Emotionally Aware Automated Portrait Painting

Simon Colton
Department of Computing
Imperial College London
sgc@doc.ic.ac.uk

Michel F. Valstar
Department of Computing
Imperial College London
mvalstar@doc.ic.ac.uk

Maja Pantic
Department of Computing
Imperial College London
m.pantic@imperial.ac.uk

ABSTRACT

We combine a machine vision system that recognises emotions and a non-photorealistic rendering (NPR) system to automatically produce portraits which heighten the emotion of the sitter. To do this, the vision system analyses a short video clip of a person expressing an emotion, then tracks the movement of facial features and uses this tracking data to analyse which emotion was expressed and what the temporal dynamics of the expression were. The image where the emotion is expressed strongest, the location of the facial features in that image and a keyword describing the emotion detected are passed to the NPR software. This keyword is used to choose appropriate (simulated) art materials, colour palettes, abstraction methods and painting styles, so that the rendered image may heighten the emotion being expressed. We describe the vision and rendering systems and their combination, and provide examples of portraits produced in this emotionally aware fashion.

Categories and Subject Descriptors

H.1.2 [**User/Machine systems**]: Human information processing

General Terms

Automatic facial expression recognition, artificial creativity

Keywords

non-photorealistic rendering, machine vision, affective computing, emotion detection, computational creativity.

1. INTRODUCTION

We are interested in the notion of computational creativity, in particular the question: under what circumstances (if any) is it appropriate to describe the behaviour of a computational system as creative. Visual art is a domain where human creativity flourishes, so Non-Photorealistic Rendering (NPR) – where art materials and artistic styles are simulated – would appear to be an ideal domain in which to test computational models of creativity. Unfortunately, however, NPR researchers have tended to eschew the potential for software to act as creative collaborators in art projects, opting instead to build systems which merely enhance the efficiency/creativity of users. In fact, due to its close relationship with human creativity, some authors seem almost apologetic about simulating artistic techniques, for instance [19] state unequivocally that:

> "Simulating artistic techniques means also simulating human thinking and reasoning, especially creative thinking. This is impossible to do using algorithms or information processing systems" (Page 113).

We disagree with this assessment, and we are currently building an automated painter (called The Painting Fool) which we hope will eventually be accepted as a creative artist in its own right. We have taken the approach of identifying what appear to be some necessary high-level conditions for creative behaviour, and improving The Painting Fool to meet these conditions. One such condition is that the software exhibits appreciation in its behaviour, and we describe here how we have addressed the issues of the software appreciating both its subject matter and the way its rendering choices affect the picture it produces. To start addressing these issues, we added an expert system to The Painting Fool which takes a high-level description about either the nature of the picture to paint, or the nature of the subject matter, and chooses from an extensive range of abstraction, colouring and rendering methods which – taken together – specify an artistic style appropriate to the high-level description.

Concentrating on portraiture, we have integrated The Painting Fool with a machine vision system which is able to detect the emotion being expressed in short video clips of people smiling/frowning/etc. This is achieved by automatic feature extraction, action unit analysis and emotion recognition methods. The vision system passes to The Painting Fool: (a) the [apex] image from the video clip where the emotion is most strongly expressed, (b) the location of the facial feature tracking points, and (c) one of six keywords which identifies the emotion being expressed. We trained the expert system in The Painting Fool in such a way that it can map the emotion keyword onto one of a number of artistic styles which may result in the emotion being heightened in the rendered image. For instance, if the vision system correctly identified the keyword 'sadness' to describe the sitter, The Painting Fool would chose to simulate pastel strokes with muted colours to heighten the melancholy expressed in the portrait it painted. It would also use the feature tracking information supplied by the vision system to pay particular emphasis to the facial features of the sitter. In this way, our contribution to computer graphics has been to build the first (to the best of our knowledge) automated portrait painting system which works live from a video feed and uses both the sitter's facial features and their emotion to improve its output.

This paper is organised as follows. In section 1.1 we describe re-

DIMEA'08, September 10-12, 2008, Athens, Greece.

lated work. In section 2, we describe the segmentation and rendering techniques implemented in The Painting Fool and how we trained its expert system to control these techniques. In section 3, we describe the various stages by which the vision system determines the emotion being expressed in a video clip, and in section 4, we describe how we combined the vision and rendering systems to produce portraits. In section 5, we conclude and present some directions for future work.

1.1 Related work

There is a great deal of work related to both emotion detection and non-photorealistic rendering, and we cover only the most relevant here. Our vision system analyses facial expressions by first recognising which facial muscles are activated. These facial muscle actions are defined by the Facial Action Coding System (FACS) as Action Units (AUs, see section 3). In the area of automatic AU recognition, a number of successful approaches have been proposed fairly recently. Previous work includes detection of 16 AUs using lip tracking, template matching and Artificial Neural Networks (ANNs) [20], detecting 20 AUs occurring alone or in combination by using temporal templates generated from input face video [23] and detection of 18 AUs using wavelets, AdaBoost and Support Vector Machines [2]. Our AU analysis system can detect the most AUs (22), with equal or greater classification accuracy than these other systems. The problem of basic emotion detection itself is older than the AU recognition problem and many researchers consider it to be solved under the constrained conditions that apply for our system (i.e., posed expression, frontal-view, controlled lighting, and little head motion). Recent work on emotion detection has used ANNs [7], Support Vector Machines (SVMs) [1, 2], and Bayesian Networks [25]. Kotsia and Pitas have recently proposed a method that uses geometric features and multiclass SVMs to detect both AUs and emotions [12]. For a thorough overview of the work done on AU and emotion detection from still images or face video, see [13, 21]. Facial expression temporal dynamics (e.g., the timing and duration of facial actions) have been shown to be a critical factor for the interpretation of human behaviour [21]. Despite this, very few projects have been undertaken to explicitly analyse the temporal dynamics of facial expressions with machine vision techniques. The only existing works are described in [23].

