Example-Based Composite Sketching of Human Portraits

Hong Chen[1,2], Ziqiang Liu[1,2], Chuck Rose[3], Yingqing Xu[1], Heung-Yeung Shum[1], David Salesin[4,5]

[1] Microsoft Research, Asia
[2] University of California , Los Angeles
[3] Microsoft Corporation
[4] University of Washington
[5] Microsoft Research

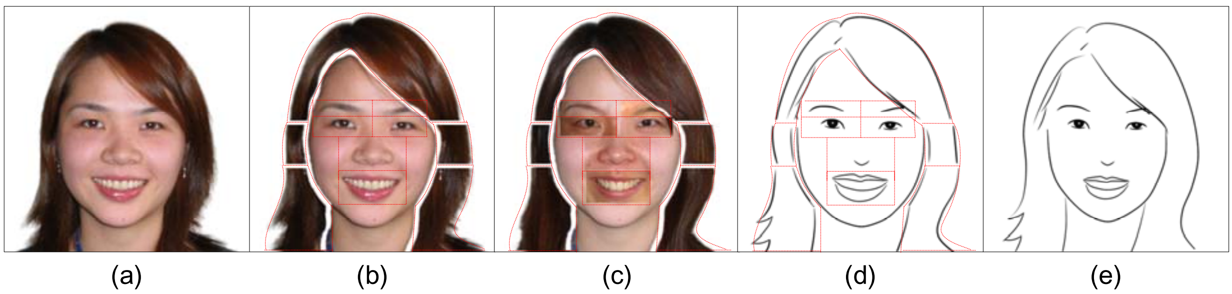

Figure 1: (a) Input image; (b) Image decomposed into components; (c) Best match for each component found from training examples; (d) Corresponding drawings of components in (c); (e) Composite drawing of separate parts as the final drawing.

Abstract

Creating a portrait in the style of a particular artistic tradition or a particular artist is a difficult problem. Elusive to codify algorithmically, the nebulous qualities which combine to form artwork are often well captured using example-based approaches. These methods place the artist in the process, often during system training, in the hope that their talents may be tapped.

Example based methods do not make this problem easy, however. Examples are precious, so training sets are small, reducing the number of techniques which may be employed. We propose a system which combines two separate but similar subsystems, one for the face and another for the hair, each of which employs a global and a local model. Facial exaggeration to achieve the desired stylistic look is handled during the global face phase. Each subsystem uses a divide-and-conquer approach, but while the face subsystem decomposes into separable subproblems for the eyes, mouth, nose, etc., the hair needs to be subdivided in a relatively arbitrary way, making the hair subproblem decomposition an important step which must be handled carefully with a structured model and a detailed model.

Keywords: Non Photorealistic Rendering, Computer Vision

1 Introduction

We describe in this paper an interactive computer system for generating human portrait sketches. Our system takes a human face image as input and outputs a sketch that exhibits the drawing style of a set of training examples provided by an artist. Our artist created the training set in the style of Japanese cartooning, or "manga". Our training data has two prominent characteristics. First, each example sketch is a highly abstract representation of the original source image, using realistic as well as exaggerated features to achieve an evocative likeness. Second, the training set contains a limited number of examples, as is often the case in example-based art applications. From this limited set, we can construct sketches for any image that satisfies certain input requirements.

Our system tackles this problem with a learning based rendering approach. Although there have been several successful similar efforts, discovering the relation between the source portrait and the corresponding sketch is a problem worth continued study. The Image Analogy [11] technique synthesizes a new "analogous" image B' that relates to an input image B in "the same way" as the example image A' relates to A. This technique, while good at mimicking the local relationships from image pair (A',A) to (B',B), lacks the power to capture the high level structural information present in our data. Another system, Example Based Sketch Generation [5], assumed a Markov Random Field (MRF) property in order to use non-parametric sampling of sketch point. It does not address face exaggeration and hair rendering which is handled in our system. Where our work improves upon the existing body in this area is this: in addition to making use of local information, we use the inherent structure in the data for a global synthesis step.

