Mixed Media Painting and Portraiture

Stephen Brooks

Abstract— This paper presents a technique for mixed media non-photorealistic painting and portraiture. The goal of this work is to transform digital images into renderings that approximate the appearance of mixed media artwork, which incorporates two or more traditional visual media. We achieve this by first separating an input image into distinct regions based on the degree of local detail present in the image. Each region is then processed independently with a user-selected NPR filter. This allows the user to treat highly detailed regions differently from regions of low frequency content. The separately processed regions are then smoothly fused in the gradient domain. In addition, we extend our work to the rendering of mixed media portraits. Portraits pose unique challenges that we address with our method of segmentation, which is based on a composite of face detection and image detail. Our approach offers the user a great deal of flexibility over the end result, while at the same time requiring very little input. This input takes the form of a few simple and discrete choices. The results demonstrate an impressive array of transformational possibilities.

Index Terms— Mixed media, non-photorealistic rendering, image detail, portraits, face detection.

1 Introduction

We present a novel method of non-photorealistic rendering which transforms digital images into mixed media renditions that approximate the appearance of mixed media artwork. Such mixed media compositions incorporate two or more traditional visual media such as oil, watercolor, tempera, ink or pencil.

But this begs the question: why mix media either digitally or with real materials? Typically, artists explore mixed media when a single medium by itself is insufficient to fully develop an artistic idea [1], with “certain creative effects…only possible by fusing various media” [2]. One can take “advantage of the special qualities of each medium and come up with a total that is greater than all its parts” [3]. By mixing materials the artist is able to explore new channels of artistic expression or rejuvenate existing ones. The combination of often disparate elements can be used to create dynamics within a painting that would otherwise be difficult or impossible with a single medium. Another important use of mixed media is the emphasis of scene features or the contrasting of multiple features.

Mixed media artwork has a long and distinguished history within the fine arts. The earliest examples that could be considered mixed-media include illuminated manuscripts from Egyptian, classical, early Christian, and Byzantine sources. These works often featured combinations of gold leaf along with various types of ink. One of the first modern European painters to create mixed media artwork was Giotto di Bondone (Italian, c. 1266/76-1337) who combined tempera painting with the metallic shine of gold leaf to create a vibrant effect. Later, 15th century artists such as Jan van Eyck (Flemish, c. 1390?-1441) explored the use of oil and tempera.

Drawing media are another natural group of materials for artists to mix for sketches and other preliminary work. Rembrandt van Rijn (Dutch, 1606-1669) created preliminary studies using a combination of pen, bistre washes and gouache. Other notable pioneers of mixed media in the 19th century are Edgar Degas (French, 1834-1917) and Henri de Toulouse-Lautrec (French, 1864-1901). But it is in the 20th century when mixed media techniques proliferate, with an increasing number of artists choosing to explore its potential. These artists include Pablo Picasso (Spanish, 1881-1973), Suzanne Duchamp (French, 1889-1963) and Max Ernst (German, 1891-1976), to name a few. Mixed media has now reached beyond the fine arts into illustration and comics. A strong example of which are the works of Dave McKean (England, 1963-). This considerable tradition of mixed media suggests that the potential of mixed media should also be explored in the digital realm. But first we should consider in more detail how mixed media is used.

The extensive use of mixed media is matched by the plurality of approaches used by artists, since “mixed media techniques vary greatly from one to another” [4], not being “chained to dogmatic” [2] or “fixed rules” [4]. However, it is possible to enumerate the more common practices:

1. creating collages with paint, ink and printed matter,
2. treating detail with separate media, and
3. using one medium as an underlying base for another.

Yet, these separate cases are not mutually exclusive and can often blur in practice. In particular, the 3rd case often blurs into the 2nd case, since is it not uncommon for a base media to be left exposed in background or low-detail areas.

For concrete examples of how media is applied it is informative to consult instructional books [3, 4, 5] on the topic. As our system is focused on the selective application of media based on detail levels, we now list a few related instructional examples:

- “Once the basic drawing has been created [with colored wax], watercolor is applied with a brush to the background in order to make the flower arrangement

- *S. Brooks is with the Faculty of Computer Science, Dalhousie University, Halifax, Canada, B3H 2Y5. E-mail: sbrooks@cs.dal.ca*

stand out more" [4].

- Use watercolor for "broad areas of color" and "aquarelle pencils" to "add further details", along "with pastel sticks" for "sharpening up the shape of the petals". In the "finished work there are certain characteristics of drawing and others of painting revealed" [4].
- "Flowers, fruit, trees, grasses, stone, dirt, bark… can be made to look vibrant… using a combination of ink watercolor and colored pencil". "You'll be applying ink with a pen in the small detailed areas of your work". If something "needs punching up, try adding detail and sharpness to it with colored pencil" [3].
- "Paint a leaf in watercolor and add pencil details" [5].

Analogous to real mixed media artwork, we argue that artificial mixed media provides the user with greater flexibility when a single media is deemed insufficient. By mixing simulated media filters, the digital artist can combine disparate elements to create contrasts within an image that would otherwise be difficult to achieve. Our system focuses on the selective application of media based on detail, inspired by approaches similar to the specific cases just described. The system's capabilities are best understood with an example. Figure 1 shows the transformation of a still life image into a composite rendering, which exhibits aspects of watercolor, oil painting and colored pencil cross-hatching. To accomplish this we analyze the input image, computing a measure of image detail at each pixel. The detail levels are clustered, effectively separating the image pixels into distinct subsets that are then processed independently with separate user-selected NPR filters. The filtered image subsets are then seamlessly combined.

The primary contribution of this paper is a unified method for the intelligent and seamless mixing of NPR filters, which requires little input from the user. In particular, we perform image segmentation based on the Gaussian Mixture Model (GMM) clustering of multiscale image details. We also provide simple user controls over the filter mixing process with regard to both color and the gradient blend. For portraits, this is extended to a disjoint composite of image detail and face detection. Moreover, to determine a suitable "facial area", we present a tailored solution that combines skin filtering, graphcuts, AdaBoost and nonparametric clustering. We have also experimented with a face-classifier voting scheme using three AdaBoost variants.

2 Previous Work

Our method of mixed media NPR incorporates a number of stages including image detail analysis, clustering, NPR filtering, blending by optimization and, in some cases, face detection. Each of these areas has its own branch of related work and the number of related publications is truly vast. We will now review only a selection of papers that relate most strongly to the present topic.

NPR has experienced what could be described as a flood of activity in recent years. There is a wide array of NPR systems that attempt to emulate a variety of artistic styles [6]. But, it is possible to impose structure onto this large body of work as NPR systems vary along a basic 3 lines: rendering methods, levels of interactivity and models of image structure.

Rendering methods can often be further broken down into physically vs. algorithmically based. Many artificial drawing techniques focus on the imitation of traditional artistic media and a common approach is to physically simulate the media [7]. Complementing the simulation of paints is the modeling of virtual brushes [8] for the interactive painting of artistic imagery. Pen-and-ink illustration styles have also been algorithmically imitated for the rendering of surfaces [9]. Other work by Bangham *et al.* [10] generates reduced-detail painterly renderings and sketches based on scale-space filtering. Our method can be both algorithmically and/or physically generated depending on the selection of NPR component filters.

Regarding interactivity, the pioneering paint system of Haeberli [11] allows a user to direct the creation of impressionistic paintings and a number of interactive sketch systems followed [12]. In these interactive systems, the user's input is often supported by varying degrees of automation [13]. The work of DeCarlo and Santella [14] could be considered an extreme form of semi-automation as their system employs only gaze analysis to guide the artistic rendering. Fully automatic techniques also exist and often apply some form of image analysis [15], [16]. Our method is based on detail and facial analysis and is almost fully automated; it only requires that the user select which filter is to be applied to each level of detail. Though, we also provide optional controls targeted for experienced users.

NPR systems also differ with respect to how they model image structure, whether they exploit variability of detail within the scene or explicitly segment the scene into regions, edges and/or objects. Prior work has exploited detail in images in various ways. Returning to the work by Hertzmann [16], this system progressed the rendering of painterly images adaptively using B-spline strokes of decreasing size, aligned to normals of image gradients. Smaller strokes are applied in a course-to-fine manner, with finer strokes only painted in areas where the rendering differs from the blurred source image. It is also worth noting that the system is parametric, allowing the user to approximate styles by altering global attributes such as brush size, opacity and color jitter.

Work related to [16] by Collomosse and Hall [17], that is currently in press, produces convincing impasto style renderings using a novel genetic algorithm-based relaxation process to place strokes. In addition, their work introduces a new trainable measure of perceptual saliency which drives the rendering process. The work also goes beyond prior work by not just adjusting the size of strokes but also their accuracy based on the type of salient feature (ridge or edge). However, these related systems do not produce the kind of results shown in this paper, and as we are separately processing regions of details with different NPR filters rather than producing individual strokes, our method requires the clustering of detail levels into areas. Our system therefore completely decouples the rendering method from the rest of the system. We also must handle the merging of separate regions and, as will be seen, we must treat faces as a distinct case.

