SKIN-AWARE STYLIZATION OF VIDEO PORTRAITS

D. O'Regan[1] and A. C. Kokaram[2]

[1] Trinity College Dublin, Ireland, oregandm@tcd.ie [2] Trinity College Dublin, Ireland, anil.kokaram@tcd.ie

Abstract

This paper presents a new non-photorealistic/stroke-based rendering (NPR/SBR) framework for the stylization of videos featuring head shots of people, such as home videos, movies, and camera mobile phone clips. Spatiotemporal skin and edge detection are used to locate and emphasize the semantic content in the stylization process. The SBR portion of the algorithm features novel techniques for motion expression with elliptical brush strokes, brush stroke anchor point distribution, spatio-temporal color-sampling, and brush stroke animation with regard to state-of-the-art issues such as object occlusion and uncovering in the source video. A wide user-accessible parameter space and finishing touches such as cartoon-like edge decoration and other quirky effects empowers a variety of artistic outputs. The resulting stylized sequences are fun and interesting with regard to compression, summarization, motion visualization, story-boarding and art. Both the semantic content, and underlying video motion is highlighted and summarized on every frame of the stylized output sequence.

Keywords: Video Signal Processing, Probability, Motion Analysis, Rendering, Animation

1 Introduction

Non-photorealistic rendering (NPR), or the *stylization* of digital visual media is a wide research topic. A popular aspect is *cartoonization*, or cartoon-like rendering of images or video. Semi-automatic cartoonization tools have been used in the production of animated Hollywood films such as What Dreams May Come (1998, Polygram Filmed Entertainment) and A Scanner Darkly (2006, Warner Independent). Numerous image-based cartoonization plug-ins exist (e.g. ToonIt! plug-in for Adobe Photoshop, Digital Anarchy), and cartoonization applications are now present in some mobile camera phones.

Another interesting NPR concept is that of *stroke-based rendering* (SBR) [10], in which a *source image* or video is transformed to look as if it has been painted by an artist applying *brush strokes* to a *canvas*. This technique has been used to recreate painting styles such as impressionism [7, 18, 11, 8, 20], and drawing styles like sketching [4, 19] and stippling [22, 12]. Painterly rendering tools can be found in desktop publishing and professional, creative software such as Adobe Creative Suite 4.

Current NPR research seems to involve the idea of content-based applications. Hertzmann [9] presents an SBR algorithm in which user-defined semantic regions of the source image (e.g. people, faces) are painted in more detail. Santella and DeCarlo [21] and DeCarlo and Santella [5] present SBR and cartoonization algorithms respectively that utilize participant's eye-tracking data to ensure that the visually *salient* features of the source image are stylized clearly. Another content-based aspect of NPR is the visualization and summarization of the motion in image sequences using cartoon-like motion trails [3], or storyboard-like action key-frames [16].

Given the subtle relationships between these various techniques it is interesting to explore the possibility of merging these ideas into a single framework. This paper presents such a framework that combines several ideas in NPR: cartoonization, SBR, non-uniform stylization based on content, and motion depiction. The aim is not to emulate a particular real-life artistic style, but rather to experiment with merging these ideas. Automatic spatio-temporal skin and edge detection is incorporated to direct a skin-aware stylization, making this an ideal application for video portraits or head shots of people (e.g. home movies, cinema, mobile phone clips). In addition, this paper presents novel techniques for brush stroke distribution, dealing with occlusion and uncovering of objects in video SBR, and spatio-temporal color sampling. The resulting stylized sequences and source videos can be downloaded from `http://www.deirdreoregan.com/VS_CVMP09.html`.

2 The Specifics of SBR

A popular framework for SBR originated from the work of Haeberli [7], and has persisted to the current state-of-the-art. Hertzmann [10] provides a good overview of SBR and its issues. Typically, the process of painterly rendering an isolated video frame is commonly simulated in the following steps:

1. The canvas is initialized as blank, or with an *underpainting* [6] (i.e. a blurred version of the source image).
2. The source frame is pre-processed to obtain some *reference data* such as the gradient profile or a color segmentation.
3. A list of brush strokes needed to render the output painting is formulated with regard to the reference data.
4. The brush strokes are composited on the canvas in the *order* defined by the list.

5. Final touches such as simulated lighting, texture mapping, or hue adjustment finish the painting.

Brush strokes are modeled as entities with *attributes* including those relating to style (e.g. rectangular or spline-like), dimensions, orientation, color, opacity (i.e. for compositing), and *anchor point* coordinates. The latter relates to the position of the brush stroke's centroid in terms of the canvas dimensions, $\mathbf{X} = [x, y]$ where x and y are the horizontal and vertical coordinates respectively. A simple elliptical style brush stroke is utilized for the SBR portion of our stylization framework. The main attributes of an elliptical brush stroke can be visualized in Fig. 1 including; anchor point coordinates, $\mathbf{q} = [x_c, y_c]$, the length of the major and minor axes, a_e and b_e, and the angle between the major and the horizontal axes, θ. The brush stroke also has a color, $\mathbf{c} = [r, g, b]$, and opacity, α, both uniformly applied to it's masked area, $s(\mathbf{X})$.

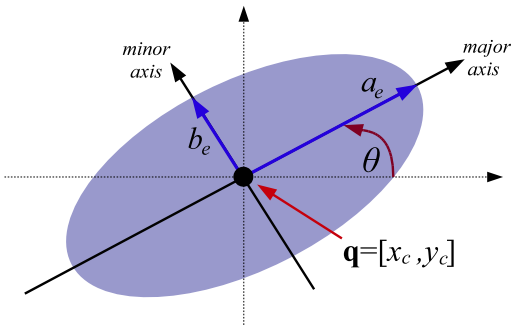


Figure 1: Elliptical brush stroke attributes and parameters.

3 Important Issues with SBR

The method of distributing brush stroke anchor points on the canvas is an interesting problem in SBR, that is reminiscent of the rendering technique of stippling [22, 12]. Haeberli [7] distributes the points in pseudo-random canvas locations until there are no *gaps* left in the painting, while Litwinowicz [18] anchors them on the nodes of a regular grid partitioning the canvas. The former has the problem of *redundancy*, where too many brush strokes may be placed close to or on top of one another, which is computationally inefficient. The latter produces a painting that can look too regular, and so noise must be added to the anchor point coordinates to emulate a hand-painted look.

Hertzmann [9] and Mignotte [19] present algorithms that distribute brush strokes on the canvas by varying their anchor coordinates and other attributes to minimize an energy comparing the output painting to the source frame. Often referred to as Relaxation [7, 9, 10], this technique is usually employed to drive finer painting detail (i.e. smaller, and/or more brush strokes) towards regions of texture and edges in the source image. Other algorithms such as that of Hays and Issa [8] achieve a similar effect by painting in multiple passes or *layers*, whereby an increasingly fine network of brush strokes are distributed in the region of edges detected at decreasing *scales* in the source image.

