

# Supervised genetic search for parameter selection in painterly rendering

John P. Collomosse

Department of Computer Science, University of Bath, Bath, U.K.  
jpc @ cs.bath.ac.uk

**Abstract.** This paper investigates the feasibility of evolutionary search techniques as a mechanism for interactively exploring the design space of 2D painterly renderings. Although a growing body of painterly rendering literature exists, the large number of low-level configurable parameters that feature in contemporary algorithms can be counter-intuitive for non-expert users to set. In this paper we first describe a multi-resolution painting algorithm capable of transforming photographs into paintings at interactive speeds. We then present a supervised evolutionary search process in which the user scores paintings on their aesthetics to guide the specification of their desired painterly rendering. Using our system, non-expert users are able to produce their desired aesthetic in approximately 20 mouse clicks — around half an order of magnitude faster than manual specification of individual rendering parameters by trial and error.

## 1 Introduction

Techniques for processing photographs into artwork have received considerable attention in recent years and comprise a rapidly developing branch of computer graphics known as *image based non-photorealistic rendering* (NPR). Perhaps the most well studied NPR problem is that of painterly rendering; the automated stylisation of imagery to give a hand-painted appearance. A number of these algorithms now exist capable of emulating a variety of styles, such as water-colour [1] and oil [2–7].

Although the output of contemporary painterly rendering algorithms is often of high aesthetic quality, the usability of such algorithms is impeded by the plethora of low-level parameters that must be set in order to produce acceptable output. Many of these parameters are data dependent, for example determining the scale of image features to paint [3, 4]. The presence and fine-tuning of such parameters is necessary to retain generality of the algorithm, but can be difficult for the non-expert user to achieve. Furthermore, some algorithms [3, 6] seek to emulate a broad range of artistic styles using additional user configurable parameters. For example, [3] varies brush size, colour jitter, and stroke length to interpolate between pseudo “expressionist” and “pointillist” styles. Often these parameters can be time consuming to set — both due to their number, and due

to their low-level nature, which can make them non-intuitive for inexperienced users to manipulate when aiming for a conceptually higher level effect (e.g. a gloomy painting, or an energetic, cheerful composition). Moreover, parameters can interact in complex ways leading to emergent behaviour within the painting that the user may not expect or understand. The end result is often a slow, iterative trial and error process before the user is able to instantiate their desired results.

This paper presents a solution to the problem of NPR parameter selection by framing the task as a goal-centred evolutionary search to be solved using a Genetic Algorithm (GA). We do not wish to eschew interaction, for this is often where artistic creativity is expressed when applying NPR algorithms (such techniques are better regarded as tools, rather than black-box processes for creating artwork). Instead we draw inspiration from Sims [8] who adopted user supervision in his evolutionary artistic processes. Our system iteratively evolving a population of painterly renderings towards the user’s aesthetic ideal, and on each evolutionary cycle presenting a sample of the population to the user for fitness evaluation. The user’s evaluation affects the natural selection phase of the GA, and so affects the composition of subsequent generations of paintings. Similar architectures have also been applied in a other computer graphics domains; for example recent work facilitating the interactive evolution of pixel-vertex shaders [9] and animated screen-savers [10]. To facilitate timely interaction in our system we make use of a fast, segmentation based algorithm for painterly rendering. This algorithm not only draws upon existing painterly rendering literature (encapsulating many of the styles available using existing techniques), but also draws upon colour psychology to allow tonal variations that influence the emotional context or “mood” of the painting. This is achieved by mapping HSV colour transformations on to Russell’s 2D “pleasure-arousal” emotional state space [11], and harnessing emotional state as a high level NPR parameter for painterly rendering [12].

The remainder of the paper is organised as follows. We begin by briefly surveying existing painterly techniques in Section 1.1. We then give detailed descriptions of our painting algorithm (Section 2) and interactive GA search (Section 3). We conclude with a gallery of results and discussion in Section 4.

