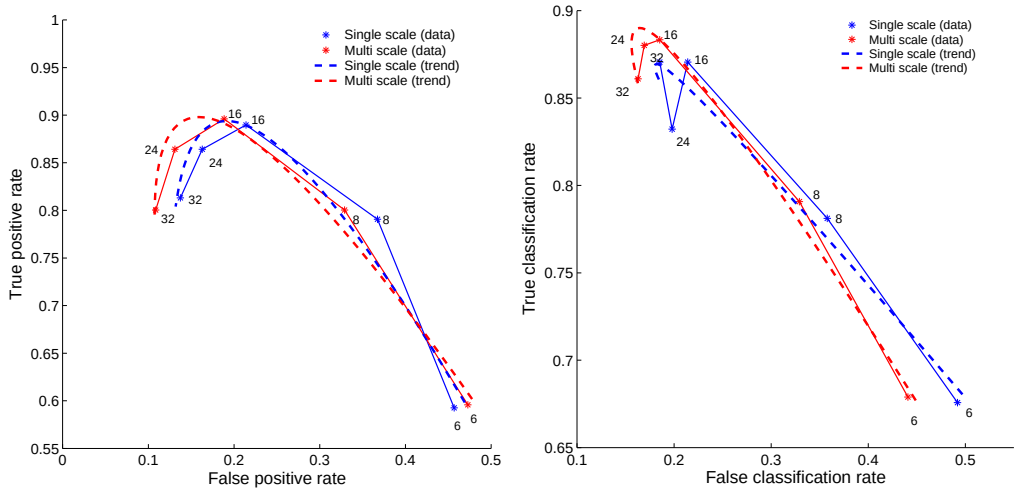
The upper left graph of Figure 4-5 summarises the results of this experiment. The graph shows the ROC curve for the detection of salient pixels, where $P_{salient} \geq 0.5$ (that is, where pixels are more likely to be salient than not). The curve plots sensitivity (true positive rate; the fraction of ground truth salient points classified as salient by the measure), versus specificity (one minus the false positive rate; the fraction of ground truth non-salient points classified as salient). These measures were averaged over the ten image test set (β). A range of radial sampling rates were tested $\{6, 8, 16, 24, 32\}$ for both the single and multi-scale versions of the salience measure. The performances of both measures over the range of sampling rates were plotted with accompanying trend lines.

Both measures perform well at sampling rates of around 16, providing a good compromise between true positives and false positives, and motivating the choice of 16 sampling intervals in our painting process. Both the true positive and false positive rates fall as sampling rate increase beyond the neighbourhood of 16; in general fewer points are identified as salient at these rates. Although the internal representation of the circular signal is superior at higher sampling rates, it is likely that the higher dimensionality of the space in which these signals are distributed inhibits clustering (since both true and false positive signals decline uniformly). Note that although the rate of true positives (pixels correctly identified as salient) does not increase greatly using the multi-scale approach, the false positive rate declines significantly, so improving performance.

Experiment 2: Accuracy of Classification

Our second experiment tested for any specific improvements in classifier accuracy due to the multi-scale extensions. The experiment applied the same single and multi-scale measures, trained over image set α, to the test image set β. However, the ground truth in this experiment was a manually specified map in which corners, edges and ridges had been manually identified (in a similar manner to the classifier training process itself). As with experiment 1, the same person trained the classifier as provided the ground truth data.

Figure 4-5 contains the results of this experiment. The upper right graph shows an ROC curve, which compares the true and false classification rates (with respect to the manually specified ground truth). This graph was generated by forming a "confusion matrix" for each test image. Examples of a confusion matrices for a real image (the kitchen image, Figure 4-3, bottom-left), and a synthetic image (the Mondrian image, Figure 4-3, bottom-right), and also given in Figure 4-5. The values in the confusion matrix represent the proportion of ground truth artifacts (specified by the horizontal,



Conf. matrices (kitchen)

Single scale

	$\mathbf{c}_t$	$\mathbf{e}_t$	$\mathbf{r}_t$
$\mathbf{c}_d$	0.97	0.03	0.00
$\mathbf{e}_d$	0.01	0.81	0.05
$\mathbf{r}_d$	0.02	0.09	0.89

Multi-scale

	$\mathbf{c}_t$	$\mathbf{e}_t$	$\mathbf{r}_t$
$\mathbf{c}_d$	0.97	0.00	0.00
$\mathbf{e}_d$	0.00	0.91	0.04
$\mathbf{r}_d$	0.03	0.03	0.90

Training set (α)

Conf. matrices (synthetic)

Single scale

	$\mathbf{c}_t$	$\mathbf{e}_t$	$\mathbf{r}_t$
$\mathbf{c}_d$	0.89	0.15	0.17
$\mathbf{e}_d$	0.05	0.65	0.15
$\mathbf{r}_d$	0.03	0.14	0.64

Multi-scale

	$\mathbf{c}_t$	$\mathbf{e}_t$	$\mathbf{r}_t$
$\mathbf{c}_d$	0.98	0.18	0.16
$\mathbf{e}_d$	0.01	0.72	0.10
$\mathbf{r}_d$	0.00	0.09	0.73

Test set (β)

Figure 4-5 Comparison of salience measure performance under varying circular sampling rates, using single-scale vs. multi-scale classification (results discussed in Section 4.3.4). Top: Two ROC graphs of the single-scale (blue) and multi-scale (red) measures, showing performance over various sampling rates (labelled). Top-left: True and false positives wrt. detection of salient points. Top-right: True and false classifications rates of identified salient feature points. Bottom left: Confusion matrices when classifying a sample real and synthetic image, with and without multiple scale classification (c, e and r represent rates for corners, edges and ridges respectively. Subscript t indicates ground truth classifications, while subscript d indicates detected classification). Bottom right: The distinct training (α) and test (β) image sets.

subscript "t") deemed to be of a certain classification (specified by the vertical, subscript "d"). Note that columns in the matrix need not sum to unity, since it is possible that a true corner, for example, may not be picked out as salient by the measure (a false negative). Likewise, the rows need not sum to unity since salient artifacts identified by the measure need not have been specified as such in the ground truth (a false positive).