The method of animating brush strokes in video-based SBR is also important. Litwinowicz [18], Hays and Issa [8], Hertzmann [11], and Park and Yoon [20] utilize various applications of optical flow [13] or motion estimation [17] for video-based SBR. The technique presented in the first two of these examples involves continually motion compensating the brush strokes according to the forward flow or *motion field* obtained from the source video. Hence, a particular brush stroke's anchor point *trajectory* is continually determined by

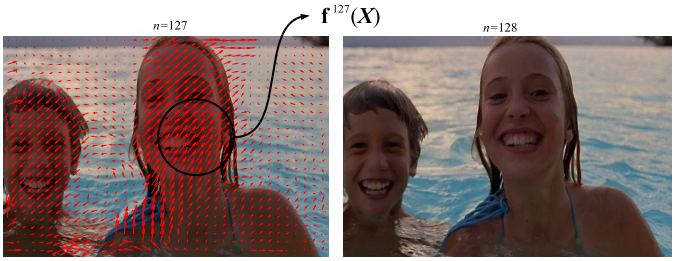
$$\mathbf{q}^{n+1} = \mathbf{q}^n + \mathbf{f}^n(\mathbf{q}^n) \tag{1}$$

where $\mathbf{f}^n(\mathbf{X})$ is the forward motion field estimated between frames n and $n+1$ of the video. The stylized effect is visually stunning in that the brush strokes appear to move as if they are stuck to the objects in the scene. This concept can be visualized in Fig. 2. A rectangular brush stroke - similar to the kind used in the work of Litwinowicz [18] and Hays and Issa [8] - is shown here.

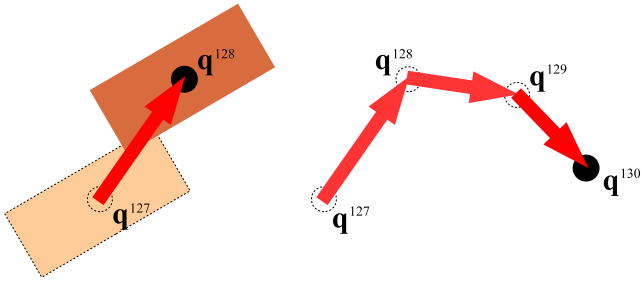
The motion models behind some motion estimators do not address the phenomena of *occlusion* and *uncovering* of objects in the video, and this causes specific problems in video-based SBR. Motion-compensated brush strokes tend to pile up in regions of occlusion, and redundancy results. Gaps in the painting are formed as the brush strokes migrate from regions of uncovering. The problem can be visualized in Fig. 3. Elliptical brush strokes are used to stylize frame $n = 75$ of the Hollywood sequence, and continually motion-compensated to stylize following frames.

Litwinowicz [18] deals with redundancy and gaps by a process involving Delaunay triangulation of the anchor point nodes in the grid. Hays and Issa [8] monitor the canvas for gaps and redundancy and vary the opacity of brush strokes temporally to transition them into and out of the painting as required. In contrast, this paper addresses this issue by detecting the underlying cause of these problems; occlusion and uncovering.

Temporal coherency is an issue in any video stylization algorithm. With SBR, the attributes of brush strokes should vary smoothly between frames. Of particular interest is the method of coloring the brush strokes, or *color-sampling*. It is well known that point-sampling the color from beneath the brush stroke's anchor point coordinates in each frame of the source video creates flicker in the stylized sequence [8, 1]. Hays and Issa [8] average the motion-compensated color values over a spatio-temporal window centered on the current frame and with spatial dimensions equivalent to those of the brush stroke. Similarly, the video watercolorization algorithm of Bousseau et al [1] smooths the motion-compensated color by taking an average over a cone-like spatio-temporal window tapering outwards from the current frame at the center. This paper uses a different spatio-temporal filtering strategy and avoids the heavy averaging process.



(a) Forward motion field



(b) Animating a brush stroke

Figure 2: Brush stroke animation; (a) the forward motion field for frame $n = 127$ of the Swimming sequence, and (b) motion compensating a brush stroke's anchor point (left) and its trajectory over a few frames (right).

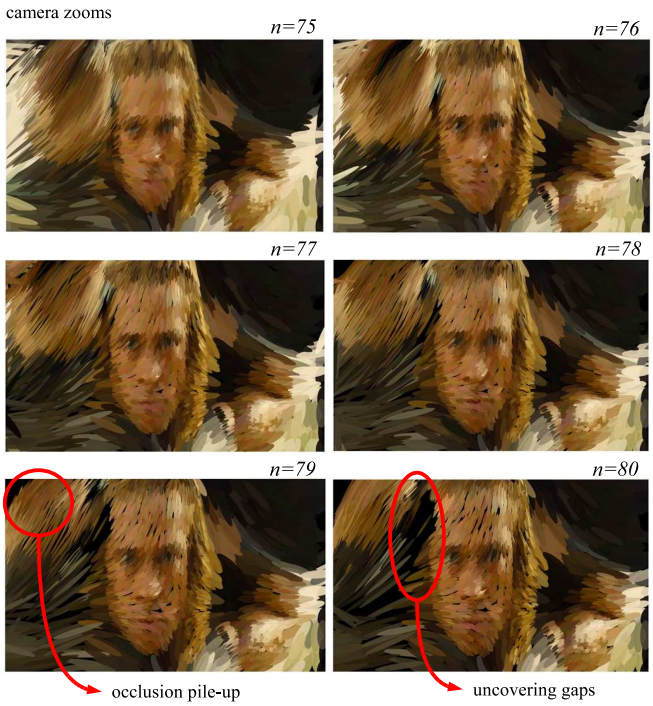


Figure 3: Redundancy and gaps; brush strokes pile up in regions of occlusion and drift apart in regions of uncovering. Here, a camera zoom triggers these problems.

4 Motion Expression

In previous SBR algorithms, the orientation of brush strokes is influenced by the direction of local gradients or edges detected in the source image [7, 18, 11, 9, 8, 20]. Although this is a good technique for preserving the high frequency detail in the video, the work presented in this paper experiments with the idea of varying the orientation and stretch of the elliptical strokes with the behavior of the underlying video motion.