## 1.1 Related Work

The majority of image based painterly rendering algorithms adopt the stroke-based rendering paradigm, generating paintings by compositing ordered lists of virtual “brush strokes” on a 2D virtual canvas. The development of such algorithms arguably began to gain momentum with Haeberli’s semi-automatic “impressionist” painting system [13]. In Haeberli’s system, stroke attributes such as colour or orientation were sampled from a source photograph whilst stroke size, shape and compositing order was set interactively by the user. Litwinowicz [2] was the first to propose a fully automated 2D painting algorithm, again focusing upon the impressionist style. Paintings were synthesised by aligning small

rectangular strokes to Sobel edge gradients in the image and stochastically perturbing the colour of those strokes. Hertzmann later proposed a “coarse to fine” approach to painting using curved  $\beta$ -spline strokes. Spline control points were extracted by hopping between pixels in directions tangential to Sobel edges. The process operated at several discrete spatial scales, concentrating stroke detail in high frequency areas of the image. The  $\beta$ -splines used in this technique were later extended to active contours, enabling a relaxation based approach to curved stroke painting [14]. Other early painterly rendering algorithms such as [15] and [4] also made use of local image processing operators; placing strokes according to variance measures within a local window.

Our approach contrasts with these early painterly rendering algorithms in that we operate using multi-scale segmentation only, substituting image gradient and variance measures for region shape properties to guide stroke placement. Among the first to propose the use of segmentation algorithms for painting were Gooch *et al.* [5], who placed strokes along medial axes of segmented regions to produce painterly artwork. The benefits of their approach included a significant reduction in the number of brush strokes whilst preserving fine detail in the rendering. Segmentation was also used by [16, 17] to produce painterly abstractions. Notably these systems used a human gaze tracker to correlate level of detail in the painting with perceptually salient detail in the source image. An automatic system for salience adaptive painting, driven by machine learning rather than run-time interaction, was recently presented by Collomosse and Hall [7]. This work also applied evolutionary search techniques to NPR, harnessing GAs as a relaxation mechanism for automatically controlling level of detail in paintings within a single artistic style. By contrast we here apply GAs for style selection using interactive aesthetic evaluation, and as such, our work is also closely aligned with algorithms encompassing a range of visual styles selectable via user parameterisation. Hertzmann [3] claims expressionism, pointillism, impressionism and “abstract” styles through the variation of low level parameters such as stroke length. Similarly low-level parameters are used to tune the visual style of paintings in [6]. We have encompassed a similar gamut of rendering styles within our painterly framework.

## 2 Painterly Rendering Algorithm

In this section we briefly describe our fast multi-resolution technique for stylising photographs at interactive speeds. The scope of this paper is such that we have focused on the parameterisation of the algorithm and interested readers are directed to [12] for a more detailed description. The algorithm accepts a 2D photograph as input, and outputs a 2D painterly rendering of that photograph — the visual style of which is a function of eight user-configurable scalar parameters,  $p_{1..8}$  (Figure 1) which are output by the evolutionary search step (Section 3).

We begin by creating a colour band-pass pyramid segmentation of the source image by applying the EDISON [18] algorithm at several spatial scales. This segmentation is computed only once, during system initialisation, to facilitate

| Param | Description       |
|-------|-------------------|
| $p_1$ | Colour jitter     |
| $p_2$ | Maximum hop angle |
| $p_3$ | Region turbulence |
| $p_4$ | Colour (pleasure) |
| $p_5$ | Colour (arousal)  |
| $p_6$ | Stroke jaggedness |
| $p_7$ | Stroke undulation |
| $p_8$ | Region dampening  |

**Fig. 1.** Summary of the eight rendering parameters  $p_{1..8}$  used to control visual style in our painting algorithm.

real-time interaction during the search process. To produce a painting, the pyramid layers are rendered in coarse to fine order. Segmented regions within each layer are painted using a combination of “interior” and “boundary” strokes; as we explain in the next subsection. For each layer, the “interior” strokes of all regions are first rendered, followed by the “boundary” strokes of all regions.