The Painting Fool system performs image segmentation and simulates the usage of artistic media. These are very standard non-photorealistic rendering techniques, so we refer the reader to standard textbooks such as [19]. There are a number of projects which combine machine vision with painting programs. For instance, the program described in [3] uses face detection tools to identify the location of facial features in portraiture, and then highlights these areas during the rendering process. Similarly, [4] use image salience in still images to determine important regions of images in order to produce the rendered image with emphasis in these places. The work that resembles ours most is that of [18]. In that paper, the authors propose a system that analyses video data of users to guide the painting style employed. Their system attempts to detect AUs first, and to map the AUs to an emotion description. In contrast to our approach, they do not attempt to recognise the six basic emotions but instead use Russell's 2-dimensional pleasure/arousal model. This approach limits the number of possible effects that the painting sub-system can employ. Also, the authors base their approach on FACS, but no experimental results for their AU or emotion recognition sub-systems are reported. A major difference between their project and ours is that their aim was to investigate a novel human-computer interaction method, whereby the emotion of the user of a graphics package can be used to control that package to paint non-portrait images. In contrast, we used emotion detection to enable the system to paint an improved portrait of the sitter/user.

2. THE PAINTING FOOL NPR SYSTEM

The Painting Fool is a non-photorealistic rendering (NPR) system which is given a digital image which may or may not have been annotated with the boundaries of scene-elements within the image (for instance, the user might choose to provide details of where the eyes, nose and mouth of a person are in a digital image). The Painting Fool produces an artistic rendering of the image in a two-stage process. Firstly, it segments the entire image and separately segments the scene-elements producing a list of segmentations, as described in 2.1. Secondly, it takes each shape in each segmentation and renders it with simulated art materials such as acrylic paints, pastels, pencils, etc., as described in 2.2. We do not claim that our NPR techniques are particularly novel. However, we provide details of them in order to describe the parameters which guide the process that have been used to train an expert system for choosing artistic styles, as described in 2.3.

2.1 Segmenting Images

The segmenting of images is controlled by eight parameters, including the image scale ims, the number of segments ns, and the smallest segment allowed ssa. Before segmentation begins, the image is scaled by a factor of *ims*, and at the end of the process, the result is scaled back by a factor of 1/ims. Segments are grown from the top left-hand pixel in the image, adding new pixels if their colour difference with respect to the first pixel in the segment is less then a threshold ndt and starting a new segment otherwise. Segments containing fewer than *ssa* pixels are merged with a larger neighbouring segment, which is chosen as the neighbouring segment with the closest colour match. Following this, the smallest segments are continually merged with larger ones until only *ns* remain. Each remaining segment is smoothed and abstracted. To smooth the boundary of a segment, the points within a circle of radius 8 pixels centred on each boundary pixel are added to the segment, which increases its size. To abstract the edges of segment, a path around the boundary of the segment, and paths around each hole in the segment, are determined using a back-tracking search. The upper and lower abstraction distances (lad and uad) parameters then come into play. Segments are ordered in terms of the number of elements they contain and we use L to represent the area of the largest segment and M to represent the area of the smallest. For a given segment, t, the abstraction distance for t is calculated as: $a(t) = lad + (\frac{r-M}{L-M})(uad - lad)$. Traversing the boundary of t, every $a(t)$-th pixel is added to a set of defining points for that boundary. A path is then plotted which passes through each *defining point* for the boundary and hole curves of a segment. The path can either be formed of straight lines or a Bezier curve, as per the es parameter. Note that the use of an abstraction distance proportionate to the size of a segment enables the larger segments to be more heavily abstracted than smaller segments, which helps preserve detail while still allowing abstraction to occur. A final parameter control whether holes are allowed in each segment (ha). In Fig. 1, we present four example segmentations. We see the effect of using abstraction levels proportionate to the segment size in segmentation C, where the flower detail is left fairly intact, but the background is abstracted. Note that combinations of the eight parameters ($ims, ns, ssa, ndt, lad, uad, es$ and ha) are used by the expert system to define various segmenting styles, as described in section 2.3 below.

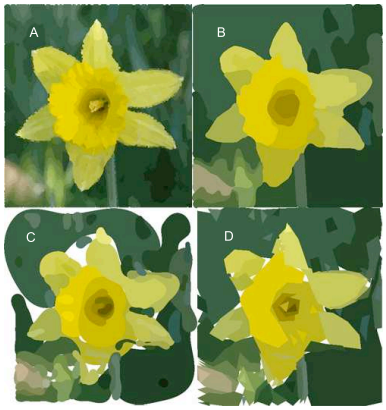

Figure 1: Example segmentations of a flower image.