With reference to Fig. 1, the area of an elliptical brush stroke is defined by $\phi_e = \pi a_e b_e$, while the relative lengths of a_e and b_e determine its eccentricity. To set $a_e = b_e$ means that the ellipse becomes a circle of radius, r_{eq}. By defining the stretch ratio $k_e = \frac{a_e}{b_e}$, the ellipse can be stretched by varying k_e while holding the area, ϕ_e constant. The stretch k_e of a particular elliptical stroke with anchor point q in frame n of the stylized output is then varied with motion using the following expressions

$$k_e{}^n = \frac{u^n}{\hat{u}}\hat{k}_e \qquad (2)$$

$$\text{where } u^n = \sum_{j=1}^{N_u} \sqrt{(\mathbf{u}_y^j)^2 + (\mathbf{u}_x^j)^2} \qquad (3)$$

$$\text{and } \mathbf{u}^n = \frac{1}{N_u} \sum_{j=0}^{(N_u-1)/2} \left[\mathbf{f}^{n+j}(\mathbf{q}^{n+j}) - \mathbf{b}^{n-j}(\mathbf{q}^{n+j})\right] \qquad (4)$$

where $\mathbf{f}^n(\mathbf{X})$ and $\mathbf{b}^n(\mathbf{X})$ are the forward and backward motion vector fields over a number of frames, N_u, and centered on the current frame, n. The terms $\mathbf{u}_y^j$ and $\mathbf{u}_x^j$ are the vertical and horizontal components of $\mathbf{u}^j$ respectively. The motion estimation algorithm of Kokaram [17] is used to obtain the motion fields. A temporal window with $N_u = 7$ is the default setting, and $\hat{u}$ is the assumed maximum value of u^n. The stretch of the ellipse is limited by the user-defined maximum, $\hat{k}_e$.

The orientation of the brush stroke in frame n of the stylized output, θ^n (radians), is defined as

$$\theta^n = \arctan \frac{\sum_{j=0}^{N_u} \mathbf{u}_x^n}{\sum_{j=0}^{N_u} \mathbf{u}_x^n} \qquad (5)$$

Using these formulae, elliptical brush strokes stretch and orient smoothly to portray the temporally varying magnitude and direction of the underlying video motion. This effect can be seen in Fig. 5.

5 Probabilistic Anchor Point Distribution

A novel probabilistic anchor point distribution process has been developed within the video stylization framework. This process is related to the idea of Poisson-disk sampling (PDS) [15], in which point samples are distributed on a plane (i.e. the canvas) in a series of *trials*. In each PDS trial, a candidate sample is generated pseudo-randomly, but rejected

if it falls within radial distance r_{disk} of previously generated sample. This is interesting in that r_{disk} could be chosen to compliment the dimensions of brush strokes. As previously explained, however, the shape of our motion-expressive elliptical brush strokes vary over the canvas. Our novel approach attempts to soften the process of PDS by posing it in a probabilistic light.

Anchor points are generated on the canvas in a series of trials, i. The point generated by each successful trial is $\mathbf{q}_i = [x_{c_i}, y_{c_i}]$. During the trials, each location on the canvas, $\mathbf{X}$ is associated with sampling probability. At the beginning of the process, therefore, we have an initial uniform distribution, $p_0(\mathbf{X}) = k_0$, as can be seen in Fig. 4.

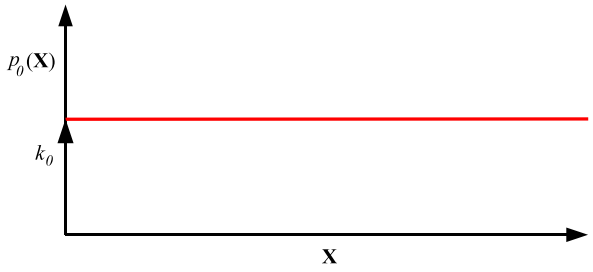


Figure 4: Initially a uniform distribution over canvas coordinates, X.

Here, k_0 is a constant. To generate the first anchor point, $\mathbf{q_0}$, a sample is drawn from this distribution numerically

$$(\tilde{x}_0, \tilde{y}_0) \sim p_0(\mathbf{x}, \mathbf{y}) \mapsto \mathbf{q}_0 = [x_{c_0}, y_{c_0}] \tag{6}$$

The aforementioned PDS algorithm would maintain a uniform distribution throughout the sampling process, explicitly rejecting those points whose coordinates fall within r_{disk} of an existing anchor point. Here however, the sampling probability is modified after the placement of a brush stroke so as to suppress the likelihood of anchoring another nearby. Therefore we have the continuous process

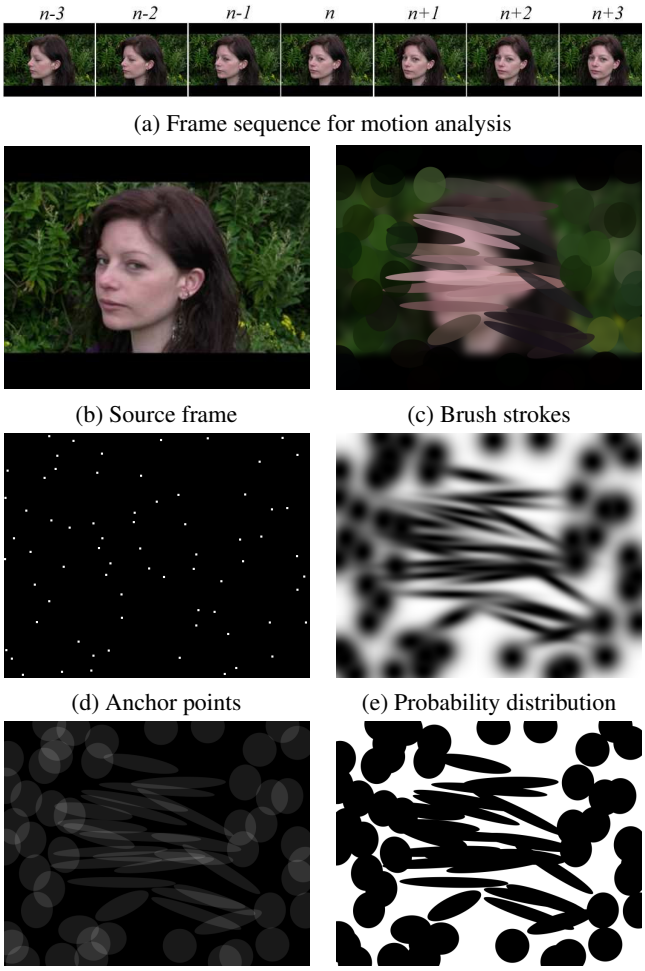
$$(\tilde{x}_i, \tilde{y}_i) \sim p_i(\mathbf{X}) \mapsto \mathbf{q}_i = [x_{c_i}, y_{c_i}] \tag{7}$$

$$p_i(\mathbf{X}) = p_{i-1}(\mathbf{X}) p_{b_{i-1}}(\mathbf{X}) \tag{8}$$

where $p_{i-1}(\mathbf{X})$ is the probability distribution from the previous trial, and $p_{b_{i-1}}(\mathbf{X})$ is the modification that resulted from it. The modification models a suppression field with peak at the previously generated brush stroke's anchor point, $\mathbf{q}_{i-1}$, and smoothly decreasing outwards from the center. Generally, the Gauss-like suppression function generated in trial i is defined as

$$p_{b_i}(\mathbf{X}) = \frac{1}{Z}\left[C - e^{-(v_s \mathbf{X}^T \mathbf{S} \mathbf{X})}\right] \tag{9}$$

where Z and C are a normalizing and scaling factor respectively, $\mathbf{S}$ is a 2D covariance matrix whose diagonal entries correspond to a_{e_i} and b_{e_i} of the elliptical stroke anchored at $\mathbf{q}_i$, and $v_s \in \Re$ is a weight. $C = 0$ is the special case preventing two anchor points from being placed in the same location on the canvas.