The diagonal of the matrix may be averaged to give the true classification rate for an image. The remainder of the matrix may be averaged to obtain the false classification rate. The values used for the ROC curve correspond to these true and false classification rates, averaged over the entire test image set (β). On average, the multi-scale approach to classification demonstrates superior performance than the single scale approach (exhibiting higher true positive, and lower false negative rates for classification).

Compare the confusion matrices for the single and multiple scale approaches. Performance for both the real and synthetic images is improved in both cases using the multi-scale approach. This may be visually verified by comparing the classification results of the real (kitchen) scene in Figure 4-3 with (upper-right) and without (upper-left) the use of multi-scale information; ridges and edges are discriminated more clearly in the former case. Indeed, the confusion matrices show the greatest performance increase is in the discrimination of these two feature classes — confirming our suggestion that scale is of great importance when deciding between a classification of "edge" or "ridge". The performance increase on the synthetic (Mondrian) image is less pronounced than that of the real scene. We suggest that this is possibly due to the training of the classifier on image set α, which consists entirely of real world images; the frequency characteristics of a synthetic scene may exhibit differences which impede the classification process, although acceptable classifications are produced by the system (see Figure 4-3, middle-right).

4.4 Generating the Painting

We now describe our algorithm for generating a painting from a 2D image. We begin by computing a salience map for the source image using the technique of Section 4.3. An intensity gradient image is also computed via convolution with directional Gaussian derivatives, from which a gradient direction field is obtained by taking arc tangents. In areas of low gradient magnitude the directional field can be unreliable, and so is interpolated smoothly from neighbouring pixels using a distance transform. The source image, direction field and salience map are used in subsequent stages of the painting algorithm. We first describe how individual strokes are placed to create a painting, and then describe the relaxation stage which results in the generation of an "optimal" painting.

4.4.1 Stroke placement algorithm

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 [58, 71]), control points are interpolated. We begin by placing seed points

on the canvas, from which strokes are subsequently grown bidirectionally. Seeds are placed stochastically, with a bias toward placement of seeds in more salient regions. As a heuristic we make provision for a stroke to be seeded at every other pixel; the salience map then governs the distribution of these strokes over the image. In practice we scatter 95 percent of the n strokes in this manner, the remaining 5 percent are scattered uniformly; this prevents holes appearing in areas of relatively low salience (Figure 4-6, right).

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 algorithm [71] in that we hop between pixels in the direction tangential to their intensity gradient. A history of visited pixels is recorded, and used to form the control points for the spline stroke.

We observe that noise forms a component of any real image, and any locally sampled direction estimate is better regarded as being sampled from a stochastic distribution (Figure 4-6, left). We assume that noise obeys the central limit theorem, and so model this distribution as a zero centred Gaussian, $G(0, \sigma)$; we determine σ empirically (see next subsection). Given a locally obtained gradient 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 μ' and σ' being inversely proportional to the local value of the precomputed salience map. Provided that a preset minimum number of hops have been executed, the growth of a stroke end is halted when either the curvature between adjacent pixels, or the distance (in *JND* space) between the

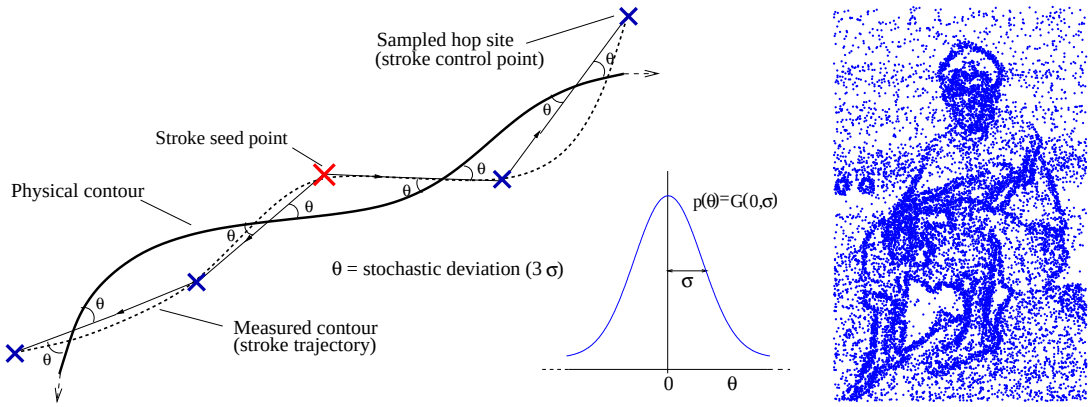


Figure 4-6 Left: Illustrating the stochastic growth of strokes from a seed pixel. We choose strokes with hop sites which minimise our objective function, under the constraint that hop angles are drawn from the distribution $p(\theta) = G(0, \sigma)$. Right: The salience-biased stochastic distribution of strokes, corresponding to the painting of Figure 4-13.

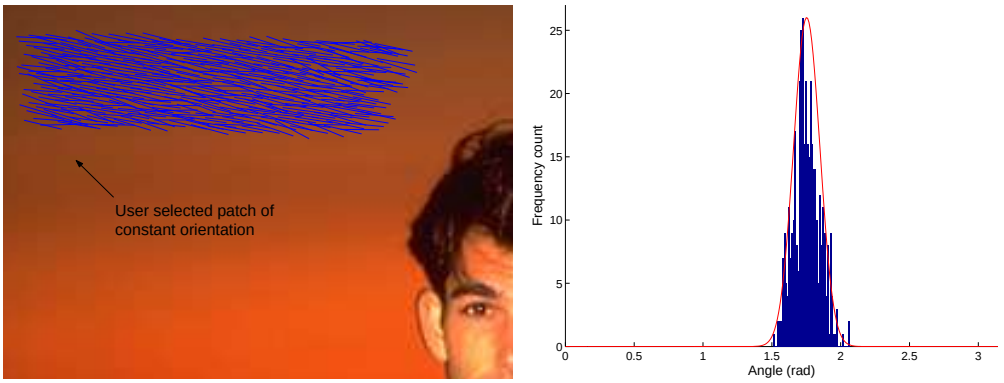


Figure 4-7 The system estimates the standard deviation of image noise by accepting a "ground truth" from the user. The user selects a region of the image within which they deem paint strokes should be orientated in a uniform direction (left). The gradient direction of pixels within this region is then measured (blue). We have observed that such image noise tends to obey the central limit theorem, and so is well modelled by a Gaussian (above, the mean direction is approximately 88°, standard deviation $\sigma \approx 5°$.

colour of the pixel to be appended and the mean colour of visited pixels, exceeds a threshold (preset at 3 JNDs).