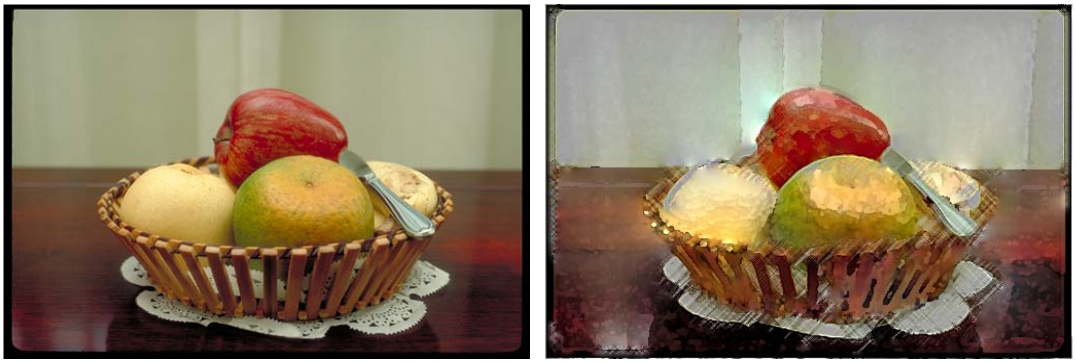

Fig. 1: A real image is converted into a mixed media rendering style. The still life image of a bowl of fruit (left) is adaptively re-rendered (right) using a composite of oil painting, watercolor and colored pencil cross-hatching.

Returning to DeCarlo and Santella [14], their system uses eye tracking data as a measure of image saliency. Using this measure of image importance, the system is able to prune a hierarchy of image segments and produce effective renderings with bold edges over smoothed regions of constant color. Beyond the obvious difference of not making use of eye-tracking data to guide the process, our system again differs in that it fuses separately styled regions and uses face detection for human portraits. In doing so we are able to produce a variety of combinational styles rather than one single style. An additional benefit of this approach is that it leverages existing filters and can easily incorporate new filters as they are developed.

Another system that requires specialized hardware is the compelling concept of an NPR camera [18]. Here a camera is used with an array of flashes to derive a depth map of the scene. Discontinuities are detected in the map, disambiguated from material discontinuities. Once detected, it produces renderings with smoothly colored regions outlined with salient contours. However, the system does not produce mixed renderings, requires a special camera and, as they note, has difficulties with bright outdoor or distant scenes.

There have also been attempts at incorporating NPR elements into real video in order to produce a cartoon animation effect [19] or highlight visual information in nighttime video [20]. Raskar *et al.* [20] also present NPR lighting effects by combining video frames that have been taken under different lighting conditions. However, these NPR effects are only applicable to video with dynamically changing illumination and, again, are not capable of producing any of the results shown in this paper.

Other systems have combined various rendering elements for the display of 3D scenes. Gooch *et al.* [21] combine Gooch shading with silhouettes to accentuate geometric information for technical illustrations, while Tietjen *et al.* [22] combine line rendering and volume rendering for rendering medical visualizations. Halper *et al.* [23] have also proposed a framework (OpenNPAR) that integrates both 2D and 3D NPR modules into a single rendering pipeline.

Software such as Adobe Photoshop® and The GIMP also incorporate a number of NPR filters that target specific NPR styles. Additionally, there exist systems that are focused on the interactive generation of NPR imagery, such as Corel Painter™. Another lesser-known system, named Segmentis Buzz™, is able to segment, simplify and combine regions for painterly effects. However, it should be noted that the complete details of algorithms underlying commercial systems are not publicly available.

Another related area is the fusion of multiple images. The optimization process that underlies Poisson Image Editing [24] allows the seamless integration of multiple image elements under user guidance. Although we also solve Poisson equations to blend separate NPR regions, our overall system is more automated and tailored to a different purpose. The present work also relates to the photomontage system of Agarwala *et al.* [25] that has been applied to a number of tasks such as relighting, selective compositing and panoramic stitching. But our method differs in that we auto-segment the image according to image detail and face detection, with different user-selected NPR filtering operations applied separately to these regions. Moreover, our system is more accessible as we do not require the user to supply a stack of input images. Importantly, we are also solving a problem that has not yet been addressed – image-based mixed media.

As our method also extends to portraiture, prior work on human face detection is also relevant [26], [27], [28]. In addition, a number of systems have been developed that process or generate facial imagery. The pioneering work by Lance Williams [29] generates facial animation, driven by a performer's face that was later extended to cartoon characters [30]. However, neither system generates mixed media renderings and both require the artificial placement of dots for facial tracking. Work has also been conducted that makes use of perceptual characteristics to composite facial expressions [31] or generate sketches [32]. Though, once again, neither of these systems produces mixed media output nor do they automatically detect the location of faces on arbitrary backgrounds.

Chen *et al.* [33] have developed an example-based system, which produces portrait line drawings. Though successful in constructing line drawings, questions remain concerning the extendibility to alternate styles and its ability to handle a broader range of ages, genders and races. Other aspects that differ from the present work include a reliance on blank backgrounds and a dependence on the availability of example sets produced by human artists.

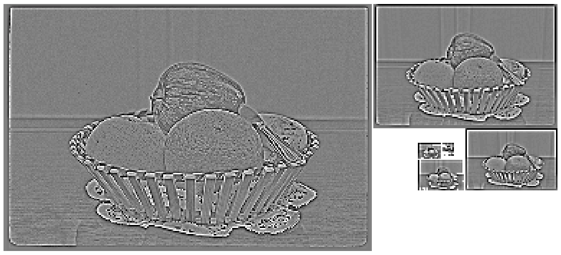

Fig. 2. Multiple levels of a Laplacian pyramid.

Impressive results have also been achieved by Victor Ostromoukhov [34] that generate digital face engravings, but with considerable reliance on user direction. Creating caricatures from realistic faces has proved to be another popular topic [35], [36], as has the non-photorealistic rendering on 3D facial models [37].

Finally, concurrent work [38] on the automatic generation of collages has used human face detection to ensure that if a face is included in a collage then it is retained as a whole. In addition to being concurrent, our work on faces differs since we extend the method of face detection using Gaussian Mixture Models, skin detection and a voting AdaBoost approach. And, of course, we also have an entirely different purpose.

3 Method Overview

We now present a method for the seamless mixing of NPR filters, which requires minimal input from the user. As we are mixing NPR transformations, it must be determined which pixels are to be transformed by which filters and the results of these separate filtering operations must be fused in a seamless fashion. Our solution requires multiple stages and we summarize the key steps as follows:

1. Analyze the input image, *I*, by computing a measure of image detail at each pixel, $D(x, y)$.
2. Segment the input image into pixel clusters, $S_1..S_n$, which correspond to n levels of detail.
3. Perform morphological closing on each of the clusters separately, producing $M_1..M_n$.
4. Allow the user to select an NPR filter for each of the detail levels. The filters are applied separately, producing filtered regions, $F_1..F_n$.
5. Fuse the separate regions by solving Poisson equations, yielding the final result, *F*.

Each of these stages will now be clarified in sections 3.1-3.6 and we will subsequently extend our discussion to the generation of portraits in section 3.7.

3.1 Image Detail

The initial stage for transforming the input image into a mixed media NPR rendering involves grouping the pixels of input image, *I*, into distinct subsets, based on levels of detail. For example, we might separate the image into three regions with low, medium and high detail. This effectively separates the image into distinct subsets, which are then processed independently.

We therefore need a measure of the 'amount' of detail at each location in the image. We show that the summed responses from a Laplacian pyramid produce sufficiently good results for our system. Laplacian pyramids are a decomposition of the input image into a hierarchy of images such that each level corresponds to a different band of image frequencies [39]. It is termed a Laplacian Pyramid since each level, h_i , is computed by subtracting a blurred copy of h_{i-1} from h_{i-1}, and downsampling. This is approximately equivalent to convolving h_{i-1} with the Laplacian of a Gaussian blurring filter. Further details of the decomposition can be found in the original paper. A decomposition of the fruit bowl image is shown in figure 2. As seen in the figure, we decompose the image into multiple scales of detail.

Once the pyramid is constructed, we then sum the responses for each scale at each pixel. This produces a measure of image detail:

$$D(x,y) = \sum_{i=1}^{L} h_i\left(x_1 / 2^{l-1}, y_1 / 2^{l-1}\right) \qquad (1)$$

where h_i is level i of the Laplacian pyramid, and L=4 is the number of levels used. Note that when summing the 4 pyramid levels, the higher levels are upsampled to the original image size with bicubic interpolation. We use smooth interpolation to avoid any blockiness in the computed detail map. The details map for the fruit bowl image is shown in Figure 3. Next, we will cluster the pixels based on $D(x, y)$.

3.2 Pixel Clustering

Once the detail is measured, we can group the input image's pixels into clusters, $S_1..S_n$, which correspond to various levels of detail. We achieve this by clustering the data with a Gaussian Mixture Model (GMM) which partitions the data points, $D(x, y)$, into n disjoint subsets, $S_1..S_n$ [40]. GMMs are a statistical generalization of k-means clustering and in 1D define each cluster as a Gaussian, each with its own mean, μ_k, and a variance, ϕ_k. We compute the density of component k as:

$$p(x \mid \mu_k, \phi_k) = \frac{1}{\sqrt{(2\pi) \mid \phi_k \mid}} \; e^{-(x-\mu_k)^2 / (2\phi_k)} \qquad (2)$$

and the mixture density as:

$$f(x) = \sum_{k=1}^{n} a_k \; p(x \mid \mu_k, \phi_k) \qquad (3)$$

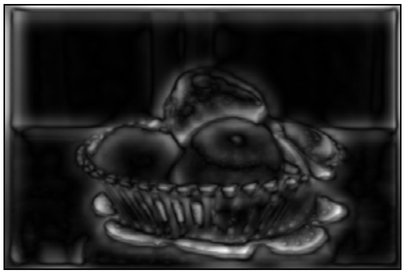

Fig. 3. Detail map for fruit bowl image, scaled to a range of 0 to 255.