2.2 Rendering Segmentations

The output from the segmentation process is an ordered list, S, of segments. Each segment is described by (a) its boundary path and paths around each hole (b) the defining points for the boundary path and paths around each hole (c) the original segment colour, and (d) a label describing the scene-element from which the segment came. Looking at the operation of the rendering process from the top down, at the highest level of description, The Painting Fool renders multiple *painting layers*. Each painting layer is produced by rendering every segment in S in turn. Each segment may be rendered multiple times in layers, and each layer may concentrate on outlining the segment or filling the segment. In either case, the rendering is achieved through multiple curves being drawn. Each curve is composed of a number of strokes, which simulate the usage of natural media such as pencils or acrylic paints. Below, we describe these processes from the bottom up (from strokes to painting layers), highlighting the parameters which control each process.

Rendering strokes. Strokes form part of a curve. The curve determines the path for the stroke, the orientation of the brush, and the base colour C for the stroke. The stroke is simulated using a straight line of n pixels $(p_1, \ldots, p_n)$ in the given orientation and centred on the first point of the path. Each p_i effectively simulates a bristle in a brush. The line is swept along the curve, maintaining its orientation and keeping the path point at its centre. There are 17 parameters that describe the effect of the stroke, such as the brush size, tapering and transparency proportions, friction probability and colour and shade variability. Without going into detail, these enable The Painting Fool to simulate acrylic/oil paint strokes in addition to pencils, chalks, charcoals and pastels. In Fig. 2, we present 12 strokes simulating pencils/charcoals and paints.

Rendering curves. The rendering of the outline of a segment or the filling of its body is achieved by rendering a series of curves. The path of each curve is determined by the particular segment rendering process being employed, and rendering a curve is subject to 5 parameters, including the stroke length min $slmin$ and max $slmax$, the stroke length variability slv and the stroke backtrack proportion sbp. To render the curve, a baseline stroke length L is chosen which is between $slmin$ and $slmax$ and is proportionate to the size of the segment being rendered. The curve is then split into strokes sequentially, taking slv and sbp into account. That is, the first stroke length F is randomly chosen within the range $L \pm slv * L$, then the next stroke starts at curve point $F * (1 - sbp)$ and its length is chosen similarly, etc. If the eas parameter is set to *end-points*, each stroke will be replaced by a straight line from the start to the end points of the stroke. If this parameter is set to *gradient* instead, each stroke will be replaced by a straight line in the direction of the gradient to the mid-point of the stroke. Once the curve has been split into strokes, the orientation of the brush is calculated to be in the direction of the line which is perpendicular to the straight line between the start and end points of the stroke. Each stroke is then rendered as above.

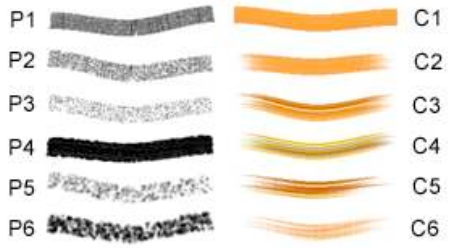

Figure 2: Example pencil/charcoal/paint rendering strokes.

Rendering segment layers. As previously mentioned, each segment may be rendered a number of times, with each layer rendered on top of the previous one. We have implemented one method for rendering the outline of a segment and five for rendering the body. Each method has its own control parameters and effectively defines a set of curves which are rendered as above. The *outline renderer* simply takes the boundary of the segment and the boundary of each hole and turns them into appropriate curves. The segment renderers which fill the body of the segment are controlled by the following parameters: the parallel line overlap plo, the wobble distance wd and radius wr, the coverage cov, the boundary decrease bd and decrease steps bds, and the boundary movement bm. The *parallel line fill renderer* splits the interior of the segment into a series of parallel lines of width equal to the brush size, which overlap with each other by plo pixels. Each parallel line is specified by a set of points along the line which are of distance wd apart. Each of these points is moved randomly to within a radius of wr from their original position. Then, a Bezier curve is drawn through the points, to produce a wobbly line. With the *random line fill renderer*, the segment is filled by randomly choosing pairs of points within the segment and drawing a straight line between them until cov of the area of the segment has been covered. Each line is subject to a wobble factor, as dictated by wd and wr. With the *decreasing circle fill renderer*, the segment is filled by drawing its outline, then reducing its boundary by moving each defining point bd places along the line which is perpendicular to the gradient of the segment boundary at that point. This is repeated bds times, or the parameter can specify that it runs to completion, i.e., fills the segment. The *star line fill renderer* starts at a random point on the segment boundary, then draws a line (subject to wobble factor) to the centre of the segment. It then moves bm points around the segment boundary and repeats this, until a full loop has been achieved. Some examples of segment fillers are given in Fig. 3.

Rendering painting Layers. The Painting Fool renders painting layers (the entire set of segments) multiple times in succession. Before it begins each painting layer, it first alters the segmentation, subject to these parameters: the number of colours noc, palette constraints pc, colour assignment method cam, the segment ordering, and segment shape transformations. Each segment is supplied with the original colour arising from the image. This can be used

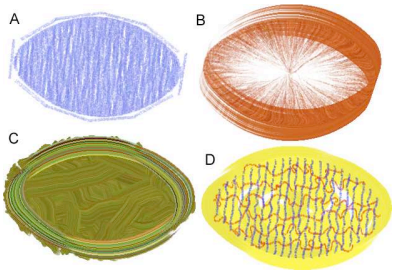

Figure 3: Examples of different segment filling styles.

to determine the rendering colour of the segment via a map onto a user-defined colour palette in a number of ways. Firstly, the noc parameter describes the maximum number of different colours which will appear in the rendering, and the pc parameter indicates which colours in the palette are allowed (e.g., only the top 10% in terms of saturation). Given a particular segment S, as dictated by cam, one option is for the palette colour closest to that of S to be assigned, and then optionally the brightness of this colour adjusted to the brightness of the original colour. Alternatively, colours can be assigned randomly from the palette, or could be equally distributed (so that all colours in the palette appear in the rendering). After segments have been assigned a colour, they are re-ordered within the segmentation. This is typically by segment size, so that the smaller segments are rendered on top of the larger segments. Finally, the user can specify a set of shape transformations (e.g., rotate, scale, stretch) for the segments to undergo. In particular, the entire segmentation can be scaled, rotated or stretched.