This method initially yields a sub-optimal trajectory for the stroke with respect to our measure, described in Section 4.1. For a "loose and sketchy" painting this is often desirable (see Figure 4-8), but for painting styles exhibiting tighter stroke placement, trajectories must be closer to the optimal. The degrees of freedom resulting from each of the many stochastic 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 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 Section 4.4.2. 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.

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 and sketchy painterly system, whilst a high value will lengthen the relaxation process unnecessarily 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 multiple samples of the gradient components, from which we compute gradient direction by taking arc-tangents. We have observed the natural distribution of these values to be Gaussian, confirming our assumption that such image noise obeys the central limit theorem (Figure 4-7). We therefore take the mean angle $\mu(.)$ 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 bidirectional 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 lower CCD quality or JPEG compression). This variation allows between twelve and thirty degrees of variation per hop for noisy images which, given the number of hops per stroke, is a wide range of loci for a single stroke. Such observations 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 (i.e. single iteration) painterly rendering systems [58, 71, 103, 140].

Rendering and Differential Styles

Stroke rendering attributes are set automatically as a function of *stroke salience*, taken as the mean value of the salience map under 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. As with our previous algorithm (Section 3.3) 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 4.3.3). In Figure 4-8 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 — while 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

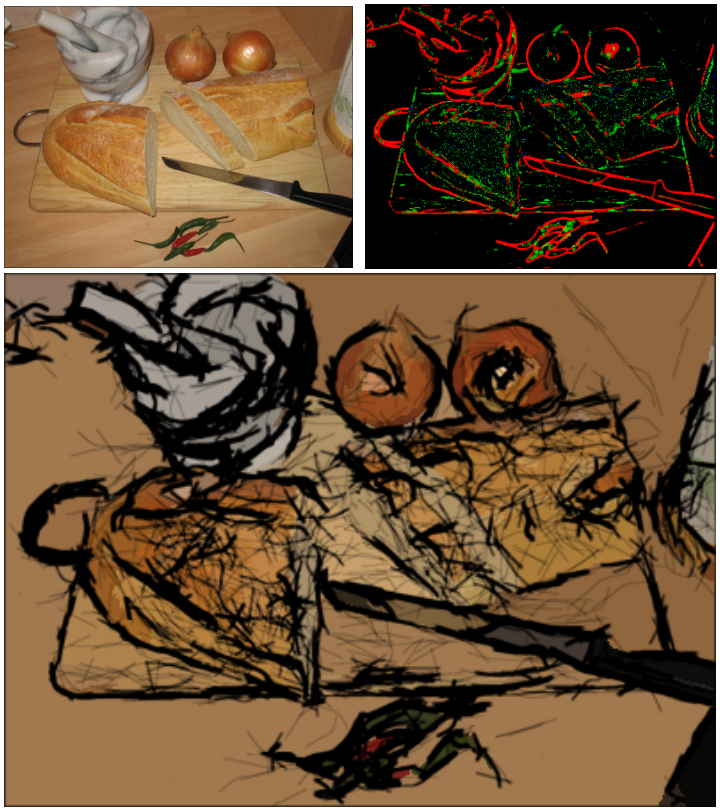


Figure 4-8 Top: a still-life composition and corresponding salience map. Bottom: the above rendered as a loose and sketchy painting, exhibiting differential stroke rendering styles determined by local feature classification. Edges are drawn with hard, precise thick strokes; ridges with a multitude of light, inaccurate strokes. Rendered prior to the relaxation step of Section 4.4.2.

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. This process also adds semantic to the rendering, in that we render ridges as single strokes, rather than as two separate edge strokes. To the best of our knowledge the rendering of paintings in differential styles via an automated heuristic is a novel contribution to AR.

4.4.2 Relaxation by Genetic Algorithm

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 algorithm such individuals are paintings; ordered lists of strokes and their associated attributes. Recall that we define the fitness of a given painting as proportional to the correlation between the salience map of the original image and level of (high frequency) detail within the corresponding painting.

Fitness and Selection

We begin by seeding an initial generation of paintings using the approach described in Section 4.4.1 and then enter the iterative phase of the genetic algorithm (Figure 4-11). First we perform evaluation; the entire population is rendered, and edge maps of each painting produced using by convolution with Gaussian derivatives, which serve as a quantitative measure of local, high frequency detail. The scale of the Gaussian is carefully chosen so as to smooth the fine inter-stroke edges, and prevent these influencing this detail measure in the painting. The generated edge maps are then compared to the precomputed salience map of the source image. The mean squared error (MSE) between maps is used as the basis for fitness measure $F(.)$ for a particular painting; the lower the MSE, the better the painting:

$$F(I, \psi) = 1 - \frac{1}{N} \sum |S(I) - E(\Psi(I, \psi))|^2 \tag{4.11}$$

The summation is over all N pixels in source image I. $\Psi(.)$ denotes our painterly process, which produces a rendering from I and an ordered list of strokes ψ (ψ corresponds to an individual in the population). Function $S(.)$ signifies the salience mapping process of Section 4.3, and $E(.)$ the process of convolution with Gaussian derivatives to produce an edge map. In this manner, individuals in the population are ranked according to fitness. The bottom ten percent are culled, and the top ten percent 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 top ninety percent are used to produce the remainder of the next generation through simulated natural selection. 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.