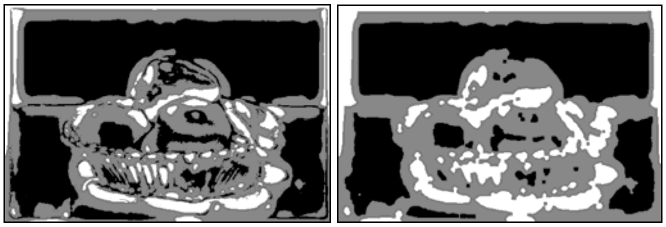

Fig. 4. Clustered sets of pixels before (left) and after (right) morphological closing is applied. In this case, three detail levels are shown as black, grey and white, respectively.

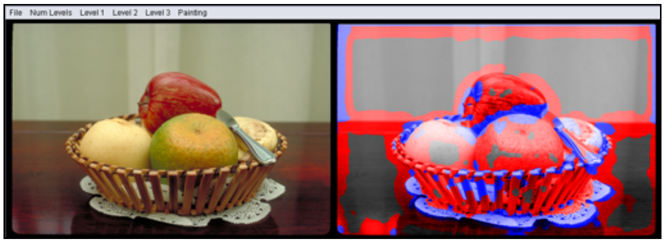

Fig. 5. User is shown original image and color-coded segmentation. User is able to optionally over-ride automatic segmentation by direct painting.

where the prior probability (weight) of component k is a_k. The parameters (a_k, μ_k, ϕ_k) are estimated by the maximum likelihood (ML) criterion using the Expectation Maximization (EM) algorithm [41].

Since we are working in 1-D, one could argue that k-means clustering might suffice. However, GMM tend to produce more accurate results. We compared the segmentation of using both clustering methods and, importantly, we did not find any degenerate clustering results using the GMM that would indicate that it is not sufficiently robust.

The computed levels of detail for our fruit bowl image is decomposed into three regions (n=3), shown to the left in figure 4. For this example, we refer to $S_1..S_n$ as low, medium and high detail regions. We next refine the clusters by performing morphological closing on the regions, which effectively removes small holes from the clusters and makes the regions more contiguous. Our structuring element is a disk with a radius of 5 pixels. Closing is usually applied to binary images, however, we have n discrete levels for the clusters, $S_1..S_n$. We therefore apply the closing operation in (n-1) recursive stages. First we apply closing to the binary image that has 1's at pixels belonging to ($S_2 \cup S_3 \cup \ldots \cup S_n$), producing M_2. Closing is then applied to the binary image that has 1's only at pixels belonging to ($S_3 \cup \ldots \cup S_n$), producing M_3. This recursive process continues until M_n is produced. The final clustering is composited, starting with M_1 covering the entire image, then adding the modified M_2, $M_3 \ldots M_n$ areas sequentially, in order. For our three level example, the final clusterings, $M_1..M_3$, are shown to the right in figure 4.

Although we find that this approach produces good results, the user is given the option of modifying the region clusterings for greater flexibility. Figure 5 shows the interface with the original image to the left and a grayscale version shown to the right. This grayscale version is color-coded according to the detail clusters. At any time the user can 'paint' with one of the cluster colors to override the default segmentation. The cluster painting color is chosen from the painting menu. We do stress, however, that none of the results in this paper required manual editing.

3.3 Region Filtering

Region filtering is the only stage that *requires* user input. Here the user is able to select a different NPR filter for each of the levels of detail, $M_1..M_n$, producing filtered regions $F_1..F_3$. In figure 6 is a selection of off-the-shelf NPR filters that we use in this paper, drawn from Adobe Photoshop® and The GIMP. All filters are used with default settings.

The user interface consists of simple menus with discrete choices. The first determines the number of levels of detail into which the image is decomposed. The number of levels selected determines the number of filter menus that appear. Using the filter menus the user can choose a different filter for each level, the same filter at multiple levels or opt to use no filtering at one or more levels. Figure 7 shows the simplicity of the interface. We also allow the user to save mixed media combinations as presets, which can be later applied to other images.

3.4 Fusing Regions by Optimization

In many works of mixed media, the various media are blended. For example, in [4], "delicate washes are fused with lines from the aquarelles and the pastels", and in [5] the reader is instructed to "gently merge soft pastel into a watercolor wash, so that it resembles a wet-into-wet wash" or to "draw some felt-tip spirals, then wash over to give a diffuse and misted effect". And so, to mimic this aspect of mixed media, after the regions are filtered (see figure 8) we

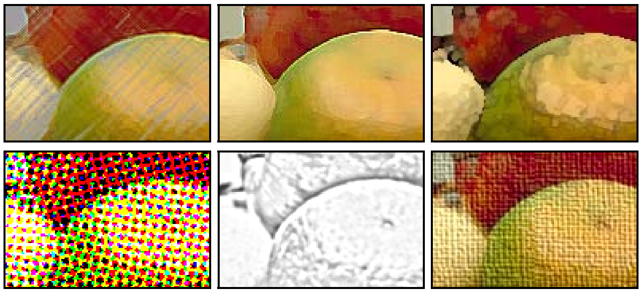

Fig. 6. Example filters applied to the input image. 1st row: three Adobe Photoshop® filters (colored pencil, paint daubs, watercolor). 2nd row: three GIMP filters (newsprint, photocopy, apply canvas).

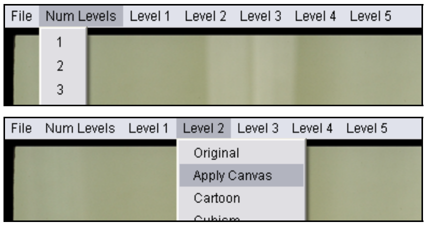

Fig. 7. Top: the user selects the number of detail levels. Bottom: the user (optionally) selects an NPR filter for each detail level.

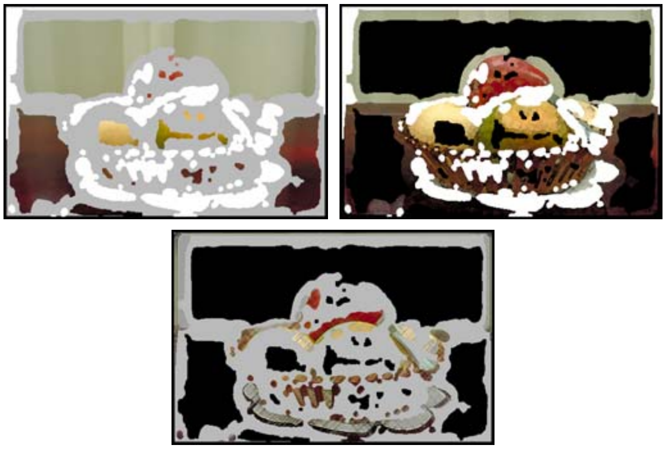

Fig. 8. An example with three levels of detail used. Top left: low detail with oil filter applied. Top right: medium detail with watercolor filter applied. Bottom: high detail with colored pencil filter applied.

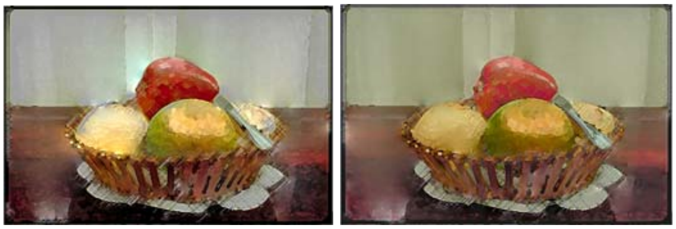

Fig. 9. Using colors from the filtered (left) and original image (right).

fuse the separate regions together in a seamless fashion without significantly altering the content of each region. For this we adapt the work of Pérez *et al.* [24].

The optimization process seamlessly inserts new content into a subset, Ω, of an existing image, *h*. It computes a new image, *f*, whose gradient, ∇f, within Ω is closest to the gradient ∇g, taken from a second image, *g*. The original boundary, ∂Ω, of region Ω from *h* is also used as a constraint. It is this additional constraint that ensures that the region Ω blends with the surrounding image, *h*. The final image contains an interpolation of the boundary conditions, ∂Ω, inwards while conforming to the spatial changes of the guidance field from *g* as closely as possible within Ω. The minimization problem is written as:

$$\min_{f} \iint_{\Omega} \left|\nabla f - \nabla g\right|^2 \text{ with } f\big|_{\partial\Omega} = h\big|_{\partial\Omega} \qquad (4)$$

The reader is directed to [24] for the discretization of the problem and for suggested iterative solvers.

In a similar fashion to the application of binary closing, we perform the gradient blending in recursive stages. First, we fuse region F_2 into F_1 using ∇F_2 for the gradient ∇g and F_1 as bounding image *h*. This produces an intermediate image T_1. We then fuse F_3 into T_1 using ∇F_3 for the gradient ∇g and T_1 bounding image *h*. This yields T_2. The process continues recursively until F_n is fused into T_{n-1}.

We compute the region fusing in separate stages in order to allow the user to see the intermediate processing. We argue that this would be closer to how an artist produces mixed media artwork – in stages. In practice, it facilitates the exploration of filter combinations as the effects of each filter are more easily discerned.