(a) Frame sequence for motion analysis

(b) Source frame (c) Brush strokes

(d) Anchor points (e) Probability distribution

(f) Cumulative count (g) Gap mask

Figure 5: The SBR process; (a) frame sequence, $N_u = 7$, used to influence the elliptical stretch and orientation, (b) the source frame, n, (c) brush strokes composited on an underpainting, (d) anchor point distribution, (e) probability distribution, $p_i(\mathbf{X})$, (f) cumulative count image, $h_i(\mathbf{X})$, and (g), gap mask, $g_i(\mathbf{X})$. Other parameters include $r_{eq} = 40$, $\hat{u} = 35$, $\hat{k}_e = 6$, $v_s = 3$, $C = 0$ and $\alpha = 0.75$

Fig. 2 (a-e) demonstrates the effect of a number of trials in the stylization process of frame $n = 33$ of the Female1 sequence, including an overhead view of the resulting probability distribution, $p_i(\mathbf{X})$. Dark areas represent low probability and bright areas represent high probability.

Canvas coverage is monitored by maintaining a cumulative count image, $h_i(\mathbf{X})$, with bin dimensions corresponding to those of the canvas. This can be visualized in Fig. 2 (f). After each trial, $h_i(\mathbf{X})$ is incremented over the masked area of the anchored stroke, $s_i(\mathbf{X})$, as follows

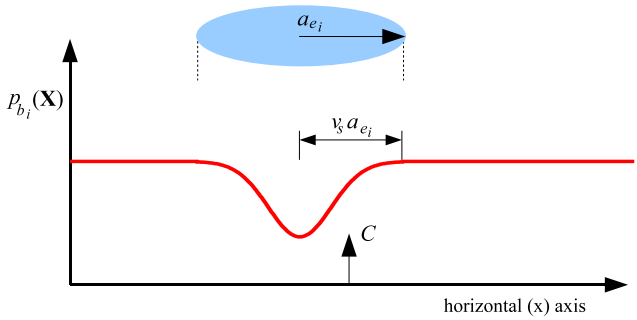


Figure 6: Gauss-like suppression function for coordinates $\mathbf{x}$ ***when the brush stroke's major axis,*** a_{e_i}***, is in line with the horizontal axis. Here*** $C > 0$ ***and*** $v_s = 1$***.***

$$h_i(\mathbf{X}) = h_{i-1}(\mathbf{X}) + s_i(\mathbf{X}) \tag{10}$$

The scheme used to terminate the process is a similar to that of Hays and Issa [8]. Essentially, the painting is completed in two passes. In the first pass, trials occur until $\bar{h}_i > t_a$, where t_a is a threshold on $\bar{h}_i$, the mean of $h_i(\mathbf{X})$. The second pass is a *hole-filling* process. Before the second pass a gap mask, $g_i(\mathbf{X})$, is defined and used to alter $p_i(\mathbf{X})$ as follows

$$g_i(\mathbf{X}) = \begin{cases} 1 & \text{where } h_i(\mathbf{X}) < 1 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

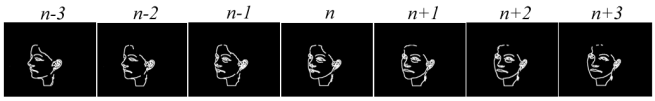
$$p_i(\mathbf{X}) \propto p_{i-1}(\mathbf{X}) g_i(\mathbf{X}) \tag{12}$$

A gap mask can be visualized in Fig. 2 (g), where gaps are represented by white areas. Gaps that form larger connected regions are perceived as *holes* in the painting, and A_j is defined as the area of a particular connected hole. Sampling trials continue in the second pass until $\hat{A}_j <= A_H$, where $\hat{A}_j$ is the largest hole, and A_H is the maximum tolerated area of a hole.

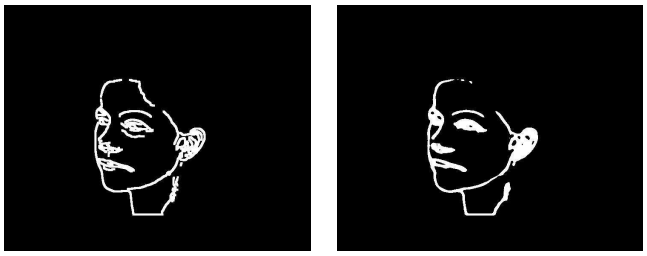
6 Semantic Layers

To achieve a skin-aware painting, three semantic *layers* are defined for each video frame, n; background, $L_b^n(\mathbf{X})$, foreground, $L_f^n(\mathbf{X})$, and detail, $L_d^n(\mathbf{X})$. Since the source videos consist of head shots, spatio-temporal skin detection is used to define the foreground layer, and edge detection is used to obtain the detail mask. Stylization parameters can then be varied between layers to emphasize the salient content (i.e. skin and edges within skin regions).

Skin detection is carried out using a statistical approach similar to that of Jones and Regh [14], and the resulting skin mask is spatially smoothed to give $L_f^n(\mathbf{X})$ using the max-flow/min-cut algorithm of Boykov and Kolmogorov [2]. To obtain $L_d^n(\mathbf{X})$, Canny edge detection is performed on the source frame, and the resulting edge mask is morphologically dilated by a small radius of $r_{d_1} =\in \{1..5\}$, echoing the approach of Hays and Issa [8]. Edges falling outside of $L_f^n(\mathbf{X})$ are discarded. The resulting layer masks are spatially but not temporally smooth, so they are filtered by means of motion-compensated median filtering over a small spatio-temporal window of width $W_L =\in \{5..11\}$ and temporal extent $N_L =\in \{5..11\}$. Fig. 7 illustrates the smoothing process with $L_d^n(\mathbf{X})$ associated with frame $n = 33$ of the Female1 sequence.



(a) Mask sequence for smoothing



(b) Canny edge detection (c) Spatiotemporal smoothing

Figure 7: Spatiotemporal smoothing; (a) mask sequence, $N_L = 7$***, used in smoothing, (b) Canny edge mask for*** n***, and (c) the spatio-temporally smoothed*** $L_d^n(\mathbf{X})$***. Here*** $W_L = 7$***.***

The user can choose to vary the size of brush strokes between these layers, as can be seen in Fig. 8. Here, brush strokes with three different sizes $r_{eq} = \{40, 15, 5\}$ are used to paint in the regions of background, skin and edge detail respectively. An interesting problem is observed in Fig. 8 (d,e) in that few small brush strokes are distributed with the aforementioned probabilistic anchor point distribution process, especially within the detail sites, $L_d^n(\mathbf{X})$. This is due to the wide suppression fields generated by the larger brush strokes deposited in the background region, coupled with the fact that the spatial sampling area of $L_d^n(\mathbf{X})$ is smaller.