Cross-over

We now describe the cross-over process in detail (Figure 4-9, below). Two difference images, A and B, are produced by subtracting the edge maps of both 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 atoms of our painterly renderings are thick brush strokes rather than single pixels, we perform several binary dilations 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. We mask these conflicting regions with a coarse chequerboard texture (of random scale and phase offset) to decide

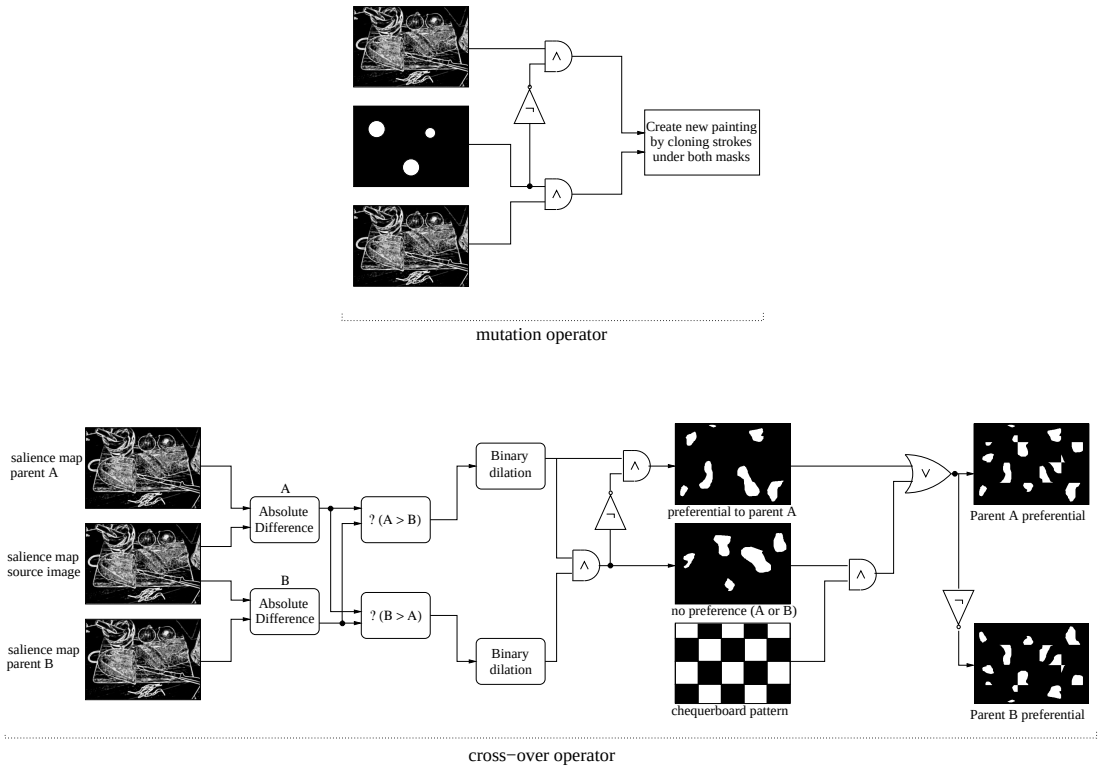


Figure 4-9 Genetic operators: the mutation and cross-over operators used during the relaxation process.

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.

Mutation

Finally, when a bred individual passes to a successive generation it is subjected to a random mutation. A new "spare" 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 spare painting; the spare painting is then discarded. In our implementation large areas of mutation are relatively rare, averaging around four hundredths of the image area per painting.

Termination

The relaxation process runs until the improvements gained over the previous few generations are deemed marginal (the change in both average and maximum population fitness over sliding time window fall below a threshold Δ), at which point the search has settled into a minima (see Figure 4-20) of sufficient extent in the problem space that escape is unlikely (Figure 4-10, right). The fittest individual in the current pop-

ulation 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 found to be a typical range of standard deviations for image noise (see Section 4.4.1). Forcing larger values of σ can result in convergence but, we observe, at the cost of an exponential increase in execution time (Figure 4-10, left).

Parallel Implementation

In practice, evaluation is the most lengthly 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/UltraSPARC) compute cluster running on our local network. Each painting in the population represents one "job" of work. Execution is blocked until the entire population of paintings are rendered, that is, all jobs are complete. In order to maintain acceptable execution speeds it is therefore important to assign jobs to machines in an efficient manner.

The time between sending a job to a helper (slave) machine, and the return of results from that machine is recorded. A mean execution time is thus maintained and updated for each machine throughout the rendering of a population. In our original implementation, jobs were simply farmed out to the first available machine. However in a heterogeneous system, machines may be of varying speeds and capabilities, and a single relatively slow machines can severely impact the performance of the whole

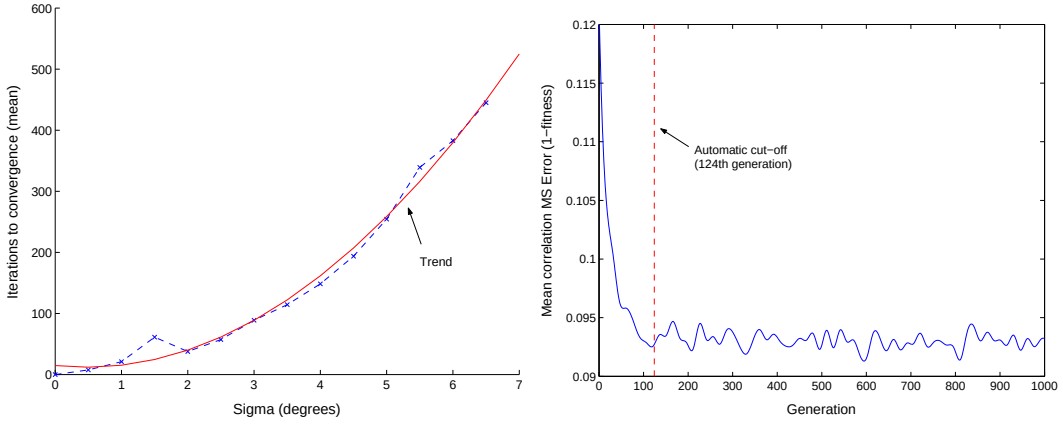


Figure 4-10 Illustrating GA termination. Left: Increasing the value of σ allows processing of noisier image at the cost of an exponentially increasing execution time. Right: the MSE (inverse fitness) averaged over the entire population. Automatic algorithm termination was suppressed, and the GA forced to run for 1000 iterations (data has been sub-sampled for presentation). This is representative of relaxation process' behaviour, and there is little advantage in exploring additional local problem space after the termination point.