We propose a composite sketching approach for our system. The basic idea is to first decompose the data into components that are structurally related to each other, such as the eyes or mouth. After these have been independently processed, these components are carefully recomposed to obtain the final result. These two steps for both face and hair form the core of our system. Generating evocative sketches of hair is one of our primary results. The principal advantage of our component-based approach is its capacity to capture large-scale correlation within the components and its ability to create an overall picture in the style intended by the artist. This can be seen in Figure 1.

2 Related work

NPR and digital arts. Many non-photorealistic rendering (NPR) techniques have been proposed to generate digital artwork. Systems have been created to emulate watercolor and impressionism. More relevant to our work, however, are the NPR results of pen-and-ink [8; 19; 20; 22; 23] and technical illustration [17; 18]. NPR techniques have also been used to depict facial images with an artistic style. Examples include digital facial engraving [15] and caricature generation [3]. However, most of these are concerned with emulating traditional artist tools to assist users in drawing pictures with a certain style. There are rare attempts to generate digital paintings by learning from artists.

Modeling and rendering hairs. Hair is an integral part of a person's appearance. A portrait does not look natural without a realistic-looking hair style. To date, hair rendering remains one of the most difficult graphics problems, although much progress has been made on photo-realistic hair rendering [12]. Most of the work in this area is based on 3D hair model editing and rendering. Recent work by Grabli et al. [10] attempts to automatically reconstruct a hair model from photographs. Little work has been done on non-photo realistic hair rendering from an input image, which is the focus of this paper, especially in an stylized way.

Example-based learning and synthesis. Recently, a number of example-based approaches have been proposed for texture synthesis and image synthesis including image analogies by Hertzmann et al. [11], face hallucination by Baker and Kanade [2], and learning using low-level vision [9]. The basic idea is to analyze the statistical relationship between the input and output images, and model the details of the artist's style with the learned statistical model rather than with hand-crafted rules. Indeed, it is natural to specify artistic styles by showing a set of examples. Chen et al. [5], for instance, developed an example-based facial sketch generating system. Using inhomogeneous non-parametric sampling, they were able to capture the statistical likelihood between the sketch and the original image, which allowed them to fit a flexible template to generate the sketch. However, this method is limited to generating sketches with stiff lines. Chen et al. [4] recently improved their system by combining features in both sources and drawings.

3 System Framework

Our goal when we designed our algorithms was to create a system that could leverage the artist's skill with a high degree of automation after an initial training phase. Our artist created a training set designed to span the gamut of east Asian female faces. Based in part on our artist's knowledge, we divided our portrait system into a face subsystem and a hair subsystem.

The face subsystem divides the problem into meaningful components, by segmenting the problem into subproblems for each of the natural facial features, i.e. eyes, mouth, hair. A component-level model handles the overall arrangement, and a local model adjusts this initial result. The hair subsystem, on the other hand, segments the problem in a more-or-less arbitrary way, based on an insight of our artist. These subproblems are tackled independently, but care must then be taken when reassembling the hair so as to create a uniform whole. The face subsystem is covered in the next section, and the hair subsystem in Section 5.

4 Composing a Face

Suppose $\{(I'_i, I_i), i = 1:n\}$ is the training set, where each I_i is a color face image and I'_i the corresponding sketch drawn by the artist. Our objective is to construct a model

$$p(I'|I, (I'_i, I_i), i = 1:n) \tag{1}$$

to take an input image I and generate a sketch I' that matches the style of the training examples.

We split the model into two layers, global and local. The global layer aims to capture how the artist places each face element in the sketch image. The local layer tries to mimic how the artist draws each independent element locally. More formally, we have assumed

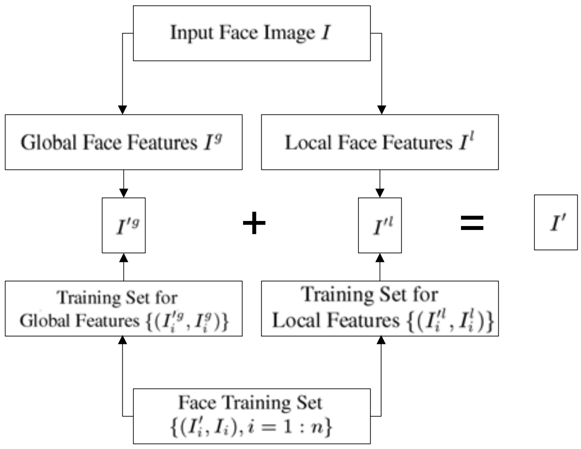

Figure 2: The framework of face subsystem.