3.5 Color Control

We can optionally offer more control to the user, particularly with regards to the color properties of the final result. We proceed with the observation that many of the component filters used in the system alter the colors found in the original image. A simple approach to color control gives the user a discrete choice as to whether the filtered colors or original colors are used. This is easily achieved by computing a new image *C* that uses the greyscale texture from *F* and the color information from *I*. Moreover, we can offer the user a slider that controls the relative weighting of color between *F* and *I*:

$$C = \alpha F + (1-\alpha)(L_F + I - L_I) \qquad (5)$$

Where α is the weighting controlled by the slider; L_F and L_I are the luminance values of *F* and *I* respectively. The above equation is calculated in L α β color space [42]. Figure 9 shows an example of taking color from *F* (left) and *I* (right).

We note that a beneficial aspect to this color control is that it can be used to eliminate color bleeding between regions, if the user so desires. Examining the left image of figure 9 shows that the colors have bled spatially across the image. The right image shows how that color bleeding has been removed.

3.6 Blending Control

In addition to color control, we can provide the user with control over the extent of region blending, ranging from a hard edge composite (figure 10, 2nd row) to a full gradient blend (figure 10, bottom). This control takes the form of a slider that shifts the result between these two extremes.

As we wish the response of the slider to be interactive, we cannot repeatedly modify and recompute a full gradient blend by optimization as it is not sufficiently quick. Instead, we construct a secondary mask with a linear outer gradient (figure 11, middle) from the original mask (figure 11, left) for a given region. This secondary mask is used to transition between a hard-edged composite and the original gradient blend. We can interactively control the amount of transition with the width of the linear gradient. And so, this could be considered a linear transition over a non-linear blend. The boundary control can be provided as a single slider to the user, or as a set of *n*-1 sliders for independent control of each boundary when *n* levels of detail are used.

3.7 Portraits

In general, our method offers a good solution for simulating mixed media renderings. However, our approach fails when processing human faces, an example of which is shown in figure 12. As can be seen, using only image detail segmentation (bottom left) does not respect the integrity of the face since the areas around the cheeks and forehead are interpreted as regions of low detail, while the eyes and mouth contain higher frequencies. In our experiments, this appears to be a problem unique to faces, since not surprisingly, the human viewer is highly sensitive to the treatment of faces. We now discuss how we can dramatically improve

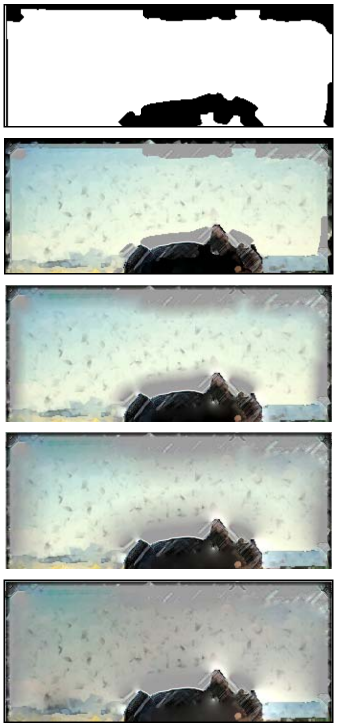

Fig.10. Top: mask of regions. 2nd row: a hard-edged composite. 3rd and 4th rows: two in-between states. Bottom: a full gradient blend.

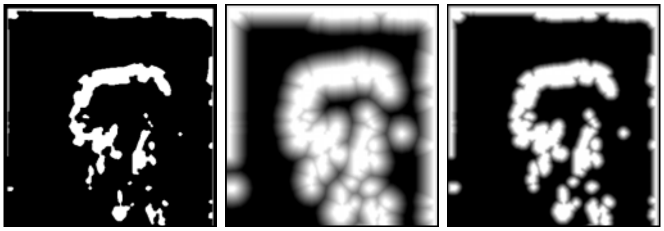

Fig. 11. Left: original mask for high detail area shown in white. Center: mask with linear outer gradient. Right: mask with smaller gradient.

upon these initial results by detecting and filtering the face as an integral unit (bottom right).

Recently, a successful real-time face detection approach has emerged based on the AdaBoost [43] that can quickly and fairly robustly detect the presence of faces [44]. However, we are not concerned with real-time face detection since we are not under strict time constraints. This implies that we can perform additional processing if necessary. Moreover, their system is only designed to detect the faces with a bounding box which is not sufficient for our purposes, as rectangular-filtered region would be rather conspicuous in the final result. We instead wish to detect a general “facial area” and one that does not necessarily have clean, regular boundaries.

We require a tailored solution to face detection, using AdaBoost as an initial starting point but adapting this core algorithm for our needs. We have opted to augment AdaBoost with skin detection and GMM clustering. In addition, we have experimented with a voting scheme that employs three AdaBoost variants: Real AdaBoost [45], Gentle AdaBoost [46] and Modest AdaBoost [47] which we show to produce slightly improved results than either variant alone for face detection.

3.7.1 Skin Filtering

To begin, we first narrow the range of possible face locations with a skin filter [48], [49]. In particular, we present a modified skin filter based on [48] but disregard the corresponding geometric filter, as we are not searching for connected limbs.

The skin filter begins by subtracting the image’s zero response and the resulting RGB values are transformed into a log-opponent color space, wherein (*R*,*G*,*B*) tuples are converted into (*log*(G), *log*(*R*) - *log*(G), *log*(*B*) - (*log*(*R*)+*log*(G))/2). The color component of this particular filter passes a restricted range of hues with moderate saturation. Skin is also filtered by detail level since most skin regions exhibit low texture amplitudes. Here we use a different approach than in [48] for the texture component of the skin filter. Instead of simply using the difference values of the image from a smooth filtered version of itself, we use the multiscale responses from the Laplacian pyramid as discussed in section 3.1. The skin filter is tuned conservatively, in order to take into account a variety of skin pigments. The output mask, $s(x, y)$, of this stage of the skin filter process can be seen in figure 13 (left).

In our system, we proceed further, reducing the output of the skin filter by retaining only those pixels that are surrounded by at least 50% skin pixels. This requirement is computed over a local 20 x 20 neighborhood, which is chosen to be slightly smaller than the 24 x 24 face window (see below). This is quickly computed with a median filter over the binary skin mask. We additionally perform hole filling on the filter output, in case certain features, such as eyes, are not completely filled in by the median filtering. The mask output, $s_m(x, y)$, of this final skin filter stage can be seen in figure 13 (right).

3.7.2 Face Point Detection

We next examine each location that has passed the skin filter as a potential face location, $f(x, y)$, using a voting AdaBoost classifier approach. In their original work, Viola and Jones [44] use the AdaBoost [43] to build an efficient classifier for face detection. The classifier is comprised of a set of weak classifiers, which are primitive Haar-like filters. The number of possible Haar-like filters is immense and so the learning task is to reduce the number of possible filters to a small set that best discriminate faces from non-faces, and also determine the threshold for each weak classifier.

We have trained three separate models to learn 30 weak classifiers with 4000 pre-classified faces and 4000 non-faces. The three models are trained with three AdaBoost variants (Real, Gentle and Modest), which are incorporated in an unwieghted voting scheme. The error rate over 60 training interations for voting AdaBoost is shown in figure 14. The false-positive (FP) and false-negative (FN) rates for our voting system is (FP=0.034; FN=0.033) after 60 iterations, which is a slight improvement over any of the three individual methods: Real (FP=0.037; FN=0.039), Gentle (FP=0.039; FN=0.039) and Modest (FP=0.046; FN=0.044).

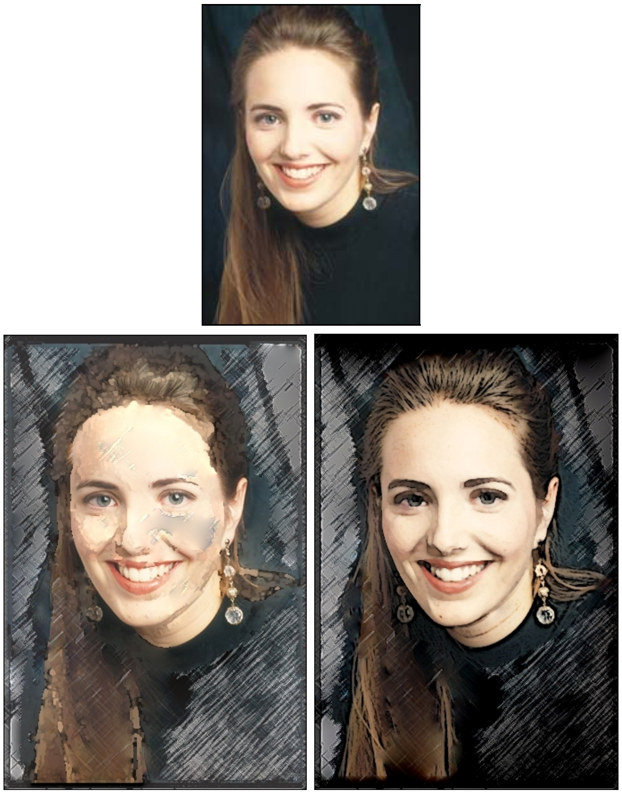

Fig. 12. Initial failure: using only image detail segmentation (bottom left) does not respect the integrity of the face. Success: detecting and filtering the face as an integral unit (bottom right).