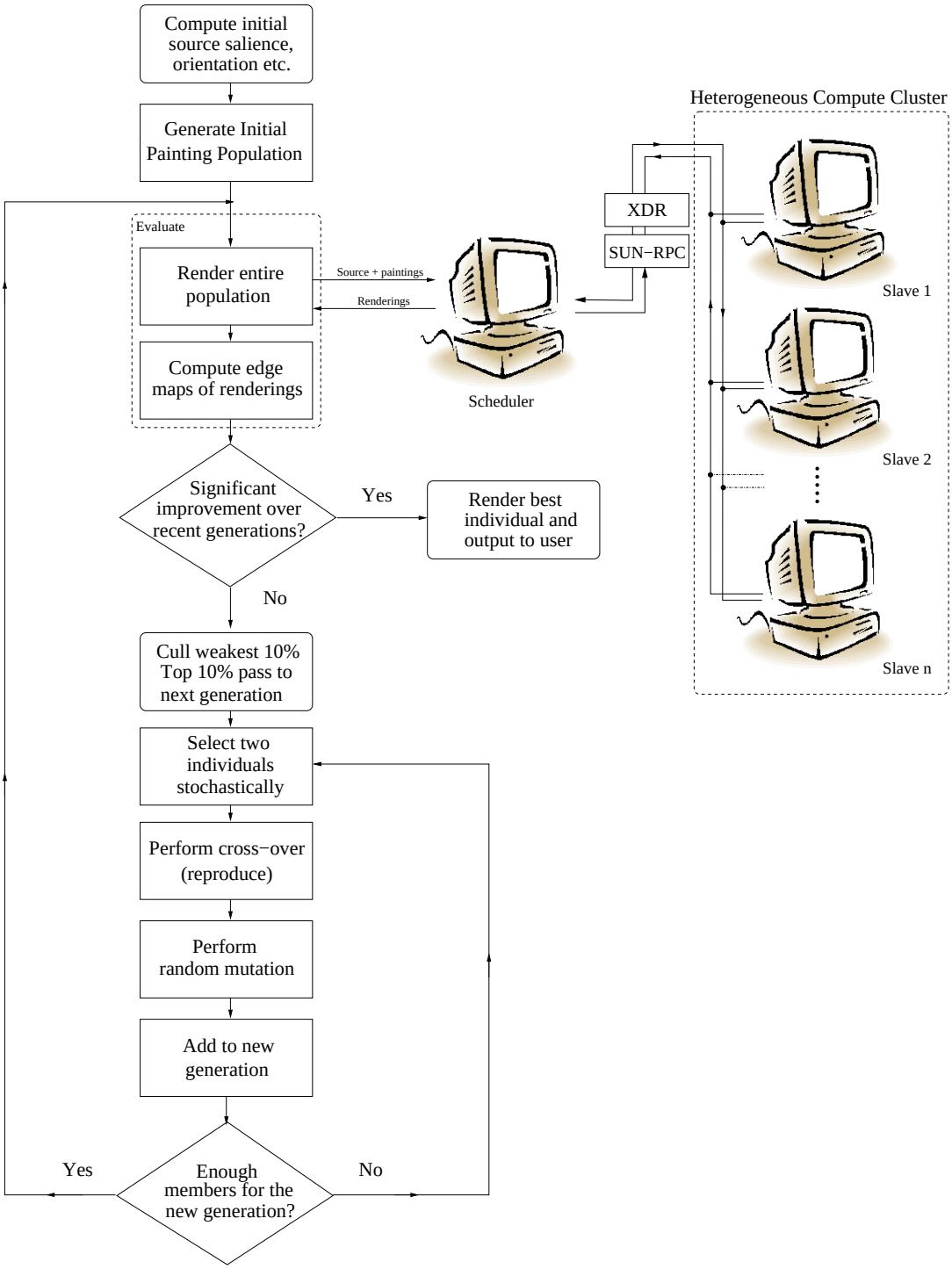


Figure 4-11 Illustrating flow of control in the genetic algorithm. The population evaluation stage is inherently parallel and rendering is farmed out to a distributed compute cluster.

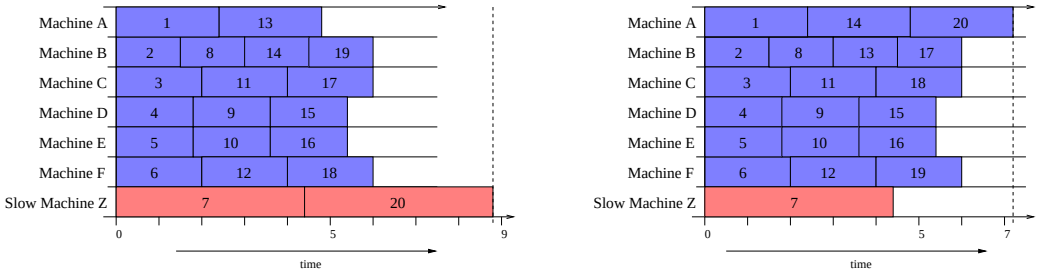


Figure 4-12 Illustrating the scheduling algorithm governing the distributed rendering of a 20 painting generation. Left: The simplistic "first free machine" scheduling strategy becomes time inefficient toward the last few jobs in the population. Right: Sending jobs to machines that we predict will complete them soonest maintains time efficiency over the entire population.

cluster using this strategy (Figure 4-12). Instead, our strategy is to predict which of the machines is likely to finish the pending job first, and then queue the job on that machine. This may involve queueing a job on a faster, busy machine, when a slower machine is idle. Note that for the first iteration of rendering we will not hold an estimate for the speed of machines, and so on this iteration all are assumed to render at equivalent speed.

The typical time to render a fifty painting generation at high (1024×768) resolution is approximately five minutes over six workstations. Relaxation of the painting can therefore take in the order of hours, but significant improvements in stroke placement can be achieved, as can been seen in Figure 4-16 and the accompanying video. The overhead of our task scheduler is low, and processing time falls approximately linearly as further machines of similar specification are added to the cluster.