the following

$$p(I'|I, (I'_i, I_i)) = p(I'^l|I^l, (I'^l_i, I^l_i))p(I'^g|I^g, (I'^g_i, I^g_i)), \tag{2}$$

which indicates that the global and local styles of a drawing are independent and their models could be constructed separately.

The overall steps of the face subsystem are shown in Figure 2. In the training phase, we decompose the training example into a global set $\{I'^g_i, I^g_i\}$ and a local set $\{I'^l_i, I^g_i\}$. In the synthesis phase, each input image is first split into a global feature vector I^g and a local feature vector I^l. Then we explore the global set $\{I'^g_i, I^g_i\}$ to find for I^g a good match I'^g in the sketch space. In the same way, the match for I^l can be found as I'^l. Finally, I'^g and I'^l are recomposed to form the final result.

4.1 Drawing the facial component with the local model

In practice, a human face is decomposed semantically into 6 local components, one for each of the major facial elements, of 4 types. They are left & right eyebrows, left & right eyes, a nose, and a mouth. Each type of feature is further divided into several prototypes based on their appearance in the training data.

As shown in Figure 4, the *eyebrow* component has two prototypes which are classified as thick and thin. The *eye* component has 11 prototypes which could be roughly clustered into 2 classes, those with or without a contour above the eye and below the eyebrow. The *nose* component has 3 prototypes and the *mouth* component has 4.

For each new component, we extract the accurate shape and associated texture information using a refined Active Shape Model (ASM) [7]. Then we determine to which prototype the component belongs and its associated warping parameters. We build a different classifier for each type of component and use these to cluster the input components into the appropriate prototype. k-Nearest Neighbor (kNN) interpolation is then used within the prototype set to calculate warping parameters. With prototype information and warping parameters, we are able to draw the face element locally.

4.2 Composing the face using the global model

Captured implicitly in the global model is the style which the artist uses to arrange each face element on a canvas. Most artists employ a semi-regular formula for drawing facial caricatures. They use a standard face as a frame of reference for determining how to exaggerate a subject's features. The two critical guides are the relationship of elements to others of their own kind and the relationship of elements to their surrounding and adjacent elements.

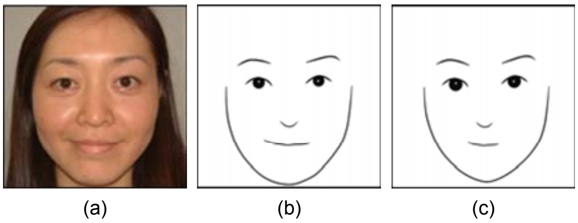

Figure 3: The effect of the local and global model. (a) The input image; (b) Sketch created with local model; (c) Sketch after global model incorporated.

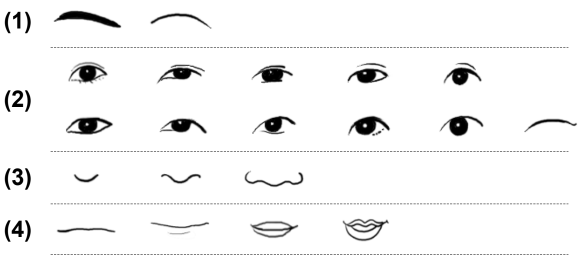

Figure 4: The prototypes extracted from the training set.