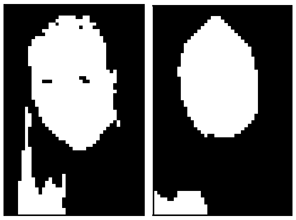

Fig. 13. Skin filter (left) and median filtered skin filter (right). Resolution shown at the pyramid level's size at which the face was found.

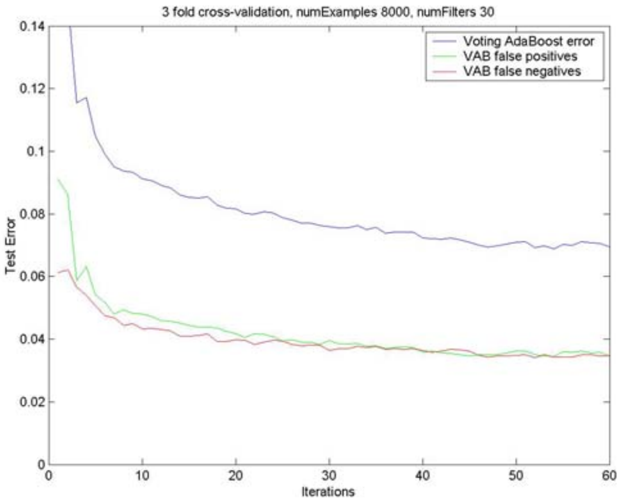

Fig. 14. False positive and false negative error rates for Voting AdaBoost face detection.

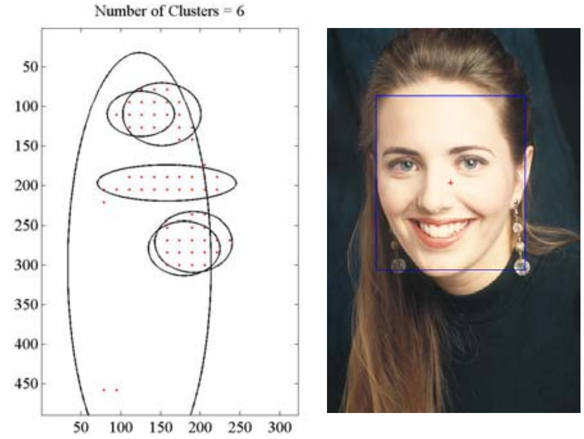

Fig. 15. Left: detected potential face points and the detected clusters. Right: best cluster found.

Once our face classifier is trained we apply it to our target image. This requires generating a Gaussian image pyramid and scanning across each level of the pyramid with a 24 x 24 window at all skin locations. For efficiency, we examine every 3 points in both the x and y directions, as there is considerable local coherence. Furthermore, we assume that the face window is a least 1% of the image area, which often eliminates the need to scan the lowest (most expensive) level of the Gaussian pyramid. We can make this assumption since there would be little point in performing special filtering on faces if they are of a tiny size. The centers, $f(x, y)$, of all windows that have been classified as faces proceed to the next stage.

3.7.3 Face Point Clustering

The AdaBoost-based algorithm is not perfect and can still produce false positives. In addition, translations of the face center can produce multiple redundant positives in the same area. To address this issue Viola and Jones [44] simply take the average of the corners of all face windows that are found. This is potentially problematic as false positives can be averaged into the face window. We instead cluster the potential face points $f(x, y)$ with a Gaussian Mixture Model (GMM) [40] and select the center of the Gaussian component with the largest weight as the face center. We are again conservative in that we select the face center at the highest level of the Gaussian pyramid to better ensure that we completely encapsulate the face region. The method proceeds as discussed previously regarding image detail clustering (see section 3.2), except in this case we are dealing with 2D datapoints, $\boldsymbol{x} \in f(x, y)$, which requires a modified density calculation for each Gaussian component k:

$$p(x \mid \mu_k, \phi_k) = \frac{1}{\sqrt{(2\pi)^2 \mid \phi_k \mid}} \; e^{-(x-\mu_k)^T \phi_k^{-1} (x-\mu_k)/\,2} \qquad (6)$$

However, there is now an additional unknown: the number of clusters, n. We estimate n with Rissanen's Minimum Description Length (MDL) criteria [50], which is an expression of Occam's razor wherein the model parameters are selected to minimize the sum of the model's complexity and the efficiency of the description of the training data with respect to that model. In our 2D case, the following is minimized:

$$MDL(n) = -\sum_{x \in f(x,y)} \log\left(\sum_{k=1}^{n} a_k \; p(x \mid \mu_k, \phi_k)\right) + \frac{6n-1}{2}\log(2M) \quad (7)$$

where M is the number of potential face points. The second term can be understood as a penalty against over fitting with too many clusters, n. Figure 15 (left) shows an example of clustering using the MDL criteria. The face detection window placed at the cluster center with the largest weight is shown to the right.

3.7.4 The Final Facial Area

The output of the previous section is simply a rectangular box that encapsulates the face, which is insufficient for our purposes. To produce a less conspicuous facial region, we first reduce the rectangle to the union of the skin mask, $s(x, y)$, and the hole-filled median skin mask, $s_m(x, y)$, as follows:

$$m(x, y) = s(x, y) \cup s_m(x, y) \quad (8)$$

In many cases this conservative skin filtering would be sufficient, but fails when non-face pixels in the rectangle fall within the range of skin colors. This issue can be addressed by performing an iterative graph cut as described in [51]. Specifically, we first modestly dilate the skin region and then use this as the initial bounding region for the graph cut process. This algorithm proved robust in practice, and further details of the process can be found in the original paper.

Figure 16 shows the final facial region for a few examples. What remains is to segment the remaining non-face regions by image detail. This is computed as before in sections 3.1-3.2 but only over the remaining non-face region. This yields a segmentation with a single face region and n detail-level regions.

Our system is fairly robust but is not completely perfect and in a few cases, it is possible that no face is detected at all. In this event we fallback on the hole-filled median skin filter, $s_m(x, y)$ of the entire image, and then perform an iterative graphcut on that image subset. This fallback solution may only work in cases where there is not a significant amount of skin-like color in the non-face regions in the image. It would also incorrectly classify other body areas such as hands, but this classification of other body parts is probably not a major concern.

4 Results

We present a further selection of results in figures 19 and 20. In particular we note a few striking examples. The first in figure 19 (1st row, middle) has rendered a starkly contrasted floral image with a mixture of impressionist, paint daubs and photocopy filters. A further result generated from the same flower image is shown to the right. Note how the results are quite varied and that good results from multiple combinations of filters can readily be generated for a given input image.

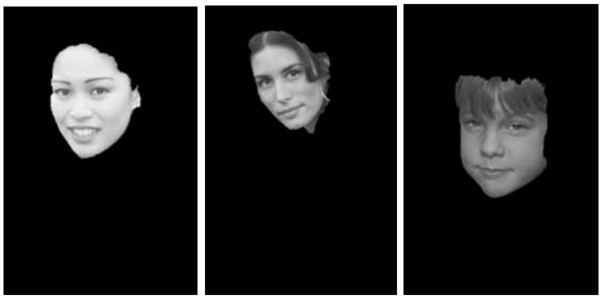

Fig. 16. The final face mask for a few examples.

For comparison we have provided the results of using individual off-the-shelf component filters in Figure 17. These filters have been used in various examples in this paper. We argue that our mixed media approach produces more complex and visually interesting results than these uniform treatments. And, importantly, our system certainly offers greater flexibility, producing a variety of results that simply could not be achieved with a single filter.

For example, the selective use of cross-hatching shown in the fruit bowl image of figure 1 and the lion image of figure 20 bears a sketch-like quality that suggests the result has been constructed in stages. Another interesting use of selective filtering allows us to give the appearance that the result has been painted on canvas. The flower example of figure 20 has been partially processed with a canvas filter giving the impression that an underlying canvas is showing through.

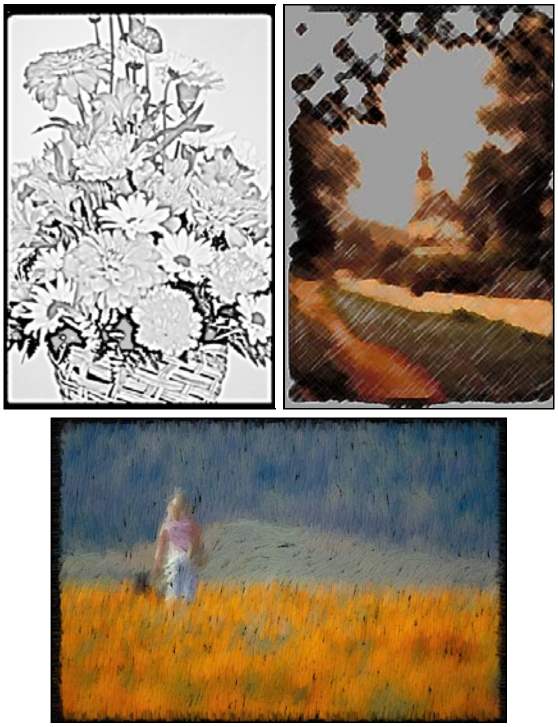

Fig. 17. Individual filters applied to various images.

An important benefit of mixing NPR filters is that it enables us to highlight the importance of finely detailed regions with a contrasting style. For example, in the bottom row of figure 19 the standing woman has been processed with a cartoon filter which crisply highlights her shape against the background of fields that have been processed with an impressionist filter. If the entire image is filtered with only the impressionist filter, the form of the woman becomes obscured in the strokes of the impressionist style. In this example, one might argue that our method has benefited from the existing depth of field effect in the image. But, we would note that in many photos, the regions of interest are intentionally made to be in focus. A further example of this intentionality is the flower example of figure 20. We consider it an advantage of our system that it can exploit this when present.