## 2.1 Interior and Boundary Stroke Placement

Brush strokes are formed using Catmull-Rom piecewise cubic splines, the control points of which are computed from the binary mask of each region as follows.

**Interior Strokes.** The interior of a segmented region is first filled using a modified boundary-fill algorithm that paints strokes tangential to the region’s principal axis, obtained by computing the eigenvectors of pixel coordinates inside the region. Lines parallel to the principal axis are traversed, and strokes are started and terminated as region boundaries are encountered. The spacing between these traversal lines is proportional to stroke thickness. As each stroke is placed, control points are distributed uniformly over the stroke’s length, and jittered via small translations to disguise the regularity of the stroke placement process. Stroke colour is set to the mean colour under the stroke, computed from the source image. This colour is subject to random perturbation, the magnitude of which forms one parameter of the rendering process, written  $p_1$ . Stroke thickness is set on a per region basis, in proportion to region area. In the case of very large regions, thickness is capped and strokes are painted horizontally (after [19]) to preserve natural appearance.

**Boundary Strokes.** The boundary of the segmented region is vectorised to produce a closed polygon. Points on the polygon are visited in order, and added to an initially empty “working set”. Upon each point’s addition, we sum the distance between all points in the working set to a line drawn between the first and last points in that set. If the distance is above a threshold (or no further points remain in the chain code), we output the most recently added point as a stroke

**Fig. 2.** Left: Russell’s 2D pleasure-arousal space used to parameterise tonal variation in the painting. Right: False colour schematic illustrating the various colour transformations performed within regions of the pleasure-arousal space. Functions  $G$ ,  $U$ ,  $L$ ,  $D$ ,  $T_1$  and  $T_2$  are defined in Section 2.2.

control point. The working set is then emptied. A brush stroke is terminated, and a new stroke started, when the angle between adjacent control points rises above a preset threshold. In practice this threshold governs the typical length of brush strokes, and we allow this to vary between  $0 - 50^\circ$  as another of our rendering parameters (written  $p_2$ ). The stroke must also be terminated if the colour of a new control point differs significantly from the mean colour of those already present in the stroke. Stroke thickness and colour are set as with the interior stroke placement process.

## 2.2 Rendering Parameters

In addition to the two rendering parameters ( $p_1, p_2$ ) governing stroke placement (Section 2.1), we also incorporate the following six parameters that allow modulation of the painting’s visual style. All parameters are summarised in Figure 1.

**Region Turbulence.** Flat expanses within paintings, for example sky or water, may be depicted in a variety of artistic styles. Our system encompasses a gamut of rendering styles ranging from the calm, serene washes of a watercolour to the energetic swirls of a Van Gogh oil or the chaotic strokes of a Turner seascape. We introduce similar effects by repeatedly performing boundary stroke placement (Section 2.1) subjecting region masks to morphological erosion prior to each iteration. The number of iterations is proportional to rendering parameter  $p_3$ . This has the effect of allowing boundary strokes to grow into the interiors of regions in an unstructured manner, so breaking up flat expanses in the painting.

**Tonal Variation.** Certain combinations of colours can evoke particular emotions, so helping to convey a particular mood to a composition. We have identified a number of cues from colour psychology, and mapped these to regions of Russell’s 2D pleasure-arousal emotional space [11] — see Figure 2 (left). By specifying an emotional state (a point  $(p_4, p_5)$  in this space defined by two further rendering parameters), we allow the user to interactively vary the emotional ambiance or “mood” of the composition. Wright and Rainwater [20] have found the notion of happiness (pleasantness) to be primarily dependent on colour brightness (luminance), and to a lesser degree on saturation. Intuitively arousal corresponds to colour saturation, but can also be linked to hue. Wright and Rainwater’s study has shown calmness to be blue-correlated [20], but according to Mahnke blue may also suggest depression and cold [21].