For the representation of I^g, we carefully chose 14 features from a pool of approximately 30 recommended facial features in a caricature drawing textbook [16]. They are

$$\begin{matrix} w_1/w & w_2/w & w_3/w & w_4/w & w_5/w & w_6/w & w_7/w \\ h_1/h & h_2/h & h_3/h & e_1 & e_2 & e_3 & e_4 \end{matrix} . \quad (3)$$

These relations describe the proportion of the face devoted to a particular facial feature. w_4/w, for instance, relates the width of the head to the width of the mouth. By not tying these relations to fixed values, the model can adjust the size of the features as the overall size of the head is changed. For any input face image I, we first use an Active Appearance Model (AAM) [6] to determine the 87 control points. We then use these control points to generate I^g. To determine the placement of these face elements on the cartoon canvas, each element needs five parameters $\{(t_x,t_y),(s_x,s_y),\theta\}$. (t_x,t_y) represents the translation of the element in the x and y directions respectively. (s_x,s_y) are the scaling parameters and θ is the relative rotation angle. Additionally, the face contour needs the warp parameter c_w. Together, these constitute I'^g. As shown in Figure 3, while each of the facial features drawn using the local model are correct, their overall composition is lacking. The global model improves the sketch, making a more vivid overall face.

Learning the relation between I'^g and I^g from a set of examples is non-trivial. Instead of simple linear mapping, we use k-NN interpolation to reproduce this non-linear mapping. We also make use of heuristic methods adopted by the artist. A previous system by Liang et al. [14] used partial least squares to learn facial features automatically. Since the number of examples is usually very limited and hand-annotating each example is a relatively minor additional cost over their initial creation, we believe our method is more appropriate and robust.

5 Composing hair

Hair cannot be handled in the same way as the face. This is due to a number of reasons. First, hair has many styles and is not structured in the same regular way that faces are, so building a model is not straightforward. The hair is in many ways a single unit, often rendered using long strokes, making meaningful decomposition challenging. Even if we decompose the hair into regions, recomposition remains a difficult step. Finally, there is no clear correspondence between regions of two different hairstyles. Lacking a correspondence, we cannot use blending techniques.

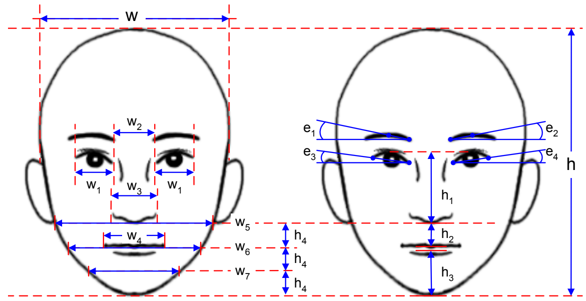

Figure 5: 14 features defined for a global model.

Due to these factors, we synthesize the hair independently from the face, employing a different mechanism. The overall flow of our hair system is shown in Figure 7. First the image is dissected into 5 regions we call the structural components. Our subject's structural components are matched against the database and the best match is selected for each. The n-best matches can be chosen to create a range of pictures for the user. These matched components are warped and assembled into an overall model for the hair. To this, details are added based upon the subject image. This process is detailed in the remainder of the section.

5.1 Hair composite model

Two key aspects make it possible to render the hair. Critically, the global hair structure or impression is more important than the details, especially for a highly-stylized look like manga. Attention to basic detail is not necessary since a person's hair details are rarely static. Wind, rain, rushed morning preparations, or severe sleep deprivation brought on by Siggraph deadlines, can all affect the details of one's hair. Figure 6 shows the three best results for a single subject's hair. All exhibit the correct shape, and picking between them can be at the behest of the user or chosen automatically by the system.

As suggested by our artist, we coarsely segment the hair into five segments, as shown in Figure 7. Each is in a fixed position and may divide long strokes that are fixed later on. We chose these segments because each indicates important global information about the hair, so we name these our "structural components". In addition to the structural (global) model, we also use a detail model. Artists often use detail to add uniqueness and expression to a portrait. These details confuse the global model, but we take them into account with a detail model, the final phase in our hair sketch synthesis.

5.2 Extracting the image features for the hair

When an input image is presented to the system we first perform an image processing step. We use this step to determine the image features of the hair which we can then match against the database. The two techniques we employ are an estimated *alpha mask* and hair strand *orientation fields*, as shown in Figure 8.