But often, we may simply want to treat highly detailed regions differently, even when no strong depth of field is present. If we consider the snowy image of figure 19, we are able to place the wavy impressionistic effect only within low frequency regions (sky and snowy ground). If we applied the impressionistic affect to all regions indiscriminately, then we would produce an overly filtered and obscured result, like the one shown in figure 17.

In some cases each of the three filtering operations is obvious; in others a dominant filter may be all the viewer notices. This parallels real mixed media works where one can often "perceive a dominant medium in the illustration" [1]. This does not necessarily imply that the other filters do not contribute to the overall effect nor that the introduction of the other materials in real mixed media are superfluous [1]. For example, the wavy impressionist filter applied to the low detail regions dominates the snowy image of figure 19. But, this is also supported by the more subtle paint-daubs filtering of the medium detail regions that has mainly affected the trees.

Returning to the lion example of figure 20, we note an unusual artifact at the top of the image. It appears that the colored pencil filter introduced a number of green pixels along the top boundary of the image. These local green pixels were then blended across the image in a subsequent gradient blend. Although this might simply be considered an artifact of the artificial black boundary in the original image, it is worth noting nonetheless. In any case, the color control as discussed in Section 3.5 can remove this type of artifact, if desired.

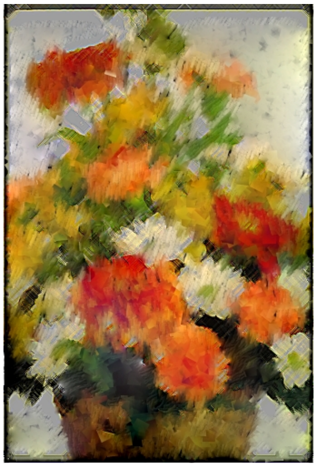

Fig. 18. An unfocused example using all *heavy* filters.

In general, our approach produces a variety of convincing results, though we would note that not all combinations of filters will be equally effective. For lack of better terminology, some component filters could be described as *mild* and others *heavy,* in terms of the degree of distortion of the original content. Good results can be achieved with a combination of *mild* and *heavy* filters, or *mild* and *mild* filters.

However, using a mixture of many heavy filters can produce results that might be described as unfocused. For example, in figure 18, is an example using three *heavy* filters: cross-hatching, cubist and impressionist. Although it is not inconceivable that a user might want this kind of effect for some application, using all heavy filters such as these can often produce an unfocused result. Better results can be obtained by retaining some amount of original detail.

Our experiments have indicated that separating the image into only three detail levels is often sufficient to generate a wide variety of interesting results. In fact, there are $(n+1)^3$ - 1 possibilities for n filters if we allow up to two detail levels to remain unfiltered. And so, for naïve users, we could restrict the interface so that only 3 levels of detail are used and name them low, medium and high.

Nevertheless, we have also generated further examples using 5 detail levels. An example of which is shown in the bottom row of figure 20. Impressive results can be achieved and an even greater number of possibilities exist with 5 levels. Though it does become increasingly difficult to discern the effects of each individual filter as the number of detail levels increases. This serves to highlight the need to fuse the detail regions in stages (as discussed in section 3.4) so that the user can observe the filter contributions incrementally.

A selection of portraits are shown in figure 21. The system is once again able to generate a variety of styles. Also note how by processing the facial areas with a separate filter the result can add additional focus to the portrait. One particularly challenging case is shown in the top-left of the figure. The challenge lies in the fact that most of the image is skin colored, meaning that skin detection alone would not be sufficient. However, the combination of AdaBoost-based face detection, skin filtering and graphcuts has correctly identified the facial region.

Our approach to the treatment of faces works well in practice, though face detection has not yet been perfected for 100% of cases. Nevertheless, like Rother *et al.* [38], we argue that automating the processing of images containing faces is worth the effort. Firstly, one could envisage applications where one does not wish to manually define image regions (i.e. in the context of a photo-booth, or when used by a naïve user). Secondly, there is inherent merit in progressing the field of NPR by including the special treatment of faces, even if perfect accuracy cannot *yet* be achieved.

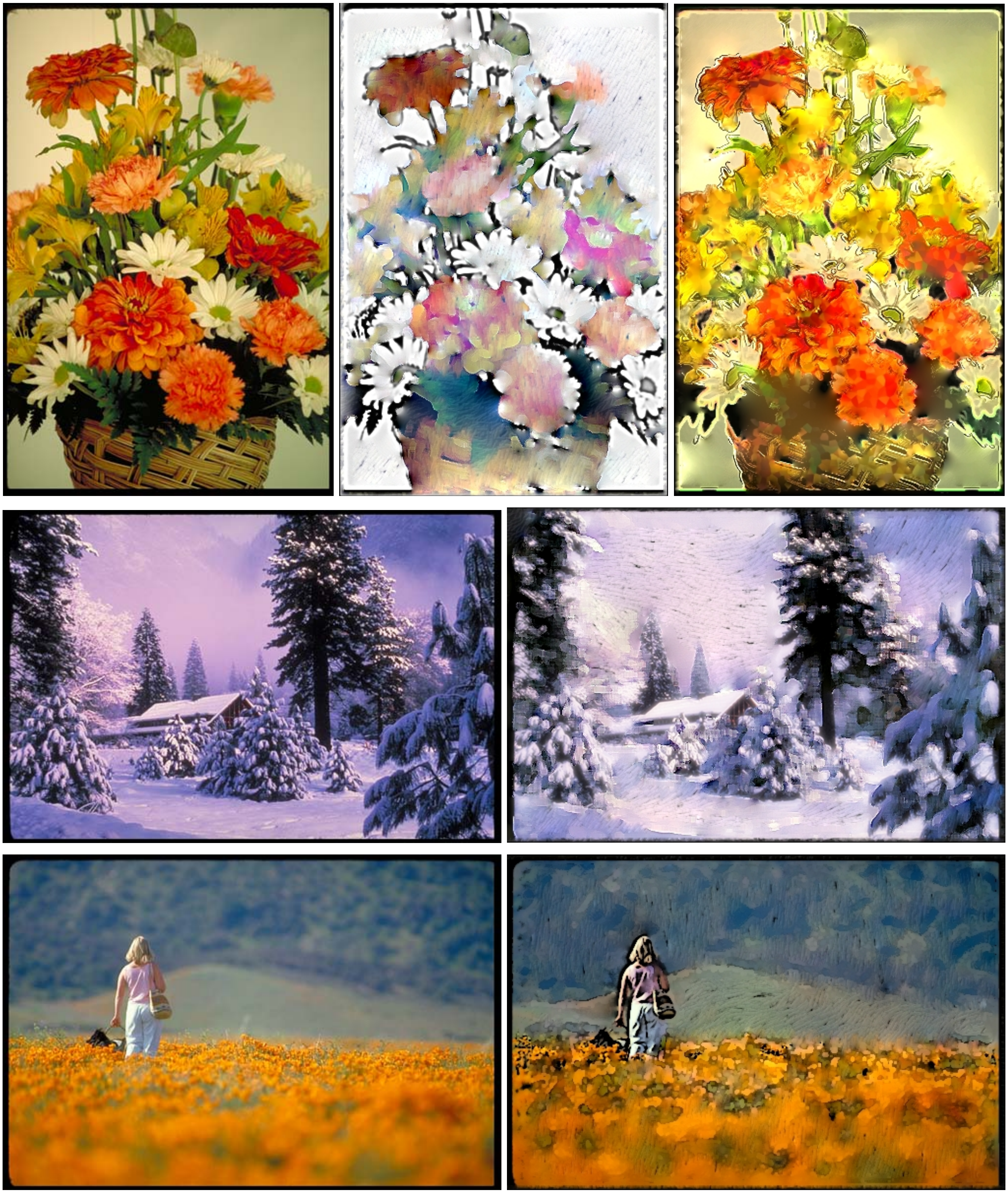

Fig. 19. Flowers, 1st row middle: F_1 (impressionist), F_2 (photocopy), F_3 (paint daubs); Flowers, 1st row right: F_1 (crystallize), F_2 (ink outlines), F_3 (accented edges); Winter scene: F_1 (impressionist), F_2 (paint daubs), F_3 (soft glow); Woman in field: F_1(impressionist), F_2 (watercolor), F_3 (cartoon).