2.3 An Expert System for Artistic Styles

As we have seen, there are a large number parameters which can be altered to change the way The Painting Fool operates. We have stored sets of parameters which describe the segmentation and rendering methods in cross-referencing XML files. These determine the artistic style The Painting Fool will employ in a session, i.e., its level of abstraction, the colour palette and natural media it simulates, and its painting style. We have organised collections of parameter settings for the various processes undertaken by The Painting Fool. For instance, there are around 50 different collections of settings for pencil drawings, and a similar number for paints. We have fewer collections for pastels, chalks, charcoals and pens, but in total there are around 150 different rendering setups available. In addition to the parameter settings themselves, The Painting Fool allows the specification of *style files* for segmenting, rendering, shape transforms and colour mapping. These dictate conditions under which certain parameter settings are to be chosen. For instance, the user might specify in a segmentation style file that the scene-elements representing the eyes of a person in an image should each have 100 segments, whereas the nose should have only 20 segments, etc. Taken together, these style files constitute an artistic simulation knowledge-base.

To use this knowledge base as an expert system, we have also enabled the mapping of keywords onto particular collections of style files. For instance, the keyword 'happy' maps onto a vivid colouring style and a slapdash acrylic painting style. With specific emphasis on building an expert system for portrait painting, to populate the knowledge base, we undertook a project to build an online gallery of images called *Amelie's Progress*. To do this, we took 22 still images from the film "Amelie", each depicting a close-up shot of the actress Audrey Tatou expressing an emotion. Each photograph was hand annotated with the regions containing the eyes, the eyebrows, the nose, the mouth, the hair, the entire face, any clothing, and any areas of background. We then experimented with around 50 distinct artistic styles, applying each to the 22 original images. In each case, if we felt that the style was able to occasionally enhance the emotion expressed in the picture, we tagged the style with one of six keywords (anger, disgust, fear, happiness, sadness or surprise). From the pictures produced using the artistic styles, we hand-curated the gallery by choosing the 222 portraits we felt were the most aesthetically pleasing and/or emotive. After the production of this gallery (which can be viewed online at `www.thepaintingfool.com`), we achieved full automation in the process. As described in the next sections, to do this, we employed a machine vision system to supply all the required information, namely a suitable image of the sitter, the emotion they were expressing and the positions of their facial features.

3. FACIAL EXPRESSION ANALYSIS

The goal of our emotion detection software is to recognise the emotion displayed by the sitter. This can be one of six basic emotions: anger, disgust, fear, happiness, sadness or surprise. Ekman et al. postulated that these six basic emotions are universally performed and recognised [11]. Following the works by Ekman et al. [6] we will use the Facial Action Coding System (FACS) to analyse what facial muscle actions are being made and use this information to decide which emotion is shown. FACS is the best known and the most commonly used system developed for human observers to describe facial activity in terms of visually observable facial muscle actions. Using FACS, human observers decompose a facial expression into one or more of 27 defined AUs that produced the expression in question. Ekman et al. have shown that it is straightforward to recognise which expression of emotion was displayed if we know what AUs were activated during that expression. In the system presented here, we will use geometric features computed from tracked facial point data to detect 22 AUs that have a high relevance for inter-human communication. (That is, this set of AUs is sufficient to detect the six basic emotions with high reliability [7]). We will then use neural networks to map detected AUs to an emotion.

Recently, we have proposed a system that is capable of analysing both the morphology of an expression (i.e. determine which AUs were present) as well as the temporal dynamics of an expression [23]. For each AU that is found to be active in a video, we find exactly when the facial action starts, when it reaches its peak, when it starts to return to neutral and when it has returned to its neutral phase. We use this knowledge to find the frame in a video in which the emotion is displayed the most intensely. This frame is then sent to The Painting Fool, together with information on which emotion was recognised and the location of the facial points in that apex frame. The Painting Fool then uses this information to paint the portrait in an emotionally enhanced way. In the following subsections, we will describe how this facial expression analysis method works.

3.1 Automatic feature extraction

The features we use to detect AUs and their temporal segments are extracted by a fully automatic method consisting of, consecutively, face detection, facial point detection, facial point tracking and the calculation of geometric features. These features are used to train and test the classifier combination described in section 3.2. We will now describe each feature extraction subsystem in some detail.