A solution to this problem is to weight the initial probability distribution, $p_{i=0}^{n=0}(\mathbf{X})$ as follows

$$p_{i=0}^{n=0}(\mathbf{X}) = v_b L_b^n(\mathbf{X}) + v_f L_f^n(\mathbf{X}) + v_d L_d^n(\mathbf{X}) \tag{13}$$

where $v_b, v_f, v_d \in \Re$ are weights of increasing value. The effect can be seen in Fig. 8 (e,f), in the painting of a frame from the Swimming sequence. Here, the weights are set to a multiple of the relative area masked by each layer. Another solution is to paint each layer separately, each with a unique $p_i^n(\mathbf{X})$ and associated parameters defined only in the region of the layer mask. The resulting effect can be seen in Fig. 8 (g,h).

7 Animating the Brush Strokes

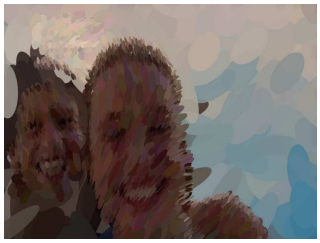
Brush strokes are animated by continually motion compensating their anchor points according to the underlying motion of the source video (refer to Section. 3 and Eqn. 1). The motion estimation algorithm of Kokaram [17] is used

(a) Frame sequence for motion analysis

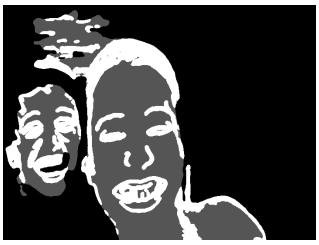
(b) Source frame

(d) Output painting

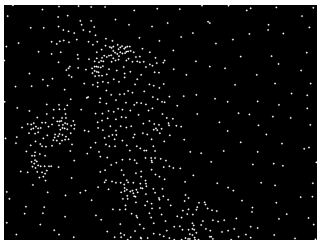
(f) Weighted painting



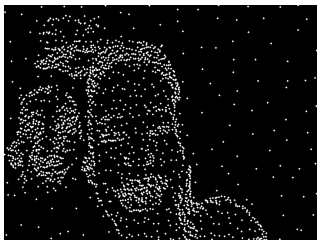
(h) Layer-based painting



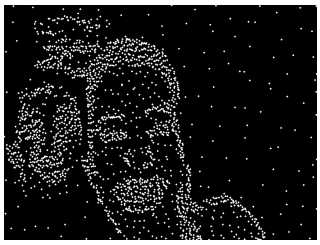
(c) Semantic layers



(e) Anchor points



(g) Weighted anchors



(i) Layer-based anchors

Figure 8: Semantic layers; (a) frame sequence, $N_u = 7$***, used to determine elliptical properties and layer masks, (b) the source frame*** n***, (c)*** $L_f^n(\mathbf{X})$ ***(gray) and*** $L_d^n(\mathbf{X})$ ***(white), (d) output painting, (e) brush stroke anchor points, (f) weighted probability painting, (g) weighted probability anchor points, (h) layer-based output painting, and (i) layer-based anchor points.***

to obtain the motion information. As previously explained, regions of object occlusion and uncovering in the source video trigger the usual problems in our motion-compensated brush stroke animation.

A novel solution to the problem of brush stroke pile-up is to detect the frame-wise regions of occlusion, and prevent brush strokes from being translated into these regions. A measure of both occlusion and uncovering is obtained by forming the image, $o^n(\mathbf{X})$ in the following steps

$$\mathbf{X}^{n'} = \mathbf{X}^{n-1} + \mathbf{f}^{n-1}(\mathbf{X}) \tag{14}$$

$$\mathbf{X}^{n''} = \mathbf{X}^{n'} + \mathbf{b}^{n}(\mathbf{X}) \tag{15}$$

$$o^n(\mathbf{X}) = |\mathbf{X}^{n'} - \mathbf{X}^{n''}| \tag{16}$$

where $\mathbf{X}^{n'}$ is the translation of the pixel coordinates in frame $n-1$ to n according to the motion estimator and $\mathbf{f}^{n-1}(\mathbf{X})$ the forward motion field estimated for frame $n-1$. $\mathbf{X}^{n''}$ is the translation of the estimated $\mathbf{X}^{n'}$ back to the locations in frame $n-1$ according to $\mathbf{b}^n(\mathbf{X})$, the backward motion field estimated for frame n.

In areas of no occlusion and uncovering it would be expected that $\mathbf{X}^{n'} = \mathbf{X}^{n''}$, and hence $o^n(\mathbf{X})$ would be low. Since the motion estimator fails in regions of occlusion and uncovering, it follows that $o^n(\mathbf{X})$ is higher in these regions. Significant regions can be detected by forming a mask of $o^n(\mathbf{X}) > t_o$, and t_o is a threshold. Anchor points that are translated into these regions via motion compensation are simply deleted, as are those translated beyond the spatial boundaries of the video, or from one semantic layer to another. Furthermore, strokes whose motion-compensated translation excites the condition $\hat{h} > t_m$ are deleted to curb redundancy. Strokes early in the list defining compositing order are deleted first, because they are less likely to be seen popping in and out since they exist at the rear of the painting.

To fill the regions of uncovering, gaps are detected using a gap mask, $g^n(\mathbf{X})$, as previously defined in Eqn. 12, and sampling activity is reduced to the canvas region defined by it. The previously discussed two-pass brush stroke distribution is then carried out on this reduced sampling space until painting termination. This stage also fills gaps that have been created by the deletion of brush strokes translated into regions of occlusion. All newly generated brush strokes are added to the head of the list defining compositing order, such that they are painted early in the composition.

8 Spatiotemporal Color Sampling

The SBR portion of our video stylization framework utilizes a novel spatio-temporal color-sampling algorithm for the coloring of brush strokes. Two filters are used in cascade; a spatial Hamming window is used to simultaneously sample and smooth spatial color, and a Butterworth filter removes variations of this sampled color temporally. By doing this we avoid a heavy spatio-temporal averaging process while still obtaining temporally smooth brush stroke coloring.

The spatial process can be visualized in Fig. 9, and it yields a color sample

$$\mathbf{c}_i^n = \sum_j [v_j \mathbf{C}_i^n(\mathbf{X} + d_j)] \tag{17}$$

where $\mathbf{c}_i^n$ is the color sampled for the brush stroke generated in trial i in the stylization of frame n, $\mathbf{C}_i^n(\mathbf{X})$ are the pixel-wise color values of the source frame, d_j indexes the samples in the spatial window, and v_j are weights corresponding to the Hamming window, both centered on $\mathbf{q}_i^n$. Hamming weights ensure that color values closer to the stroke's anchor point are given more importance in the spatial smoothing stage. The width of the spatial window, W_c, can be adjusted according to $W_c = 2k_c r_{eq}$, where $k_c \in \{0...1\}$ is a user-defined constant. The smaller the value of k_c, the more distinct the color of individual strokes will appear.