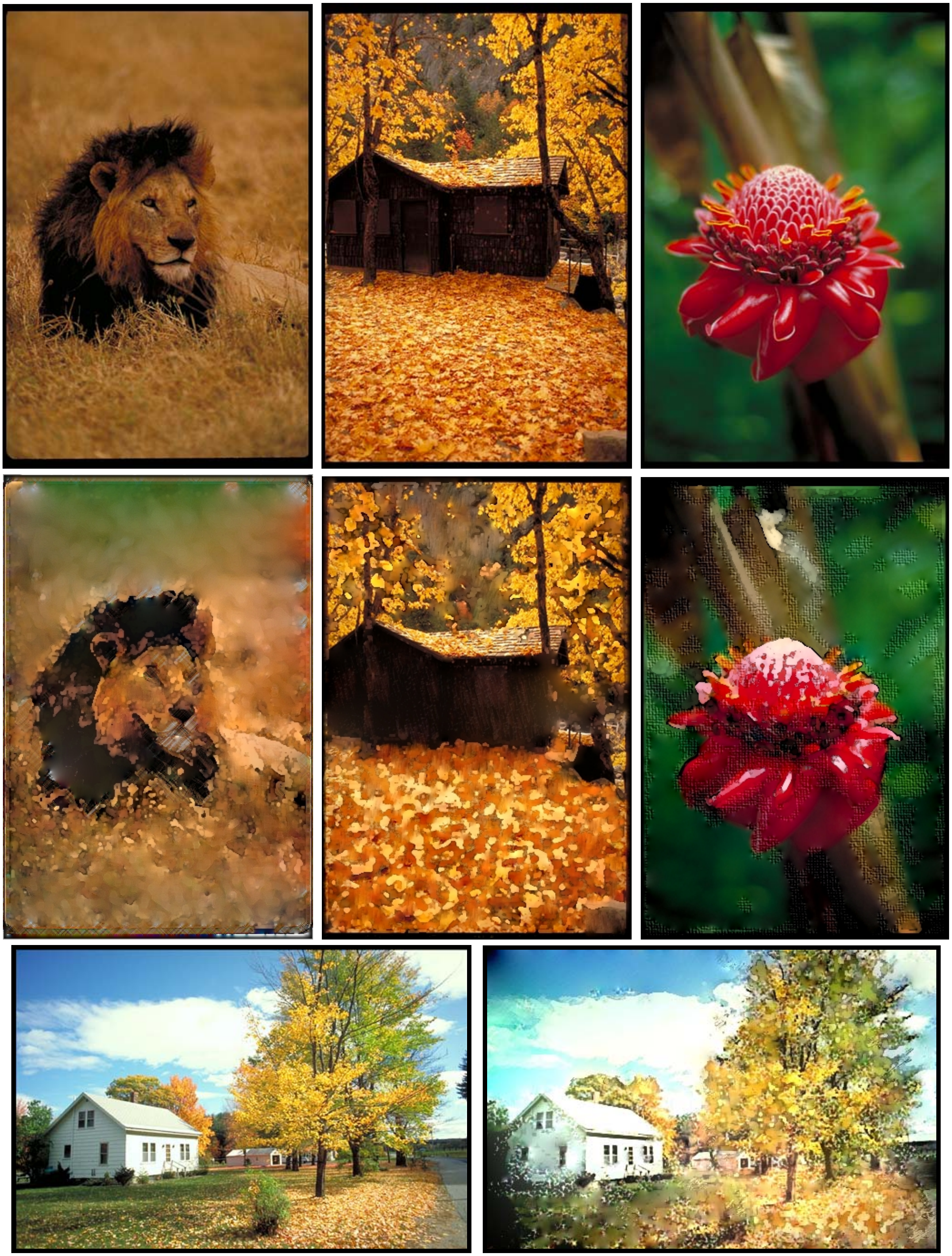

Fig. 20. Lion: F_1 (paint daubs), F_2 (watercolor), F_3 (colored pencil); Barn: F_1(impressionist), F_2 (watercolor), F_3 (none); Flower: F_1 (valuePropagate), F_2 (apply canvas), F_3 (watercolor); Farm House: F_1 (inkOutlines), F_2 (waterColor), F_3 (pointilize), F_4 (cartoon), F_5 (accentedEdges).

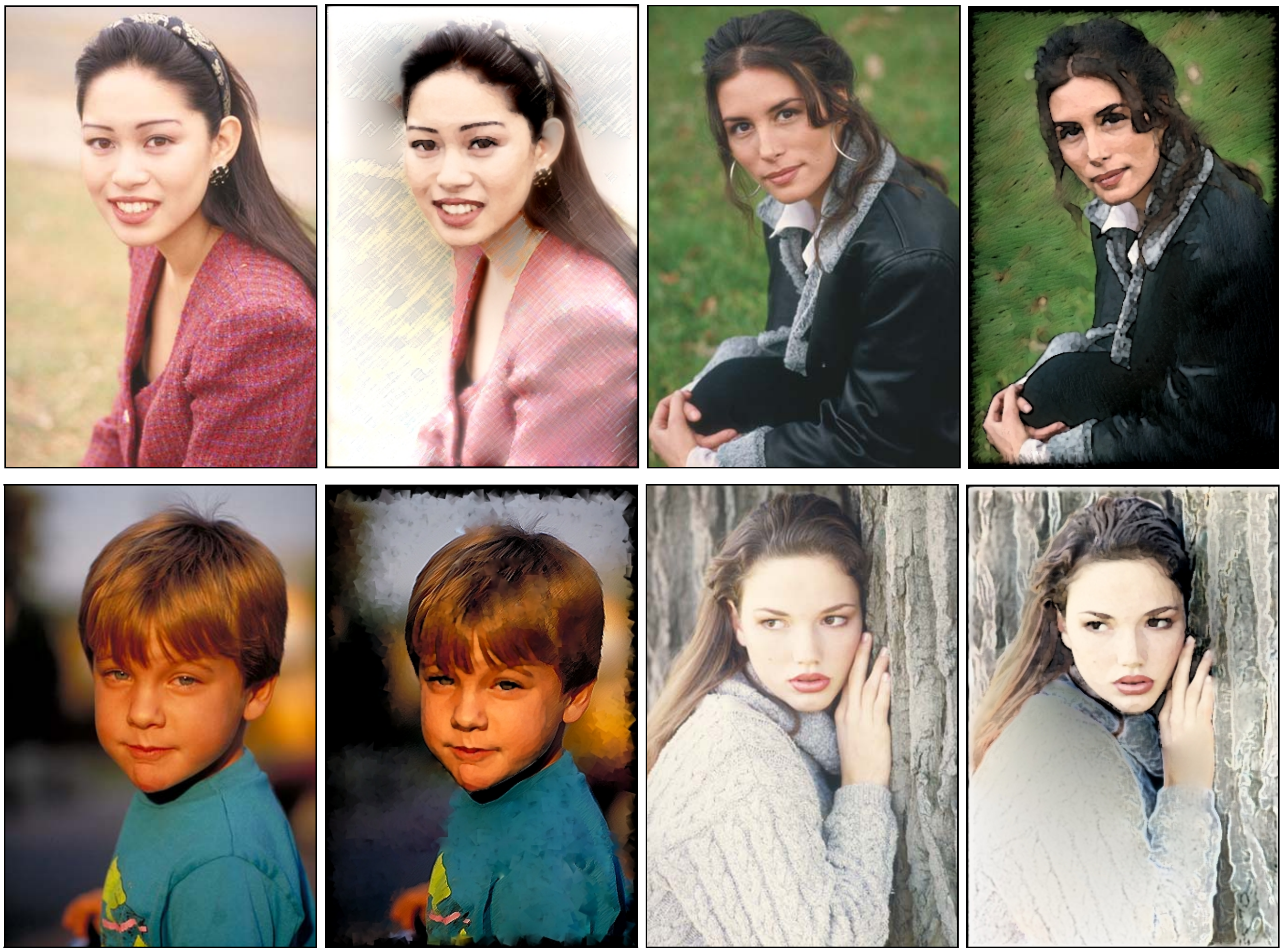

Fig. 21. Portraits with a single face region and two additional detail levels. Top-row-left: F_{face}(waterColor), F_1 (photocopy), F_2(colouredPencil); top-row-right: F_{face}(waterColor), F_1 (impressionist), F_2 (cartoon); bottom-row-left: F_{face}(waterColor), F_1(cubism), F_2(angledStrokes); bottom-row-right: F_{face}(waterColor), F_1(photocopy), F_2(accentedEdges).

An array of further results can be viewed in the supplementary results document. The average total processing time for a 512x341 image was approximately 24 seconds on a 3.6 GHz P4, including detail analysis, filtering and blending. Though we note that if alternate component filters are used then timings will vary. We also note that these timings might be optimized with GPU processing. Face detection was also computed on the order of seconds.

5 Conclusion & Future Work

In this paper we presented a method for the emulation of mixed media painting and portraiture which, until now, is a subject that has remained relatively unexplored. To accomplish this we developed a multi-stage solution that segments an input image based on image detail and face detection. This separates the image into pixel subsets which are processed independently with separate user-selected filters. The filtered image subsets are then seamlessly combined in two stages by compositing in the gradient domain.

We have demonstrated that our method produces a wide variety of results with very little input required from the user. However, our system does have limitations which could be addressed in future work. For example, our results sometimes exhibit excessive blending which might be overcome with alternate methods, such as the edge sensitive blending found in Rother *et al.* [38]; though this might be at the expense of the additional control over blending as discussed in section 3.6.

Alternate measures of region segmentation could be explored including texture and object based segmentation [52]. It may also be beneficial to incorporate edge detection or image saliency into our system [17].

Another promising path would involve the adoption of a design gallery interface [53] for exploring combinations of NPR filters. Additional directions include processing panoramic images and the automatic selection of NPR filters based on the image content. Another avenue would be to adapt mixed media NPR to video, in which case temporal coherence would become an issue. We also note that only a limited set of NPR filters have been employed in our system; many more could be explored. Lastly, a promising research direction would be to integrate 2.5D compositing of 'physical' materials into the results, such as heavy papers, felts and cardboards.

References

[1] Dye, S. 2004. *The Mixed Media Sourcebook*, Watson-Guptill Publications, New York.

[2] Braunstein, M. 1996. *Painting Various Subjects with Mixed Media*, Barrons Educational Series, New York.