We have defined a number of transfer functions that operate upon hue, saturation and luminance as a mechanism for instantiating the colour heuristics we

have distilled from the literature [20, 21]. The complex psychological theories underpinning colour and emotion generate non-linear mappings of hue, saturation and luminance variation to the pleasure-arousal space. We approximate these piecewise with a collection of linear transfer functions — different functions are applied in each of six regions of the space. Figure 2 (right) illustrates the boundaries of these regions, and the transfer functions used over the pleasure-arousal space. Functions  $G(x)$  and  $U(x)$  correspond to greying and un-greying (scaling saturation in proportion to  $x$ ), while  $D(x)$  and  $L(x)$  correspond to lightening and darkening (scaling luminance in proportion to  $x$ ). The operation of the latter function is capped for “boundary” brush strokes to prevent bleaching of fine detail. Care is taken in blending the constants of proportionality to prevent visible discontinuities near the boundaries defined over the pleasure-arousal space. Functions  $T_1(x)$  and  $T_2(x)$ , indicated in Figure 2, are two special cases that encode hue variation consistent with aroused displeasure (anger) and apathetic displeasure (depression). Hue is manipulated via an RGB space transformation prior to saturation and luminance manipulations. In the former case  $T_1(x)$ , predominantly red colours are reddened and green (associated with calm) is reduced (in proportion to  $x$ ). These effects combine with the saturation and luminance transformations already present to produce the combination of aroused reds and dismal darks that appear in psychological literature in association with anger. In the latter case  $T_2(x)$  we increase the blue in proportion to  $x$  to generate a monotonous shift into the blue spectrum, associated with sadness and calm. Colours are also desaturated and darkened in accordance with transformations already present in that quadrant of the space.

**Brush Stroke Style.** We have introduced two parameters to control variation of stroke style in our system. These afford the user some control over stroke accuracy ( $p_6$ ) and angularity ( $p_7$ ) in the painterly rendering.

When rendering a brush stroke we create an arc-length parameterisation over the piecewise Catmull-Rom spline that smoothly interpolates its control points. To enhance the angularity or “jaggedness” of strokes we create an additional linear interpolation over the control points using the same arc-length parameterisation. Lineally interpolating between these two functions, in proportion to  $p_7$ , yields our desired style of stroke. We then introduce inaccuracies into the stroke placement process by inducing undulations in the trajectory of the stroke. Whilst rendering, we translate points on the stroke along their normal vectors — the distance a particular point is moved is set by a periodic function with frequency and amplitude proportional to  $p_6$ . Finally, we introduce a further parameter ( $p_8$ ) to dampen the effects of undulation ( $p_6$ ) on interior strokes. This can lead to visually chaotic stroke placements in the backgrounds of paintings that may or may not be desirable depending on the user’s intended visual style.

### 3 Evolutionary Search for Parameter Selection

Given this parameterised rendering framework, the painting process is reduced to a search for the point in our parameter space  $[p_1 p_2 \dots p_8] \in \mathbb{R}^8$  that corresponds to a painting expressing the target aesthetics of the user. Our system adopts a genetic algorithm (GA) search strategy. GAs are often cited as appropriate for exploring high dimensional parameter spaces as large regions of problem space can be covered quickly, and local minima (e.g. arising due to interactions between painting parameters) are more likely to be avoided [22, 23]. We now describe the initialisation and iterative stages of our GA search.

#### 3.1 Initialisation

We begin by initialising a fixed size population of individuals. We have opted for a population size of 1000 individuals, determined empirically to be a suitable trade-off between diversity the real-time processing constraints of our system. Each individual contains eight normalised scalar values that comprise the genotype of a particular painting. These values are seeded randomly in the initial generation.

#### 3.2 Iterative Search

Genetic algorithms (GAs) simulate the process of natural selection by breeding successive generations of individuals through cross-over, fitness-proportionate re-production and mutation. In our implementation we terminate this iterative process when successive improvements in fitness become negligible (the change in both average and maximum population fitness over a sliding time window fall below a threshold). We now describe a single iteration of this process.