First, an alpha mask is calculated to separate the hair region from background and face. A pixel used for hair can often be blended with those used for the background and face, which is why an alpha mask is needed, rather than a simple bit mask. Knockout [1] is used to generate the mask, as shown in Figure 8(a).

Hair orientation is a very useful feature for determining the placement of strokes. We begin by using steerable filters to calculate the orientation of each pixel in the hair as demarcated by the

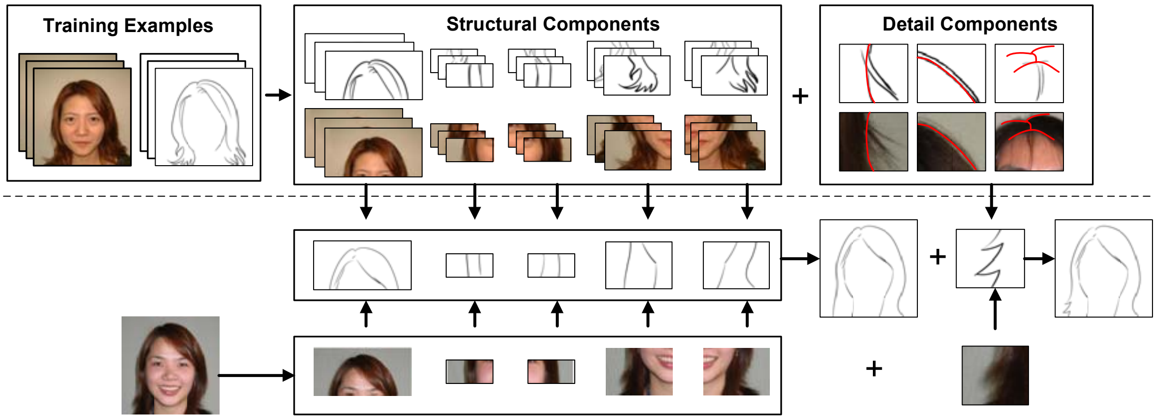

Figure 7: Hair System Flow

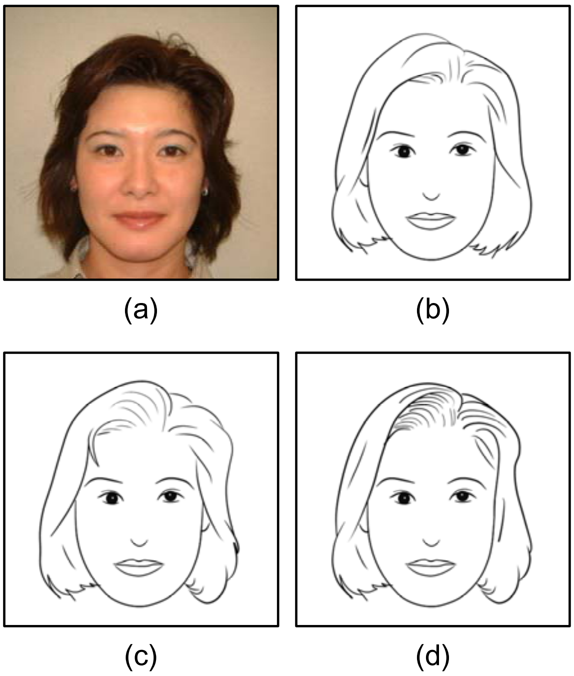

Figure 6: (a) An input image; (b)(c)(d)Three results for the input image. They all have the correct structural information.

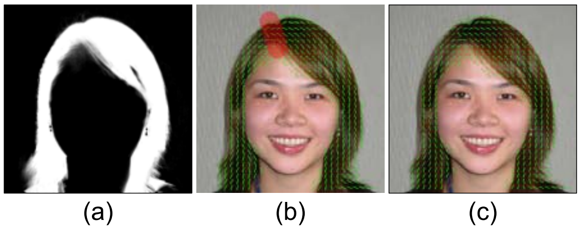

Figure 8: For the input image from Figure 7: (a) Estimated alpha mask; (b) User defined hair growing region on a smoothed edge field by red brushes; (c) Estimated hair growing field.