Temporal smoothing is achieved with an Infinite Impulse Response (IIR) Butterworth filter of order N_b, and normalized cutoff frequency, ω_b. The taps of a third order (i.e. $N_b = 3$) Butterworth filter with varying ω_b can be seen in the Table 10. Note that $\mathbf{c}_i^n$ refers to the spatially smoothed color sample for frame n, whereas $\mathbf{c}_{b_i}^{n-1}$ refers to the temporally smoothed color output of the Butterworth filter for frame $n-1$ and so on.

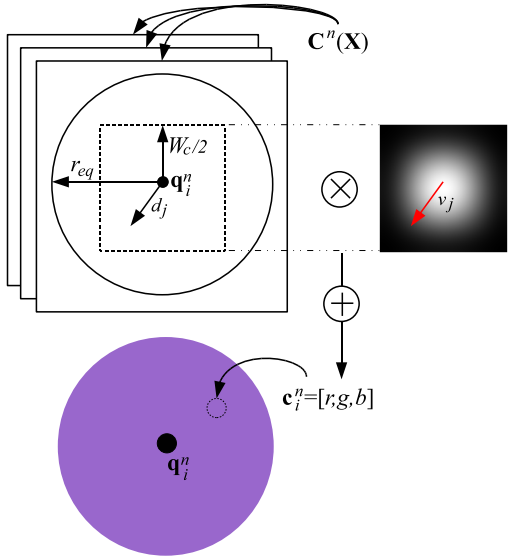


Figure 9: The process of spatial color-sampling to paint a circular brush stroke.

ω_b	$\mathbf{c}_{b_i}^{n-3}$	$\mathbf{c}_{b_i}^{n-2}$	$\mathbf{c}_{b_i}^{n-1}$	$\mathbf{c}_i^n$	$\mathbf{c}_i^{n-1}$	$\mathbf{c}_i^{n-2}$	$\mathbf{c}_i^{n-3}$
0.1	-0.5321	1.9294	-2.3741	0.0029	0.0087	0.0087	0.0029
0.3	-0.1378	0.6959	-1.1619	0.0495	0.1486	0.1486	0.0495
0.5	0	0.3333	0	0.1667	0.5	0.5	0.1667

Figure 10: Taps of an order $N_b = 3$ Butterworth filter with varying cutoff frequency, ω_b.

The results of our color-sampling process can be compared with that of simple frame-wise point-sampling at anchor point coordinates, and spatio-temporal motion-compensated averaging in the Hollywood_b1, Hollywood_point, and Hollywood_mean sequences respectively at `http://www.deirdreoregan.com/VS_CVMP09.html`. In the former $N_b = 3$, and a spatial color-sampling window defined by $k_c = 0.5$ is used with brush strokes of dimensions $r_{eq} = \{30, 15, 5\}$ to give $W_c = \{30, 15, 5\}$ on the three semantic layers. In the latter, a uniform symmetrical spatio-temporal volume of equivalent W_c and a temporal extent of 7 frames centered on the current frame is tested for comparison. These videos demonstrate that the former performs well compared with the latter, and that frame-wise point-sampling with no temporal smoothing simply causes flicker.

9 Finishing Touches

Noise can be added to the color and orientation parameters of the elliptical brush strokes, and these can be varied between semantic layers, as can the different stylization effects. A cartoon-like look can be achieved by alpha-compositing color onto the painted canvas using $L_d^n(\mathbf{X})$ as follows

$$z^n(\mathbf{X}) = E_c z^n(\mathbf{X}) \alpha_d (1 - L_d^{n'}(\mathbf{X})) \tag{18}$$

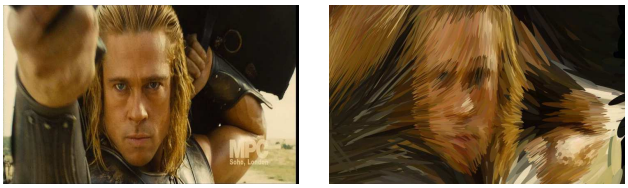
where $z^n(\mathbf{X})$ is the output painting post-SBR, α_d is the opacity of the edge decoration, and E_c is a constant defining its color. The mask $L_d^{n'}(\mathbf{X})$ is simply $L_d^n(\mathbf{X})$ dilated by a small radius, $r_{d_3} =\in \{1..3\}$, and smoothed with a normalized spatial Hamming window of width $W_d =\in \{3, 5\}$. Examples of edge decoration can be seen in Figs. 13 and 14.

The technique of stretching and orientating brush strokes according to motion presents a problem that can been seen in Fig. 11 (b), which shows the painting of (a) with brush strokes of various r_{eq}, and elliptical parameters $\hat{k}_e = 9$ and $\hat{u} = 32$. It is clearly expressive of the motion of surrounding frames, but much of the high frequency detail is lost in the stretch of the elliptical strokes. This problem is addressed by limiting the stretch of the elliptical strokes in the detail region defined by $L_d^n(\mathbf{X})$. Fig. 11 demonstrates this idea. Fig. 11 (b) shows the painting of (a) with brush strokes of various r_{eq}, and elliptical parameters $\hat{k}_e = 9$ and $\hat{u} = 32$. The idea is to to limit the elliptical stretch increasingly close to edges, reducing the ellipses to circles on edge sites. Fig. 11 (c) is a special detail (i.e. edge) distance filter, $d^n(\mathbf{X})$, designed for this purpose and defined by

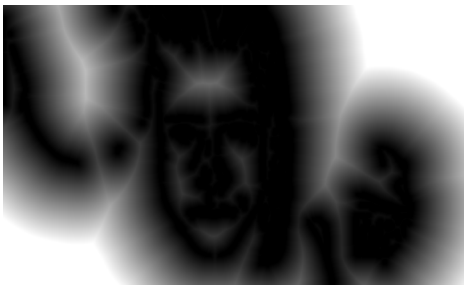
$$k_e^n(\mathbf{X}) = k_e^{n'}(\mathbf{X}) d^n(\mathbf{X}) \tag{19}$$

where $k_e^{n'}(\mathbf{X})$ is the original stretch profile of the brush strokes as calculated in Section 4. The detail distance filter, $d^n(\mathbf{X})$, is a pixel-wise measurement of the Euclidean distance from each site in $\mathbf{X}$ to the nearest non-zero value marked by a dilated version of $L_d^n(\mathbf{X})$. These values are normalized, scaled by a constant E_d, and clipped such that $d^n(\mathbf{X}) > 1 = 1$. The value of $E_d \geq 1$ determines the stretch-limiting influence of $d^n(\mathbf{X})$ across $\mathbf{X}$, and $k_e^n(\mathbf{X})$ at sites where $L_d^n(\mathbf{X}) = 1$ is always unity. Fig. 11 (c) shows $d^n(\mathbf{X})$ with $E_d = 2$ on a frame in the Hollywood sequence, (d) is the stylized result of involving

Eqn. 9. The output painting is still motion-expressive, but more high frequency detail is preserved.