To detect the face in a scene we make use of a real-time face de-

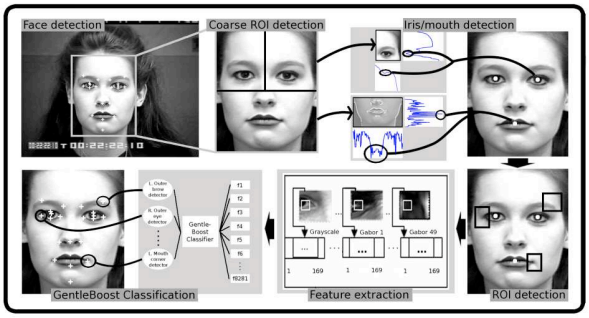

Figure 4: Outline of the facial point detection system.

tection scheme proposed in [8], which represents an adapted version of the original Viola-Jones face detector. The Viola-Jones face detector consists of a cascade of classifiers trained by AdaBoost. Each classifier employs integral image filters, which allow Haar Basis functions to be computed very fast at any location and scale. This is essential to the speed of the detector. For each stage in the cascade, a subset of features is chosen using a feature selection procedure based on AdaBoost. The adapted version of the Viola-Jones face detector that we employ uses GentleBoost instead of AdaBoost, which has been shown to be more accurate and converges faster [9]. Also, at each feature selection step (i.e., for every feature selected by AdaBoost), the proposed algorithm refines the feature originally proposed by GentleBoost. The algorithm creates a new set of filters by placing the original filter and slightly modified versions of the filter at a two pixel distance in both the x and y-direction.

The method that we use detects 20 facial feature points in a face image. To do so, it uses Gabor-feature-based boosted classifiers as proposed in [24]. The method, outlined in Fig. 4, assumes that the input image is a face region, such as the output of the face detection algorithm explained above. In this face region, the irises and the medial point of the mouth are detected first. A combination of heuristic techniques based on the analysis of the vertical and horizontal histograms of the upper and the lower half of the face-region image achieves this. Based on these three points and anthropomorphic relations, the input face region is divided into 20 regions of interest (ROIs), each corresponding to a facial point to be detected. For each pixel in the ROI, a feature vector is computed that consists of the grey values of the 13x13 patch surrounding the pixel and the responses to 48 Gabor filters (8 orientations and 6 spatial frequencies, 2:12 pixels/cycle at 1/2 octave steps). This feature vector is used to learn the pertinent point's patch template and, in the testing stage, to predict whether the current point represents a certain facial point or not. Next, to capture all facial motion, we track the detected points through all frames of the input video. The algorithm we used to track these facial points is Particle Filtering with Factorised Likelihoods (PFFL) [15]. We used the observation model proposed in [16], which is both robust against variations in lighting and able to cope with small deformations in the template. This polymorphic aspect is necessary, as many areas around facial points change their appearance when a facial action occurs (e.g. the mouth corner in a smile). The facial point tracking scheme results for every image sequence with n frames in a set of points P with dimensions $20 * 2 * n$. The facial points are registered first within each image sequence to remove rigid head motion. Next all facial point sequences are registered with respect to a pre-defined 'normal' face, to remove variations in head shape between subjects.

Now that we have a set of registered tracked facial points, we can compute our facial action features. For all points $\boldsymbol{p_i} \in P$, the first two features are simply its x and y position. We compute the features $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$ for every frame n:

$$f_1(\boldsymbol{p_i}, t) = p_{i,x,t} \tag{1}$$
$$f_2(\boldsymbol{p_i}, t) = p_{i,y,t} \tag{2}$$

For all pairs of points $\boldsymbol{p_i}, \boldsymbol{p_j}, i \neq j$, in each frame we compute:

$$f_3(\boldsymbol{p_i}, \boldsymbol{p_j}, t) = \|p_{i,t} - p_{j,t}\| \tag{3}$$
$$f_4(\boldsymbol{p_i}, \boldsymbol{p_j}, t) = \arctan\left(\frac{p_{i,y,t} - p_{j,y,t}}{p_{i,x,t} - p_{j,x,t}}\right) \tag{4}$$

Feature $\boldsymbol{f_3}$ describes the distances between two points $\boldsymbol{p_i}$ and $\boldsymbol{p_j}$, and feature $\boldsymbol{f_4}$ describes the angle that the line connecting $\boldsymbol{p_i}$ with $\boldsymbol{p_j}$ makes with the horizontal axis. The features $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$ are computed for all points $\boldsymbol{p_i} \epsilon P$, while the features $\boldsymbol{f_3}$, and $\boldsymbol{f_4}$ are computed for all possible combinations of points. The features $< f_1 \dots f_4 >$ contain only information about the positions of the facial points at a given instance in time. No information about their relation to a neutral frame, or about the rate of change of the values of these features is known. To capture this temporal information, we create a new set of features based on the single frame based features described above. First, we compute features that describe how much the feature values have changed, relative to their value at the neutral frame: $\delta(\boldsymbol{x}, t) = x_t - x_1$ where $\boldsymbol{x}$ is a time sequence and x_t its value at time t.

To determine the rate of change of the feature values at a given time instance t, we compute their first derivative with respect to time. Because we are working with discrete data, this becomes: $\frac{d(\boldsymbol{x},t)}{dt} = v(x_t - x_{t-1})$ where v is the frame rate of the corresponding recording and we use this definition to compute the features:

$$F =< f_1(t) \dots f_4(t), \delta(f_1, t), \\ \dots \delta(f_4, t), d(f_1, t)/dt \dots d(f_4, t)/dt > \tag{5}$$

This brings the total feature dimensionality to 1260.

3.2 Action Unit analysis

To detect whether an AU is active in a given video, we first detect AU activation frame-by-frame. This is done using a gentleSVM classifier. In this classifier setup, we first use GentleBoost to select the subset of most relevant features. This subset is then passed on to a SVM which learns the non-linear function that indicates when an AU is present for a given video frame. We train a separate gentleSVM classifier for each of the 22 AUs we wish to detect.