4.5 Rendering and Results

We have generated a number of paintings to demonstrate application of our algorithm using the source photographs of Figure 4-17. The reader is also referred back to the dragon (Figure 4-1) and sketchy still-life (Figure 4-8) paintings, presented *in situ*. As a note, we have found that running the paintings through a standard sharpening filter [145] can assist presentation of our paintings on the printed page, and have applied such a filter to all paintings presented in this chapter.

The painting of the model in Figure 4-13a converged after 92 generations. Thin precise strokes have been painted along salient edges, while ridges and flats have been painted with coarser strokes. Observe that non-salient high-frequency texture on the rock has been abstracted away, yet tight precise strokes have been used to emphasise

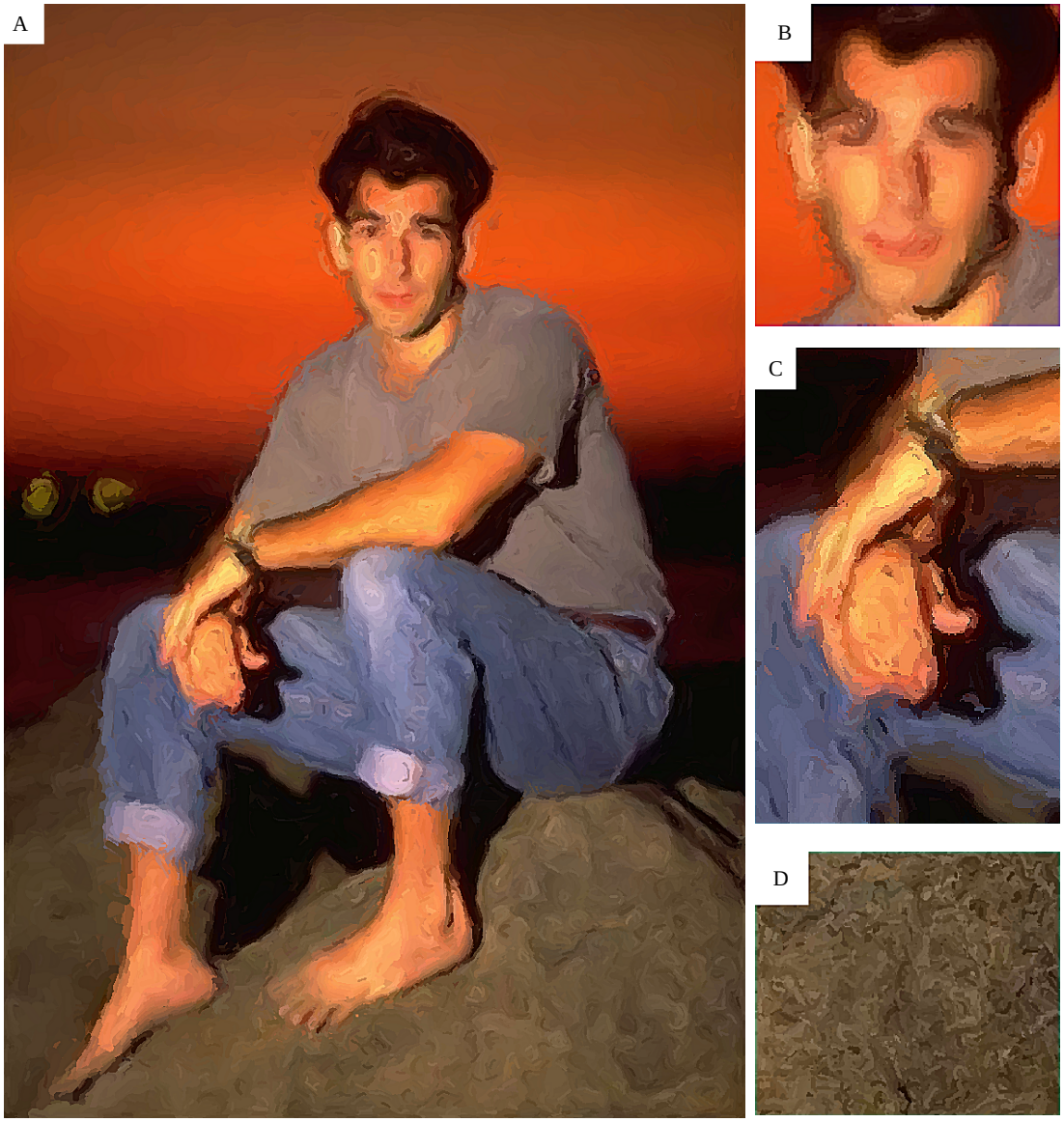


Figure 4-13 Man on rock: (a) final painting after convergence using our proposed method, close-up of hands in (c). (b) example of the face rendered with insufficient emphasis (d) example of rock texture rendered with too great an emphasis. Refer to the text of Section 4.5 for full explanation of (b) and (d), and how our salience adaptive painting avoids such difficulties.

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 techniques would, by contrast, assign both regions equal emphasis. With current techniques, one might globally increase the kernel scale of a low-pass filter [71] or raise thresholds on Sobel edge magnitude [103] to reduce emphasis on the rock (Figure 4-13c). However this would cause a similar drop in the level of detail on the face (Figure 4-13b). Conversely, by admitting detail on the face one would unduly emphasise the rock (Figure 4-13d). In our method, we automatically differentiate between such regions using a perceptual

Figure 4-14 Pickup truck after convergence. Observe salience adaptive emphasis of sign against background in (a). We have manually dampened the salience 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.

Figure 4-15 Sunflowers after convergence. Inset: a sketchy version of the sunflowers in the style of Figure 4-8, prior to relaxation.