(a) Source frame (b) Output painting



(c) Detail distance filter

(d) Stretch control

Figure 11: Preserving detail; (a) the source frame n***, (b) output painting with*** $\hat{k}_e = 9$***, and*** $\hat{u} = 32$***, (d) detail distance filter*** $d^n(\mathbf{X})$ ***with*** $E_d = 2$***, and (e) output painting with*** $d^n(\mathbf{X})$ ***influencing*** $k_e^n(\mathbf{X})$***.***

Another quirky style can be created by sampling the source frame beneath the color-sampling window defined by k_c in Fig. 9, and resizing the image to the dimensions of the brush stroke's masked area, $s_i^n(\mathbf{X})$, using bicubic interpolation. The result is a lens-like effect, and the brush strokes still move with the underlying motion of the sequence. Fig. 14 presents example frames incorporating this effect (see the bottom two frames).

10 Results

A number of source videos have been stylized, and they can be obtained from `http://www.deirdreoregan.com/VS_CVMP09.html`, along with a full list of parameters. Some edge-decorated results can be seen in Figs. 13 and 14. More stylized video frames can also be seen in Fig. 14, and the lens-effect can be visualized in in the last two example frames. Adobe AfterEffects was used to track the eyes to form the detail region in the Female1_3 sequence (bottom left), and the lens-like effect is confined to the background region in Swimming_2 (bottom right). The seventh and eighth example frames of Fig. 14 are notable in that they capture the behavior of the motion-expressive brush strokes and spatio-temporal color-sampling process on frames near to *shot cuts* in the source video. This behavior can be observed in any of the stylized results of the Hollywood sequence. Note that the filters for smoothing semantic layer masks are temporally clipped at the shot cut boundaries.

The resulting stylized videos demonstrate a variety of artistic options possible within the framework. Future work is dependent on the stylistic goals of a user. A variety of brush stroke shapes and styles could be implemented for SBR, with paint-like textures and lighting, as described by Hertzmann [10]. Rapid temporal changes in the orientation of elliptical brush strokes are observable, and θ could be smoothed in a manner similar to the temporal filtering of color described in Section 8. Only some of the videos demonstrate the result of limiting the stretch of elliptical strokes at edge sites, some with a cartoon-like edge decoration and some not. It is quite subjective which, if any, of this combination of effects is interesting or aesthetically pleasing. Rather than curbing the stretch of the detail brush strokes at edges to preserve high frequency detail as presented in this paper, they could be orientated parallel to the edges as in previous NPR work [18, 8]. The underlying skin and edge detection algorithms could also be improved. The cartoon-like edges are prone to flicker in periods of rapid motion (see stylized outputs of Swimming) or changes in illumination (see stylized Male1), for example. Background and/or hair can be falsely masked as skin (see stylized Male1 and Hollywood), but it is subjective as to whether this has a negative effect on the stylized video results.

11 Conclusion

A novel non-photorealistic/stroke-based rendering (NPR/SBR) framework for the stylization of videos of people has been presented. The resulting stylized sequences are visually interesting, and demonstrate a variety of artistic outputs. The stylization is content-based in that skin and detail regions are painted more finely and highlighted, whereas the background is abstracted. The use of motion-expressive elliptical brush strokes for SBR empowers motion visualization and summarization, such that a snapshot of the underlying source video motion is captured in every frame of the stylized sequence. Furthermore, novel ideas for brush stroke distribution, spatio-temporal color-sampling, and methods for dealing with occlusion and uncovering in brush stroke animation have been presented. This framework might be useful for the stylization of home movies, films or camera mobile phone clips.

Acknowledgments

This research was funded by the Irish Research Council for Science, and Engineering Technology (IRCSET). Grant number: RS/2005/105.

References

[1] A. Bousseau, F. Neyret, J. Thollot, and D. Salesin. Video watercolorization using bidirectional texture advection. *ACM Trans. on Graphics (Proc. of SIGGRAPH '07)*, 26(3), 2007.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Pattern Analysis and Machine Intelligence (PAMI)*, (26), September 2004.

[3] J. P. Collomosse, D. Rowntree, and P. M. Hall. Cartoon style rendering of motion from video. In *Proc of Intl. Conf. on Vision, Video, and Graphics '03*, pages 117–124, July 2003.

Figure 12: Example frames from the source videos; (l-r) Swimming, Hollywood, Male1, Hollywood, Female1, Hollywood

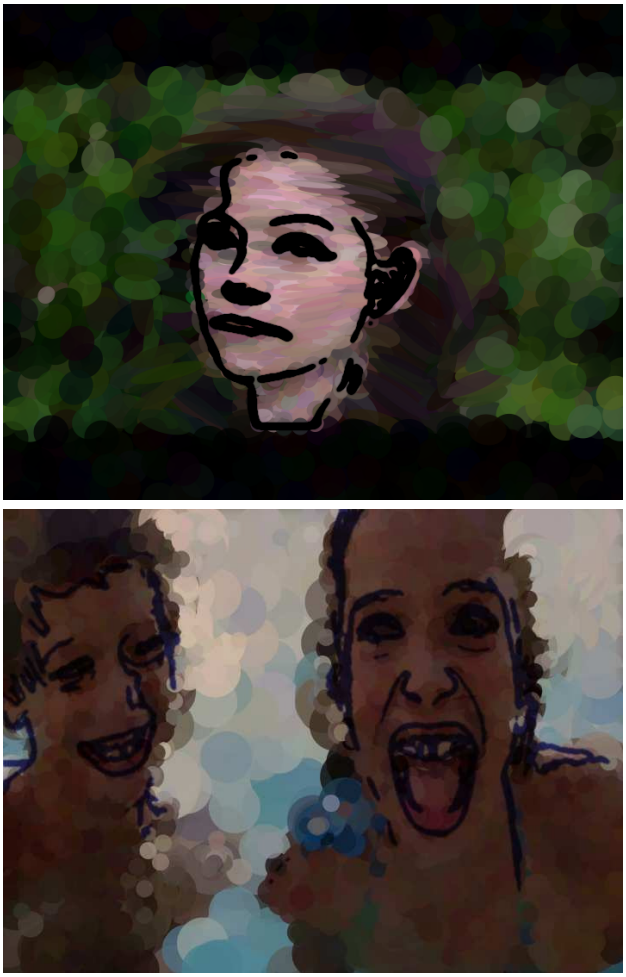


Figure 13: Example frames from stylized videos; (t-b) Female1_1, Swimming_1

[4] C. J. Curtis. Loose and sketchy animation. In *Proc. of ACM Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH '98), Electronic Art and Animation Catalogue*, page 145, 1998.

[5] D. DeCarlo and A. Santella. Stylization and abstraction of photographs. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH '02)*, 21(3):769–776, 2002.