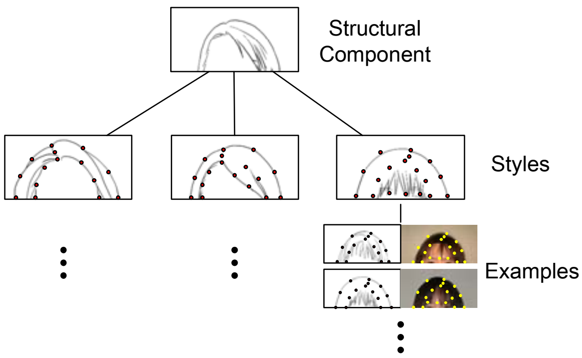

Figure 9: Structural Components of hair.

alpha mask. Weighted smoothness is then used to propagate orientation information from high strength regions to regions which are weak, in order to stifle noise in the original orientation field. These automatic orientation calculations alone cannot yield an accurate picture of what the hair is like. Some information, such as where the hair grows from is not easily calculated, so we require the user to annotate the input image to indicate hair growth (part). This is done with a simple brush stroke, as shown in Figure 8(b). The growth direction from this region can be calculated as the hair tends to grow away from the growth region, the information propagated outward subject to a smoothness constraint using belief propagation [21]. The final result is shown in Figure 8(c).

5.3 Fitting structural components

For an input hair component H, we seek to find the most similar example in the training set. For effective matching, all the examples are clustered into several styles, each of which has a roughly similar look and coarse geometric correspondence. This correspondence is determined by a set of key points, which are indicated manually in the training data, as shown in Figure 9. For an input image, finding the best training example to use is divided into two steps: first we classify H into the correct style and then we find the best training example in that style to match H.

As previously mentioned, hair components of the same style share a set of corresponding key points, which we denote as the shape $S = [x_1, y_1, \ldots x_m, y_m]$, where m is the number of key points. Then, we can deform hair components with the same style to a same standard shape using a multi-level freeform deformation warping algorithm [13]. The mean shape of training examples is chosen as the standard shape for each style.

After deforming hair to the same shape, we choose the hair orientation vector and alpha value of each pixel as the appear-

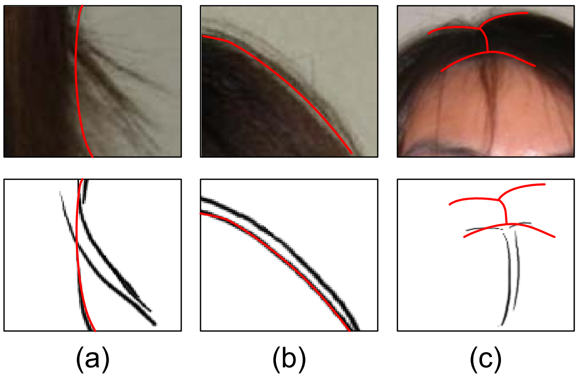

Figure 10: Detail components of the hair are divided into boundary details and bang details.

ance features of the hair. We denote the hair orientation vector as $G = [gx_1, gy_1, gx_2, gy_2, ...gx_n, gy_n]$ and the alpha value as $\alpha = [\alpha_1, \alpha_2, ...\alpha_n]$, where n is the number of pixels in the component region. Because bangs and hair tails have toothed shapes and the primary characteristics for these hair features is their length and the average density along their orientation path, we use an elliptic Gaussian kernel to smooth the alpha values.

Given all this, for two hair components with the same style, we can define the distance function of the their appearances as:

$$E(H_1, H_2) = ||G_1 - G_2|| + w||\alpha_1 - \alpha_2|| \tag{4}$$

where w is a weight.

For the ith style, with the labelled key points for each example, we get the appearance features of each example and average them to get the mean appearance $\bar{H_i}$. And we can fit the key points of hair H in the ith style by minimizing $E(H, \bar{H_i})$ using an active appearance algorithm [6].