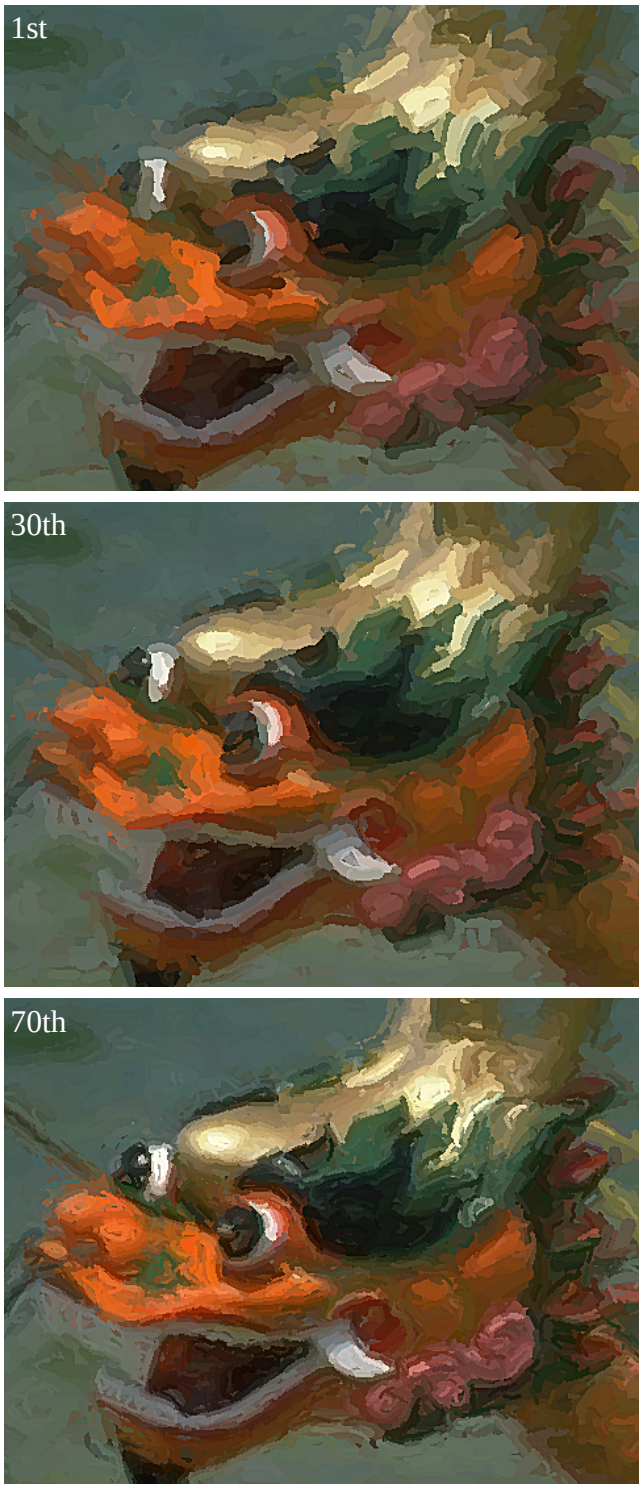


Figure 4-16 Relaxation by genetic algorithm. Detail in the salient region of the "dragon" painting sampled from the fittest individual in the 1st, 30th and 70th generation of the relaxation process. Strokes converge to tightly match contours in salient regions of the image thus conserving salient detail (an animation of this convergence has been included with the electronic supplementary material in Appendix C).

Figure 4-17 Source images used to generate the paintings of Figures 4-13, 4-14, and 4-15.

Figure 4-18 Left: The distinction between salient and non-salient detail can not be made by current image-space painterly techniques which use locally measures such as variance or the Sobel operator (rendered using [103]). All high frequency detail is afforded equal emphasis. Right: "Sheaf-binder" [Van Gogh, 1889]. The majority of Van Gogh's work is characterised by his use of elegant, curved sweeping strokes; our system is capable of producing strokes in ostensibly similar styles (Figure 4-15).

Figure 4-19 Detail from Figure 4-14, region A. Left: Section of the original photograph exhibiting non-salient background texture (shrubbery) and salient foreground (sign-post). Middle: All fine detail is emphasised using existing automatic approaches (here we use [103] as a representative example), which place 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 attempts to mitigate 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.

salience map (Figure 4-4) — 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 4-15 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 salience are given in Figure 4-14. In region A, observe that the salient 'phone sign is emphasised whilst non-salient texture of the background shrubbery is not (also see Figure 4-19 for a enlarged, comparative example). For the purposes of demonstration we have manually altered a portion of salience 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 σ parameter (Figure 4-10, left).

All of our experiments have used populations of fifty 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 con-

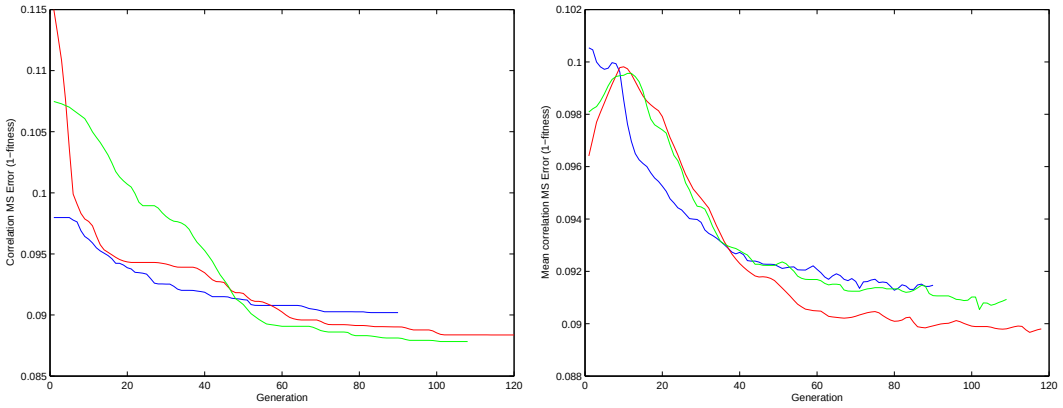


Figure 4-20 Three representative runs of the relaxation process; blue corresponds to the model (Figure 4-13), red the dragon (Figure 4-1) and green the abbey (Figure 4-21). The left-hand graph shows the MSE (inverse fitness) of the fittest individual over time, the right-hand graph shows the same measure averaged over each generation.

vergence still occurs with such population limits, it requires, on average, two to three times as many iterations to achieve. Such interactions are often observed in complex optimisation problems employing genetic algorithms [56]. We conclude that the diversity introduced by our mutation operator (Section 4.4.2) is sufficient to warrant the lower population limit.