**Interactive Evaluation** The first step in each iterative cycle is population evaluation. We wish to measure the proximity of each individual’s phenotype to the user’s “ideal” aesthetic. Specifically we require a mapping  $M([p_1 p_2 \dots p_8]) \mapsto f \in \mathbb{R}$  where  $f$  is a normalised fitness score; higher values correspond to aesthetically superior paintings. As our aim is to assist the user in style specification it is not possible to write an automatic function for  $M(\cdot)$ . Our objective is therefore twofold. First, to estimate the mapping function  $M(\cdot)$  through user interaction. Second, to search for the point  $\underline{p} \in \mathbb{R}^8$  such that:

$$\underline{p} = \underset{i}{\operatorname{argmax}} [M(i)] \quad (1)$$

Our approach is to sparsely evaluate  $M(\cdot)$  over a subset of the population, and use this data to extrapolate the behaviour of  $M(\cdot)$  over the entire population. We have designed a simple user interface, allowing us to prompt for the fitness of a given individual drawn from the population (so obtaining a sparse domain sample of  $M(\cdot)$ ). The user is supplied with a graduated colour bar, and asked to rate the aesthetics of the painting rendered from a given individual on a

**Fig. 3.** Snapshot of the interactive evaluation screen. The user is presented with thumbnails of the highest ranking 9 paintings and asked to rate one by clicking with the mouse. Depending on the horizontal location of the click within the thumbnail, a fitness score  $[-1,1]$  is assigned to the chosen rendering. This snapshot shows images from the first generation of paintings — hence the diverse selections available.

continuous scale spanning red (bad), amber (neutral) and green (excellent) — see Figure 3. Manually evaluating the entire population on each iteration would be impractical, and to reduce user load we request evaluation of only one individual per generation. The user is presented with the nine fittest individuals from the previous iteration, and asked to rate the individual that they feel most strongly about. Note that in the first iteration individuals are deemed to exhibit equal fitness (see equation 2) and so are chosen from the population at random.

We use a Gaussian “splatting” technique to encode the results of our sparse user interactions, and transform these into a continuous estimate for  $M(\cdot)$ . Each time a user evaluates an individual we obtain a point  $\underline{q}$  and a user fitness rating  $U(\underline{q}) = [-1, 1]$ . These data are encoded by adding a Gaussian to a cumulative model, built up over successive user evaluations. Each Gaussian distribution is centred at point  $\underline{q}$ , and multiplied by the factor  $U(\underline{q})$ . We assume the integral under the Gaussian to be well approximated by unity in space  $\mathfrak{R}^8 \in [0, 1]$ , and so infer the continuous function  $M(\cdot)$  as:

$$M(\underline{p}) = 0.5 + \begin{cases} 0 & \text{if } N = 0, \\ \frac{1}{2N} \sum_{i=1}^N U(\underline{q}_i) G(\underline{p}, \underline{q}_i, \sigma) & \text{otherwise} \end{cases} \quad (2)$$

where  $\underline{p}$  is an individual to be evaluated in the current generation,  $\underline{q}_i$  are individuals evaluated by the user in the previous  $N$  iterative cycles, and  $U(\underline{x})$  is the user’s score of a given genotype  $\underline{x}$ . The function  $G(\underline{x}, \underline{\mu}, \sigma)$  denotes a Gaussian distribution with mean  $\underline{\mu}$  and standard deviation  $\sigma$ , evaluated at  $\underline{x}$ . The standard deviation  $\sigma$  governs the locality in problem space over which a single user evaluation holds influence. We have used the value  $\sigma = 0.1$  for all of the results presented here. Equation 2 provides us with an estimate for  $M(\cdot)$  defined over the entire problem space, which we then apply to evaluate the whole population.

**Selection and Propagation** Once the current population has been evaluated, pairs of individuals are selected and bred to produce the next generation of painting solutions. Parent individuals are selected with replacement, using a stochastic process biased toward fitter individuals. A single offspring is produced from two parents by way of stochastic cross-over and mutation operators. Each of the eight parameters that comprise the genome of the offspring has an equal chance of being drawn from either parent. Mutation is implemented by adding a random normal variate to each of the eight parameters. These variates have standard deviations of 0.1, i.e. 97% of mutations will produce less than  $\pm 0.3$  variation in a particular rendering parameter.