[3] Ross, S. 2000. *Paint Radiant Realism in Watercolor, Ink and Colored Pencil*, North Light Books, Cincinnati.

[4] Parramon, J. 1999. *Mixed Media: Techniques and Exercises*, Lema Publications, Spain.

[5] Jelbert, W. 2002. *You Can Paint Mixed Media*, Collins, London.

[6] Gooch B. and Gooch A. 2001. *Non-photorealistic Rendering.* A.K. Peters Ltd. Publishers.

[7] Curtis, C., Anderson, S., Seims, J., Fleischer, K., and Salesin, D. H. 1997. Computer-Generated Watercolor. In Proceedings of *Computer Graphics (ACM SIGGRAPH)*, pp. 421-430.

[8] Strassmann, S. 1986. Hairy Brushes. In Proceedings of *Computer Graphics (ACM SIGGRAPH)*, pp. 225-232.

[9] Salisbury M. P., Wong M. T., Hughes J. F. and Salesin D. H. 1997. Orientable Textures for Image-Based Pen-And-Ink Illustration. In Proceedings of *Computer Graphics (ACM SIGGRAPH)*, pp. 401-406.

[10] Bangham, J.A. and Gibson, S.E. and Harvey, R.W. 2003. The Art of Scale-Space, *British Machine Vision Conference*, 1, pp. 10.

[11] Haeberli, P. 1990. Paint by Numbers: Abstract Image Representations. In Proceedings of *Computer Graphics (ACM SIGGRAPH)*, pp. 207-214.

[12] Elber, G. 1999. Interactive Line Art Rendering Of Freeform Surfaces. In *Computer Graphics Forum*, 3, 18, pp. 1-12.

[13] Winkenback, G. and Salesin, D. H. 1994. Computer-Generated Pen-And-Ink Illustration.*Computer Graphics (ACM SIGGRAPH)*, pp.91-100.

[14] DeCarlo, d., and Santella, A. 2002. Stylization and Abstraction of Photographs. *ACM Transactions on Graphics*, 21, 3, 769-776.

[15] Litwinowicz, P. 1997. Processing Images and Video for an Impressionist Effect. In Proceedings in *Computer Graphics (ACM SIGGRAPH)*, pp. 407-414.

[16] Hertzmann, A. 1998. Painterly Rendering with Curved Brush Strokes of Multiple Sizes. In Proceedings of *Computer Graphics (ACM SIGGRAPH)*, pp. 453-460.

[17] Collomosse J. P., and Hall., P. M. August 2006. Salience-adaptive Painterly Rendering using Genetic Search. *Intl. Journal on Artificial Intelligence Tools (IJAIT)*, 15(4) pp.551-576.

[18] Raskar, R., Tan, K., Feris, R., Yu, J., and Turk, M. 2004. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. *ACM Trans. Graph.* 23, 3 (Aug. 2004), pp. 679-688.

[19] Collomosse, J. P., Rowntree, D. and Hall, P. M. 2005. Stroke Surfaces: Temporally Coherent Non-photorealistic Animations from Video. *IEEE Trans. on Vis. and Computer Graphics*, 11(5), pp.540-549.

[20] Raskar, R., Ilie, A., And Yu, J. 2004. Image Fusion for Context Enhancement and Video Surrealism. In *NPAR 2004: Third International Symposium on Non-Photorealistic Rendering*, pp. 85-152.

[21] Gooch, A., Gooch, B., Shirley, P. and Cohen, E. 1998. A NonPhotorealistic Lighting Model for Automatic Technical Illustration. In *Proceedings of SIGGRAPH'98*, pp.447-452.

[22] Tietjen, C., Isenberg, T. and Preim, B. 2005. Combining Silhouettes, Shading, and Volume Rendering for Surgery Education and Planning. In *Proceedings of the Eurographics Symp. on Vis.,* pp. 303-310.

[23] Halper, N., Isenberg, T., Ritter, F., Freudenberg, B., Meruvia, O., Schlechtweg, S. and Strothotte, T. OpenNPAR: A System for Developing, Programming and Designing Non-Photorealistic Animation and Rendering. In *Proceedings of Pacific Graphics 2003*, pp. 424-428.

[24] Pérez, P., Gangnet, M., and Blake, A. 2003. Poisson image editing. *ACM Transactions on Graphics,* 22, 3, pp. 313-318.

[25] Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, A., Colburn, A., Curless, B., Salesin, D., and Cohen, M. 2004. Interactive Digital Photomontage. *ACM Transactions on Graphics*, 23, 3, pp. 294-302.

[26] Viola, P. and Jones, M. J. 2004. Robust Real-Time Face Detection. *Int. J. Comput. Vision,* 57:2, pp.137-154.

[27] Sung, K. and Poggio, T. 1998. Example-Based Learning for View-Based Human Face Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 20:1, 39-51.

[28] Rowley, H. A., Baluja, S., and Kanade, T. 1998. Neural Network-Based Face Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 20:1, 23-38.

[29] Williams, L. 1990. Performance Driven Facial Animation. *Computer Graphics (ACM SIGGRAPH' 90)*, 24(4):235-242.

[30] Ruttkay, Z. and Noot, H. 2000. Animated chartoon faces. In *Proceedigns of NPAR*, pp. 91–100.

[31] Setlur, V. and Gooch, B. 2004. Is that a smile?: gaze dependent facial expressions. In *NPAR 2004: Third International Symposium on Non-Photorealistic Rendering*, pp.79-151.

[32] Gooch, B., Reinhard, E., and Gooch, A. 2004. Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans. On Graphics,* 23:1, pp. 27-44.

[33] Chen, H., Liu, Z., Rose, C., Xu, Y., Shum, H., and Salesin, D. 2004. Example-based composite sketching of human portraits. In *NPAR 2004: Third International Symposium on Non-Photorealistic Rendering*, pp. 95-153.

[34] Ostromoukhov, V. 1999. Digital facial engraving. In Proceedings of *Computer Graphics (ACM SIGGRAPH)*, pages 417–424.

[35] Brennan, S. 1985. Caricature Generator: The dynamic exaggeration of faces by computer. *Leonardo*, 18:3, pp. 170-178.

[36] Blanz, V. and T. Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. 1999. In Proceedings of *Computer Graphics (ACM SIGGRAPH)*, pp. 187-194.

[37] Luft, T. and Deussen, O. 2004. Non-Photorealistic Real-Time Rendering of Characteristic Faces. In *Proceedings of Pacific Graphics*, pp. 339-347.

[38] Rother, C., Bordeaux, L., Hamadi, Y., and Blake, A. 2006. AutoCollage. *ACM Trans. Graph.* 25, 3 (Jul. 2006), pp. 847-852.

[39] Burt P. J. and Adelson E. H. 1983. The Laplacian Pyramid as a Compact Image Code. *IEEE transactions on communications*, 31(4), pp. 532-540.

[40] Redner, R. A. and Walker, H. F. 1984. Mixture Densities, Maximum Likelihood and the EM Algorithm. *SIAM Review*, 26, pp.195-239.

[41] Dempster, A., Laird, N., and Rubin, D. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal Royal Statistics Society*, 39:1, pp. 1-21.

[42] Ruderman, D., Cronin, T. and Chiao, C. 1998. Statistics of Cone Responses to Natural Images: Implications for Visual Coding. J. Optical Soc. of America, 15:8, pp. 2036-2045.

[43] Freund, Y. and Schapire, R. E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* 55:1, pp. 119-139.

[44] Viola P. and Jones, M. 2004. Robust Real-Time Face Detection. International Journal of Computer Vision 57:2, pp. 137-154.

[45] Schapire, R.E. and Singer, Y. 1999. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:3, pp. 297-336.

[46] Friedman, J., Hastie, T., and Tibshirani, R. 2000. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 38:2, pp. 337–374.

[47] Vezhnevets, A. and Vezhnevets, V. 2005. Modest AdaBoost - Teaching AdaBoost to Generalize Better, Graphicon '05.

[48] Fleck, M., Forsyth, D., and Bregler, C. 1996. Finding Naked People. *European Conf. on Computer Vision*, 2, pp. 592-602.

[49] Yang, M. and Ahuja, N. 1999. Gaussian mixture model for human skin color and its application in image and video databases. In *Proc. of the SPIE: Storage and Retrieval for Image and Video Databases VII*, 3656, pp 458-466.

[50] Rissanen, J. 1978. Modelling by shortest data description. *Automatica*, 14, pp. 465-471.

[51] Rother C., Kolmogorov, V., and Blake, A. GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts. *ACM Trans. Graph.* 23, 3 (Aug. 2004), pp. 309-314.

[52] Kolliopoulos, A., Wang, J. M. and Hertzmann, A. 2006. Segmentation-Based 3D Artistic Rendering. Eurographics Symposium on Rendering, pp. 361-370.

[53] Marks, J., Andalman, B., Beardsley, P. A., Freeman, W., Gibson, S., Hodgins, J., Kang, T., Mirtich, B., Pfister, H., Ruml, W., Ryall, K., Seims, J., and Shieber, S. 1997. Design Galleries: A general approach to setting parameters for computer graphics and animation. In *Proceedings of SIGGRAPH '97*, pp. 389-400.

Stephen Brooks is an assistant professor of computer science at Dalhousie University. He received his BSc from Brock University, his MSc from the University of British Columbia, and his PhD in computer science from Cambridge University in 2004. His research interests include image editing, non-photorealistic rendering, human motion editing and 3D GIS. Stephen also has a background in traditional drawing, with conté crayon, ink and pencil.