If any of the frames in a video is found to contain an AU, the video is passed on to the temporal segment recognition subsystem which determines exactly when a facial action starts, when it reaches its peak and when it returns to its neutral state. It does so by classifying for each frame whether it is in the neutral phase, the onset phase, the apex phase or the offset phase. Hence, we consider this

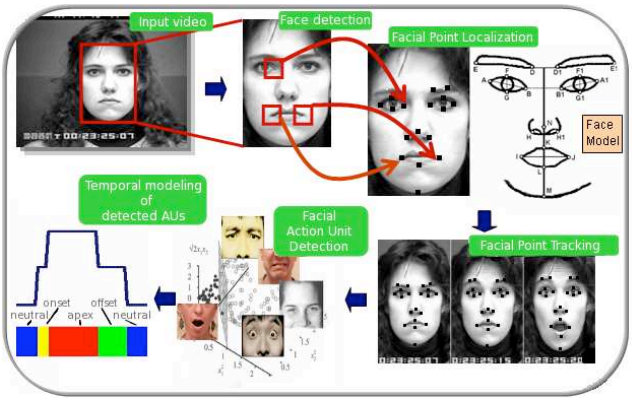

Figure 5: Outline of the Action Unit detection and temporal dynamics analysis system.

problem to be a 4-class classification problem. This temporal segment recognition process also removes any spurious AU activation detection results, e.g. when a single frame accidentally was classified to contain an AU that was not really there. If, after recognition of the temporal segments, any frame of a video is non-neutral for a specific AU (i.e., the AU is in its onset, apex or offset phase), we say that the video contains that AU. An overview of this system is presented in Fig. 5.

While the temporal dynamics of a facial action can be represented very efficiently by Hidden Markov Models (HMMs), the multiclass classification of the features on a frame-by-frame basis is normally done using Gaussian mixture models as the emission probabilities. These are know not to distinguish between multiple classes very well.

SVMs on the other hand discriminate extremely well. Using them as emission probabilities might very well result in an improved recognition. We therefore train a set of SVMs, one for every combination of classes (i.e., temporal phases neutral, onset, apex, and offset) and use their output to compute emission probabilities. This way we effectively have a hybrid SVM-HMM system.

The output of a SVM is unsuitable to use directly as a probability measure. The output $h(\boldsymbol{x})$ of a SVM is a distance measure between a test pattern and the separating hyper plane defined by the support vectors. There is no clear relationship with the posterior class probability $p(y = +1|\boldsymbol{x})$ that the pattern $\boldsymbol{x}$ belongs to the class $y = +1$. However, Platt proposed an estimate for this probability by fitting the SVM output $f(\boldsymbol{x})$ with a sigmoid function [17]:

$$p(y = +1|\boldsymbol{x}) = g(h(\boldsymbol{x}), A, B) \equiv \frac{1}{1 + \exp(Ah(\boldsymbol{x}) + B)} \quad (6)$$

Since SVMs are *binary* classifiers we use a one-versus-one approach to come to a multiclass classifier. This approach is preferred over the one-versus-all approach as it aims to learn the solution to a more specific problem, namely, distinguishing between one class from one other class at a time. For this pairwise classification we need to train $K(K-1)/2$ SVMs, where in our case $K = 4$ is the number of temporal phases. Our HMM consists of four states, one for each temporal phase. For each SVM we get, using Platt's method, pairwise class probabilities $\mu_{ij} \equiv p(q_i|\text{or}q_j, \boldsymbol{x})$ of the class (HMM state) q_i given the feature vector $\boldsymbol{x}$ and that $\boldsymbol{x}$ belongs to either q_i or q_j. These pairwise probabilities are transformed into posterior probabilities $p(q_i|\boldsymbol{x})$ by

$$p(q_i|\boldsymbol{x}) = 1/\left[\sum_{j=1,j\neq i}^{K} \frac{1}{\mu_{ij}} - (K-2)\right] \quad (7)$$

Finally, the posteriors $p(q|\boldsymbol{x})$ have to be transformed into *emission probabilities* by using Bayes' rule

$$p(\boldsymbol{x}|q) \propto \frac{p(q|\boldsymbol{x})}{p(q)} \quad (8)$$

where the a-priori probability $p(q)$ of class q is estimated by the relative frequency of the class in the training data.

3.3 Emotion recognition

If the AUs that make up a facial expression are known, and we know that the expression shown is one of the six basic emotions, it is not hard to find the corresponding emotion. Even a simple rule-based system would do [6]. However, rule-based systems are not very well suited to handle uncertainty in their input, and, as we will see in the evaluation section below, our AU detection is not perfect. On the other hand, ANNs are known for their resilience to input noise. In previous work we have shown that in the presence of input noise ANNs perform better than rule-based systems in recognising which emotion was shown [22]. We therefore learn a Neural Network with 22 input neurons (one for each AU we can detect), 3 hidden layers of 22 neurons each and an output layer of 6 neurons, one for every emotion. All neurons use the log-sigmoid evaluation function. The training data consists of the (noisy) output of the 10-fold cross-validation results of the AU recognition sub-system.