**Fig. 4.** Population statistics corresponding to the evolution of the paintings shown in Figure 5c<sub>1</sub> (blue, dotted), Figure 5c<sub>2</sub> (red, dashed), Figure 5c<sub>3</sub> (green, solid) respectively. The + symbol indicates algorithm termination. \* indicates a negative fitness rating from the user.

**Fig. 5.** A gallery of painterly renderings produced by our system, original images inset.

## 4 Results and Conclusion

We have tested our system on a wide range of images, and give representative examples in Figure 5. In Figures 5a we show single photograph (*BIGBEN*) evolved into “abstract” and “expressionist” styles reminiscent of those presented in [3, 24]. Convergence took 15 and 17 mouse clicks respectively — less than one minute of user time. Furthermore, our segmentation based painting algorithm did not require technical parameters (e.g. scale or low-pass kernel size [3]) to be specified explicitly by the user. Figures 5e<sub>1</sub>, e<sub>2</sub> give examples of further painterly styles, contrasting two different stroke placement styles encompassed by our system. The first is reminiscent of the “impressionist” style paintings generated by [2], the latter the impasto style oil paintings generated by [7]. Figure 5b gives a further examples of the broad classes of image handled by our system. Figure 5c demonstrates a single photograph (*DRAGON*) evolved into three distinct visual styles. A non-expert was asked to use our system to create paintings depicting high-level concepts such as anger (Figure 5c<sub>1</sub>), cheerfulness (Figure 5c<sub>2</sub>) and despair (Figure 5c<sub>3</sub>). Graphs recording the mean population fitness, and standard deviation (diversity) during the evolution of these paintings are also supplied in Figure 4. Convergence took between 20 to 25 generations before the termination criteria was triggered. In Figure 4 we have forced evolution to continue beyond 30 generations, however improvements in mean fitness beyond the automated termination point are negligible.

To evaluate the usability of our system we developed an alternative low-level interface using sliders to independently control  $p_{1..8}$ . Users were asked to produce identical renderings to those previously generated using our GA. Users were usually able to reproduce results, but required five or six minutes of experimentation (and several hundred mouse clicks) before doing so — approximately five times longer than when using our GA goal-based search.

When working with our system, we have found that users will often focus on a particular aesthetic property of the painting and focus on the improvement of that property. For example, users might address the issue of edge detail over, say, the style of the in-filled background. Often these properties have no direct mapping onto individual rendering parameters, providing some explanation of the timing improvements of a top-down goal seeking approach to style selection over a bottom up configuration of low-level painting parameters. The impact of this behaviour can be observed in the graph of Figure 4 (left). Gradual increases in painting “fitness” are observed over time, interrupted by short-lived dips.

These dips become less pronounced as the generation count increases. We have found the presence of dips to correlate with the issuing of negative ratings by users; typically these are issued when a user has refined one aspect of the painting to their liking, and begun to address a further aspect of the composition that they had so far neglected. By neglecting refinement of the latter aspect, a false representation of the user’s “fitness function”  $M(\cdot)$  (see Section 3.2) has been conveyed to the system and encoded in the Gaussian distribution model. This requires user correction, often in the form of rating penalisation.

Throughout our work we have assumed the user has an ideal painting in mind, and wishes to express instantiate that ideal through NPR. An alternative application of our system might be in style exploration, where the user has no well-developed goal state in mind. Early indications are that the guided search provided by our system may be suitable for such activities. However if the user substantially revises their aesthetic ideals late in the search, the reduced population diversity can require tens of iterations before the user is able to explore new regions of the problem space. If our system were to be used in this manner, we would suggest increasing the standard deviation of the variates used during mutation to maintain population diversity further into the search.

## 5 Acknowledgements

We are grateful for the assistance of users who participated in evaluation of the GA interface, and also for the contribution of Maria Shugrina in development of the colour model.