[6] B. Gooch, G. Coombe, and P. Shirley. Artistic vision: Painterly rendering using computer vision techniques. In *Proc. of Intl. Symposium on Non Photorealistic Animation and Rendering (NPAR '02)*, pages 83–92, 2002.

[7] P. Haeberli. Paint by numbers: Abstract image representations. *ACM Trans. on Graphics (Proc. of ACM Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH '90)*, 24(4):207–214, 1990.

[8] J. Hays and I. Essa. Image and video based painterly animation. In *Proc. of Intl. Symposium on Non Photorealistic Animation and Rendering (NPAR '04)*, pages 113–120, 2004.

[9] A. Hertzmann. Paint by relaxation. In *Proc. of Computer Graphics Intl. (CGI '00)*, pages 47–54, 2001.

[10] A. Hertzmann. A survey of stroke based rendering. *IEEE Computer Graphics and Applications*, 23(4):70–81, July 2003.

[11] A. Hertzmann and K. Perlin. Painterly rendering for video and interaction. In *Proc. of Intl. Symposium on Non Photorealistic Animation and Rendering (NPAR '00)*, pages 7–12, 2000.

[12] S. Hiller, H. Hellwig, and O. Deussen. Beyond stippling - methods for distributing objects on a plane. In *Proc. of Eurographics '03*, volume 22(3), pages 515–522, 2003.

[13] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[14] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. In *In Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR '99)*, page 1274, 1999.

[15] T. R. Jones. Efficient generation of poisson-disk sampling patterns. *Journal of Graphics Tools*, 11, 2006.

[16] A. Kokaram, F. Pitie, R. Dahyot, N. Rea, and S. Yeterian. Content controlled image representation for sports streaming. In *Proc. of Content-Based Multimedia Indexing (CBMI '05)*, 2005.

[17] A. C. Kokaram. *Motion Picture Restoration: Digital Algorithms for Artefact Suppression in Degraded Motion Picture Film and Video*, chapter 2. Springer Verlag, 1998.

[18] P. C. Litwinowitz. Processing images and video for an impressionist effect. In *Proc. of ACM Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH '97)*, pages 407–414, 1997.

[19] M. Mignotte. Unsupervised statistical sketching for non-photorealistic rendering models. In *Proc. of Intl. Conf. on Image Processing (ICIP '03)*, volume 3, pages 273–577, 2003.

[20] Y. Park and K. Yoon. Painterly animation using motion maps. *Graphical Models*, 70(1-2):1–15, January 2008.

[21] A. Santella and D. DeCarlo. Abstract painterly renderings using eye-tracking data. In *Proc. of Intl. Symposium on Non Photorealistic Animation and Rendering (NPAR '02)*, pages 75–82, 2002.

[22] A. Secord. Weighted voronoi stippling. In *Proc. of Intl. Symposium on Non Photorealistic Animation and Rendering (NPAR '02)*, 2002.

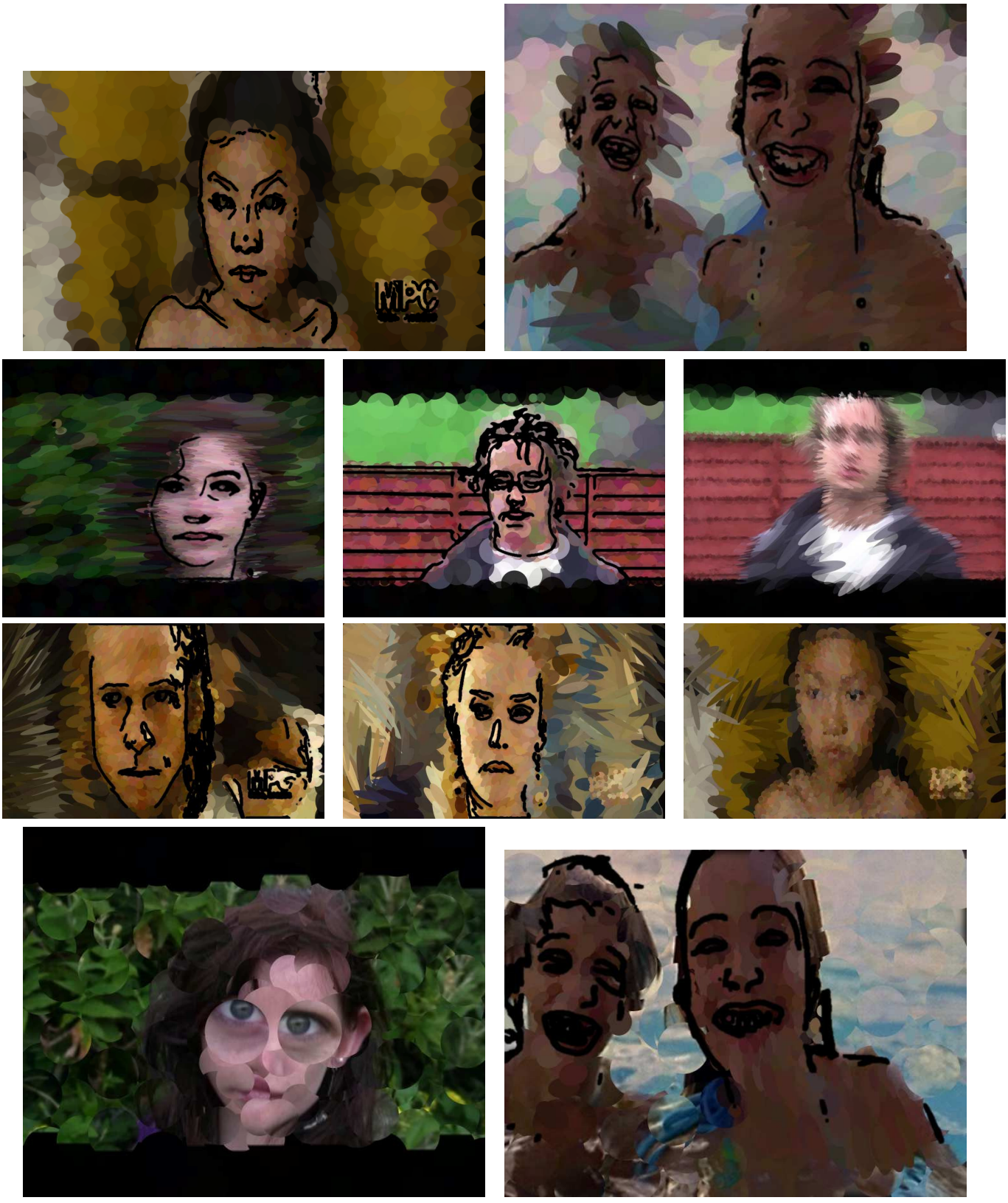


Figure 14: Example frames from stylized videos; (t-b,l-r) Hollywood_10fps_1, Swimming_3, Female1_2, Male1_1, Male1_2, Hollywood_10fps_3, Hollywood_10fps_3, Hollywood_10fps_2, Female1_3, Swimming_2