3.4 Performance Evaluation

We have evaluated our proposed methods on 244 videos selected from the MMI-Facial Expression Database [14], containing videos of 22 different AUs. We have chosen this database, instead of, for example, the Cohn-Kanade DFAT-504 dataset [10], because the videos in the MMI-Facial Expression Database display the full neutral-expressive-neutral pattern. This is essential, as it is this temporal pattern of facial actions that we are interested in. The videos were chosen so that the dataset contains at least 15 videos of every AU we want to analyse. We show here the results of two evaluation studies. The first study shows how well AUs are recognised by our detector. In table 1 we give detailed results on the performance of the AU activation detection sub-system. Besides the classification rate, we provide the recall rate, which indicates how many positive examples are retrieved from a test set, and the precision, which indicates how confident we can be that an example classified as positive is indeed positive. Finally, the F1-measure combines the recall and precision to compute a single measure that favours recall and precision equally. The temporal segments are recognised with the following accuracy, averaged over all AUs: the neutral phase with an F1-measure of 84.2%, the onset phase with 59.0%, the apex with 70.3% and the offset phase with 52.7%. A detailed evaluation of the AU temporal phase recognition can be found in [23].

Table 2 shows our results for detecting emotions using our ANN system described in section 3.3. As we can see from the table, the emotions disgust, fear, happiness and surprise are detected with

AU	Cl. Rate	F1-measure	AU	Cl. Rate	F1-measure
1	0.971	0.851	16	0.918	0.583
2	0.947	0.745	18	0.934	0.529
4	0.909	0.703	20	0.967	0.750
5	0.959	0.722	22	0.955	0.718
6	0.938	0.737	24	0.955	0.621
7	0.959	0.615	25	0.942	0.935
9	0.951	0.647	26	0.831	0.468
10	0.938	0.615	27	0.967	0.778
12	0.922	0.596	30	0.951	0.538
13	0.963	0.743	43	0.963	0.743
15	0.959	0.667	45	0.938	0.928
Avg:	**0.943**	**0.692**			

Table 1: Subject independent cross validation results for AU activation event detection after identification of the temporal segments of AUs. Results are for 244 examples taken from the MMI Facial Expression Database.

Emotion	Cl. Rate	Recall	Precision	F1-measure
Anger	0.915	0.500	0.539	0.519
Disgust	0.935	0.760	0.826	0.792
Fear	0.895	0.420	0.667	0.693
Happiness	0.948	0.917	0.868	0.892
Sadness	0.889	0.571	0.600	0.585
Surprise	0.974	0.938	0.938	0.938
Average	**0.926**	**0.734**	**0.740**	**0.737**

Table 2: Emotion recognition results. Emotions are derived from automatically detected AUs using Neural Networks.

very high accuracy. The emotions anger and sadness are more difficult to recognise with this system. For sadness this is probably because AU15, lip corners drawn down, is most important. However, as table 1 shows, AU15 is a difficult AU to recognise.

4. EMOTIONALLY AWARE PORTRAITS

To recap, we have built a non-photorealistic rendering system, and enabled it to react to emotion keywords by choosing an appropriate artistic style, and we have fully automated the process by employing an emotion detection system to provide input to the NPR system. The context within which we tested this integrated system was the British Computer Society Machine Intelligence competition, where AI software was demonstrated live. To produce the most impressive demonstration, we opted for full automation, with the demonstrators having only to start and stop a video camera into which subjects were asked to express an emotion, for example by smiling. The Painting Fool produces its pictures live, i.e., it renders each stroke in real time, which can add value to the demonstration. The full rendering process can take minutes or hours, so we chose artistic styles for each of the emotions which would complete the picture in around three minutes. To determine a suitable artistic style for each emotion, we drew heavily from the expert system developed for the Amelie's Progress gallery, although some minor changes were required. Below are descriptions of the artistic styles we chose for each emotion, and in Fig. 6, we present some example portraits for each emotion.

For anger we specified a line rather than curved segmentation style; a colour mapping to shades of green, except for the eyes, where shades of red were used; and a pencil rendering style which quickly sketched the outlines of segments.
For disgust, we specified low saturation greyish colours (to indicate rottenness); a shape transform which distorted the face by stretching it; and a decreasing-circle acrylic painting style which further distorted the face and highlighted the facial features.
For fear, we specified a colour mapping to cool hues such as blues and greys; a segmentation style which enlarged and emphasised the eyes; and a pastel rendering style where a layer of white pastel was added at the end, to produce a ghostly effect.
For happiness, we specified a mapping to a very vivid colour palette, with colours chosen randomly from the palette for each segment; and a slapdash painting style which outlined the segments, and then emphasised the facial features with simulated unmixed paint.
For sadness, we specified a mapping to a small palette of muted colours; and a rendering style which used pastels to paint a first layer, with the facial features emphasised on top with coloured pencil outlining.
For surprise, we specified a segmentation style which kept the features in tact, but abstracted the background; and a rendering style using pencils in star line fills to suggest explosions.

Figure 6: Example portraits. Emotions expressed, from top to bottom: anger, disgust, fear, happiness, sadness and surprise.

5. CONCLUSIONS AND FURTHER WORK

We have described a novel non-photorealistic rendering approach which externalises many of the parameters for controlling the process, so that an expert system of settings and style files could be developed to respond to high-level information such as the emotion being expressed by a person being painted. This response is in terms of an appropriate choice of artistic style which comprises segmentation, shape transformation, colour mapping and rendering styles. We have further described a machine vision system which analyses video clips and detects faces, tracks facial point and uses temporal information about their movement in order to determine to a fairly high accuracy the emotion being expressed. We have integrated these systems into a whole which is more than the sum of parts: the combined system takes live video of a person, detects the emotion, chooses an appropriate artistic style and then renders a portrait of that person in real-time – all with full automation.