## References

1. Curtis, C., Anderson, S., Seims, J., Fleischer, K., Salesin, D.H.: Computer-generated watercolor. In: Proc. ACM SIGGRAPH. (1997) 421–430
2. Litwinowicz, P.: Processing images and video for an impressionist effect. In: Proc. ACM SIGGRAPH, Los Angeles, USA (1997) 407–414
3. Hertzmann, A.: Painterly rendering with curved brush strokes of multiple sizes. In: Proc. ACM SIGGRAPH. (1998) 453–460
4. Shiraishi, M., Yamaguchi, Y.: An algorithm for automatic painterly rendering based on local image approximation. In: Proc. ACM NPAR Sympos. (2000) 53–58
5. Gooch, B., Coombe, G., Shirley, P.: Artistic vision: Painterly rendering using computer vision techniques. In: Proc. ACM NPAR Sympos. (2002) 83–90
6. Hays, J., Essa, I.: Image and video based painterly animation. In: Proc. ACM NPAR Sympos. (2004) 113–120
7. Collomosse, J.P., Hall, P.M.: Genetic paint: A search for salient paintings. In: proc. EvoMUSART (at EuroGP), Springer LNCS. Volume 3449. (2005) 437–447
8. Sims, K.: Artificial evolution for computer graphics. In: Proc. ACM SIGGRAPH. Volume 25. (1991) 319–328
9. Ebner, M., Reinhardt, M., Albert, J.: Evolution of vertex and pixel shaders. In: LNCS (in Proc. EuroGP’05). Volume 3447., Springer-Verlag (2005) 261–270
10. Draves, S.: The electric sheep screen-saver: A case study in aesthetic evolution. In: LNCS (in Proc. EvoMUSART’05). Volume 3449., Springer-Verlag (2005) 458–467

11. Russell, J.A.: Reading emotion from and into faces: Resurrecting a dimensional-contextual perspective. In Russel, J.A., Fernández-Dols, J.M., eds.: *The Psychology of Facial Expression*. Cambridge University Press (1997) 295–320
12. Shugrina, M., Betke, M., Collomosse, J.P.: Empathic painting: Interactive stylization using observed emotional state. In: *Proc. ACM NPAR Sympos.* (2006)
13. Haeberli, P.: Paint by numbers: abstract image representations. In: *Proc. ACM SIGGRAPH*. Volume 4. (1990) 207–214
14. Hertzmann, A.: Paint by relaxation. In: *Proc. Computer Graphics Intl. (CGI)*. (2001) 47–54
15. Treavett, S., Chen, M.: Statistical techniques for the automated synthesis of non-photorealistic images. In: *Proc. 15<sup>th</sup> Eurographics UK Conference*. (1997) 201–210
16. DeCarlo, D., Santella, A.: Abstracted painterly renderings using eye-tracking data. In: *Proc. ACM SIGGRAPH*. (2002) 769–776
17. Santella, A., DeCarlo, D.: Visual interest and NPR: an evaluation and manifesto. In: *Proc. ACM NPAR Sympos.* (2004) 71–78
18. Christoudias, C., Georgescu, B., Meer, P.: Synergism in low level vision. In: *16<sup>th</sup> Intl. Conf. on Pattern Recognition*. Volume 4. (2002) 150–155
19. Koliopoulos, A.: Image segmentation for stylized non-photorealistic rendering and animation. Master’s thesis, Univ. Toronto (2005)
20. Wright, B., Rainwater, L.: The meaning of colour. *Journal of General Psychology* **67** (1962)
21. Mahnke, F.: *Color, Environment, and Human Response*. Van Nostrand Reinhold (1996)
22. de Jong, K.: Learning with genetic algorithms. *Machine Learning* **3** (1988) 121–138
23. Holland, J.: *Adaptation in Natural and Artificial Systems*. 1<sup>st</sup> edn. Univ. Michigan Press (1975) ISBN: 0-472-08460-7.
24. Hertzmann, A., Perlin, K.: Painterly rendering for video and interaction. In: *Proc. ACM NPAR Sympos.* (2000) 7–12