During development we 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 lengthly 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 lengthly execution times for which the pay off in terms of quantitative change in the fitness function, and qualitative improvement in aesthetics, was marginal.

4.6 Summary and Discussion

In this chapter we have presented a novel automatic algorithm for creating impasto style painterly renderings from photographs. This work builds upon the pilot single-pass salience adaptive painterly technique introduced in the previous chapter, which

aimed to control emphasis in a painting through use of a global salience map. The principal disadvantage of single-pass algorithms is that strokes are placed only once and then remain fixed; image noise local to stroke location can produce inaccuracies in the rendering. By contrast, relaxation based painting techniques can iteratively tweak stroke parameters to converge the painting toward some procedurally defined "goal" state. Such an approach is of considerable benefit, since in the absence of a specific procedural model of the painting process, we can still approach a specific goal; in effect, we can specify where we want to be, and provide only vague instructions on how to get there.

We drew upon our observations of artistic practice in Chapter 3 to define the degree of optimality for a painting to be measured by the correlation between the salience map of the original image and level of detail within the corresponding painting. To search for this optimal painting, we developed a novel GA based relaxation technique. Our choice of GAs for relaxation was motivated by their superiority to other search techniques for high dimensional problem spaces with many local optima. Furthermore, although it is difficult to explicitly model the complex relationships between stroke parameters during creation of a painting, goal driven stochastic processes such as GAs are known to perform acceptably well without such models. The inherent parallelism of the population evaluation step also permits acceptable performance when combined with distributed computation techniques (Section 4.4.2).

The paintings generated by our algorithm serve to reinforce our argument that a higher level of spatial analysis is of benefit to AR, in terms of enhancing quality of output. The GA relaxation framework and salience measure serve as a more sophisticated means to the same end; that of generating paintings with a focus and emphasis driven by global image importance rather than simple local frequency content. For example, the salience adaptive discrimination between level of detail on the rock, and the model's face (see Figure 4-13), or the sign-pose and shrubbery (see Figure 4-19), would not have been possible using local, frequency driven approaches to AR.

We are not the first to describe relaxation approaches to painterly rendering. As we observe in the review of Section 2.4.1, the discussion of such approaches dates back to Haeberli [62], and the first algorithmic solutions was described in [155]. However the relaxation techniques that exist contain objective functions that specifically aim to maximise the preservation of high frequency detail in the final painting. We discussed the disadvantages of this spatially local approach to rendering in Chapter 3.

Our algorithm operates by compositing curved spline brush strokes, which are fitted

Figure 4-21 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 4.4.1). Original photograph inset, top-left.

by sampling local gradient orientation. We argued that orientation measurements are better modelled as points sampled from a Gaussian distribution, due to the presence of noise (a component of any real image). Our algorithm is provided with a calibrated estimate of the level of noise in the image, and so can produce a stochastic distribution of potential values for measurements, such as orientation, taken from the image. Our relaxation based approach varies stroke parameters within this distribution to converge toward our optimality criterion. This explicit modelling of image noise is a novel contribution to AR. Furthermore, the estimate of noise (σ) need only be approximate and should be greater than or equal to the true level of noise in the image. The penalty for over-estimating noise is an unnecessarily long execution time (Figure 4-10, left), although this often still results in convergence unless a gross over-estimation of σ has been made. The penalty for under-estimating noise is that the painting may not converge to the optimal, since stroke attributes may not be allowed to vary sufficiently to reach the best configuration. Although the placement of strokes is governed primarily by the relaxation process, each stroke has a guaranteed number of hop sites, and hop lengths in inverse proportion to the salience of the image regions they cover. Selection of reasonable values for minimum hop length and count prevents the output of system tending toward photorealism, should such an attraction evolve.

Further contributions of our method were the use of a user trained measure of salience (which the author played a collaborative role in developing, see Section 4.3). The advantages of using this method were two-fold. First, the salient artifacts encountered were classified into one of several trained categories (such as edges, or ridges). This allowed us to automatically differentiate stroke rendering style over the image; a novel contribution. Second, the measure represented a more subjective approach to problem of estimating image salience.

There are a number of potential avenues for development of the GA based relaxation process. An interesting investigation might be to allow users, rather than our objective function, to choose survivors for each iteration of the GA, and to investigate whether such a system might assist the exploration of alternative rendering styles. However this would undoubtedly increase the demands on user time, which are currently very low for our system. We might also choose to introduce novel additional constraints into the relaxation process. For example, explicitly including an information theoretic constraint controlling the density of stroke coverage for a region; this is currently a stochastic decision performed once for each painting generated. More generally, we believe the most interesting direction for development would be to explore alternative objective functions for painting using constraints. A natural extension of the system would be to devise a library of constraints, by studying a range of artists' painting styles. The distributed GA painting system could then be applied to fit a particular painting model (selected by a user) to a photograph, thus painting that photograph in one of a range of established artistic styles. Depending on the commonalities identified between styles, this "paint by model fitting" approach could potentially yield a single versatile system capable of generating many artistic styles within a single parameter space.

Simplifying assumptions have been made in the salience measure. For example, the decision to use spherical JND surfaces in the visibility operator, and the use of a single Gaussian for clustering during rarity were made on the grounds of unattractive computational complexity during clustering. Work continues at Bath to investigate higher level models for the classifier of Section 4.3.3, which take into account neighbouring regions to produce a more context sensitive classification of artifacts.

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 salience maps. Many techniques apply texture mapping to strokes [71, 103, 140], and a bump mapping technique was also proposed in [73]. Highly realistic volume based hairy brush models have recently been proposed [177]

which could be swept along the Catmull-Rom spline trajectories generated by our algorithm. However, we have concentrated primarily upon stroke placement rather than media, 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 salience measures in the production of image-space artistic renderings.

The electronic Appendix C contains high resolution versions of all paintings presented in this chapter (see `/paintings`).