There is still much room for improvement in this process. In particular, the images produced from the integrated system are of lower quality (aesthetically) than those in the Amelie's Progress gallery (see `www.thepaintingfool.com`). This is mostly because the manual annotation of the digital photographs used to create the gallery enabled more accurate rendering of the facial features. Hence, in future work, we hope to improve the accuracy of the feature detection as well as include detecting the outlines of facial features. We also hope to work with artists, in order to build a larger expert system of artistic styles. We do not claim that The Painting Fool's behaviour should be described as creative (yet). However, we do believe that a necessary condition [5] for creativity in a program is for it to have an internal appreciation of what it is doing. As it appreciates the emotion a sitter is expressing, and appreciates how its artistic styles may heighten emotion in portraits via the expert system, we can claim that the combined system exhibits some level of appreciation. Full appreciation of the artistic process would, naturally, include an appraisal of its paintings, which the system clearly does not do. Hence, it is a longer-term goal to enable The Painting Fool to analyse its own artworks.

6. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 211486 (SEMAINE).

7. REFERENCES

[1] K. Anderson and P. McOwan. A real-time automated system for recognition of human facial expressions. *IEEE Trans. Systems, Man and Cybernetics, Part B*, 36(1):96–105, 2006.

[2] M. Bartlett, G. Littlewort, C. Lainscsek, I. Fasel, and J. Movellan. Machine learning methods for fully automatic recognition of facial expressions and actions. *IEEE Int'l Conf. Systems, Man, and Cybernetics*, 1:592–597, 2004.

[3] S. Brooks. Mixed media painting and portraiture. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1041–1054, 2007.

[4] J. Collomosse and P. Hall. Salience-adaptive painterly rendering using genetic search. *Intl. Journal on Artificial Intelligence Tools (IJAIT)*, 15(4):551–576, 2006.

[5] S. Colton. Creativity versus the perception of creativity in computational systems. In *Proceedings of the AAAI Spring Symposium on Creative Systems*, 2008.

[6] P. Ekman, W. V. Friesen, and J. C. Hager. *Facial Action Coding System*. A Human Face, 2002. Salt Lake City.

[7] B. Fasel, F. Monay, and D. Gatica-Perez. Latent semantic analysis of facial action codes for automatic facial expression recognition. In *Proc. ACM SIGMM Int'l workshop on Multimedia information retrieval*, pages 181–188, 2004.

[8] I. Fasel, B. Fortenberry, and J. Movellan. A generative framework for real time object detection and classification. *Comp. Vis. and Image Understanding*, 98(1):181–210, 2005.

[9] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337–374, 2000.

[10] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pages 46–53, 2000.

[11] D. Keltner and P. Ekman. *Handbook of Emotions*. Guilford Press, 2004. New York.

[12] I. Kotsia, I.; Pitas. Facial expression recognition in image sequences using geometric deformation features and support vector machines. *IEEE Trans. Image Processing*, 16(1):172–187, Jan. 2007.

[13] M. Pantic and L. J. M. Rothkrantz. Toward an affect-sensitive multimodal human-computer interaction. *Proceedings of the IEEE*, 91(9):1370–1390, 2003.

[14] M. Pantic, M. F. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *Int'l IEEE Conf. Multimedia and Expo*, pages 317–321, 2005.

[15] I. Patras and M. Pantic. Particle filtering with factorized likelihoods for tracking facial features. *Proc. Int'l Conf. Automatic Face & Gesture Recognition*, pages 97–102, 2004.

[16] I. Patras and M. Pantic. Tracking deformable motion. *Proc. Int'l Conf. Systems, Man and Cybernetics*, 2005.

[17] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT press, 2000. Cambridge, MA.

[18] M. Shugrina, M. Betke, and J. Collomosse. Empathic painting: interactive stylization through observed emotional state. In *Proc' Int'l symp. Non-photorealistic animation and rendering*, pages 87–96, New York, NY, USA, 2006. ACM.

[19] T. Strothotte and S. Schlechtweg. *Non-Photorealistic Computer Graphics*. Morgan Kaufmann, 2002.

[20] Y. Tian, T. Kanade, and J. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2001.

[21] Y. L. Tian, T. Kanade, and J. F. Cohn. *Handbook of Face Recognition*. Springer, 2005. New York.

[22] M. F. Valstar and M. Pantic. Biologically vs. logic inspired encoding of facial actions and emotions in video. *IEEE Int'l. Conf. on Multimedia and Expo*, pages 325–328, 2006.

[23] M. F. Valstar and M. Pantic. Combined support vector machines and hidden markov models for modeling facial action temporal dynamics. In *IEEE W'shop Human Computer Interaction (HCI'07), IEEE ICCV'07*, 2007.

[24] D. Vukdadinovic and M. Pantic. Fully automatic facial feature point detection using gabor feature based boosted features. In *IEEE Int'l Conf. Systems, Man, and Cybernetics*, pages 1692–1698, 2005.

[25] Y. Zhang and Q. Ji. Active and dynamic information fusion for facial expression understanding from image sequence. *Trans. Pattern Recognition and Machine Intelligence*, 27(5):699–714, 2005.