Also we can define the distance between hair H and the ith style, and determine the best style of the hair by minimizing it:

$$E_i(H) = \omega_i \cdot E(H, \bar{H_i}), \tag{5}$$

where ω_i is a constant weight for each style, such that the mean of the $E_i(H)$ for the example hair components equal to 1.

After we determine the style of the input hair H and fit the key points S, we find the best matched training example by minimizing the distance combining the shape and appearance features:

$$\min_j E(H, H_j) + \gamma||S - S_j|| \tag{6}$$

where H_j is the jth example in the particular style, γ is a weight to balance the effect of the shape distance and appearance distance.

5.4 Fitting detail components

In addition to finding the best structural components, we also need to determine the detail information present in the input image. Different kinds of detail require slightly different approaches. We define two classes of detail, "boundary" and "bang" detail, as shown in Figure 10. Boundary details are typified by wispy strands of hair parallel to the main region of hair. Bangs are strands of hair which fall away from the bulk of the hair onto the face.

Boundary details fall into three categories, as shown in Figure 11. The alpha values and orientations for these patterns are quite different, so we can use a small window to enable classification of each point, providing a way to sort the boundary details into their appropriate categories. We summarize this information for the boundary details in the training samples, estimating a Gaussian distribution for each kind of pattern.

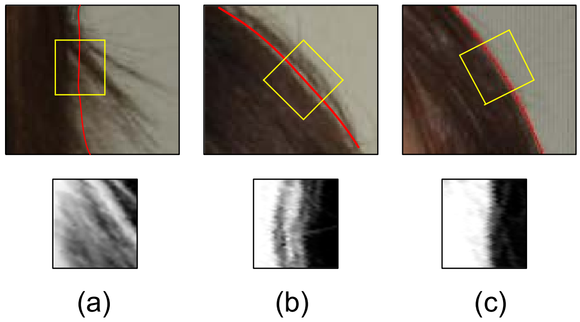

Figure 11: Three patterns of the boundary and their image features. The yellow block is the local coordinates along the boundary.

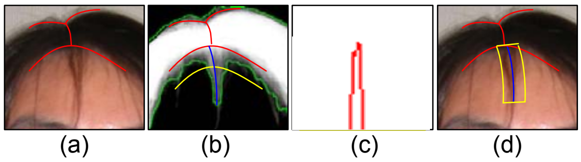

Figure 12: Fitting the bang detail components. (a) Input hair with structure spline; (b) Within the boundary of the hair in green, we trace the orientation field starting from the yellow line; (c) The length of the traced streamline along the boundary; (d) The blue line is the traced bang skeleton.

Bang detail components are used to detect the bang, such as in Figure 12(a). The first step to finding bangs uses a low threshold to segment out the bang regions in the alpha mask, indicated by the green boundary in Figure 12(b). The orientation field can be inspected in this region to find bangs, using a threshold to weed out bangs of trivial length, where this threshold is indicated by the yellow line. Next, we determine the length of the bang line, as shown in Figure 12(c). Connecting this with the smoothed bang gives us our detail bang component, as shown in Figure 12(d).

5.5 Synthesizing the hair sketch

The strokes in the training samples are all divided into two classes in the training phase: boundary strokes and streamline strokes. We define the points in the strokes crossing the boundary of a structural component to be "link points".

Face contour exaggeration requires that we adjust the lower part of the inner hair boundary using the corresponding face contour. Then the strokes of the matched structural components are warped to the target coordinates using corresponding boundaries as shown in Figure 13(a). Matching the link points in different structural components must ensure that the link points match to those in the same class (boundary to boundary, streamline to streamline), and that the matching distance is minimized. This can be handled using bipartite graph matching, as shown in Figure 13(b).

For each matched pair of link points, we adjust them to the average position and link the corresponding strokes, smoothing along the stroke and varying the width and style to make the drawing more expressive. We consider the styles of the two linked strokes, adjusting to get a good match, as shown in Figure 13(c). Unmatched streamline strokes are removed if they are too short, otherwise they are extended, tapering the end to trail off their effect on the image. Final results are shown in Figure 13(d).

Detail components are connected to strokes generated by the component match. In the training data, details are connected to

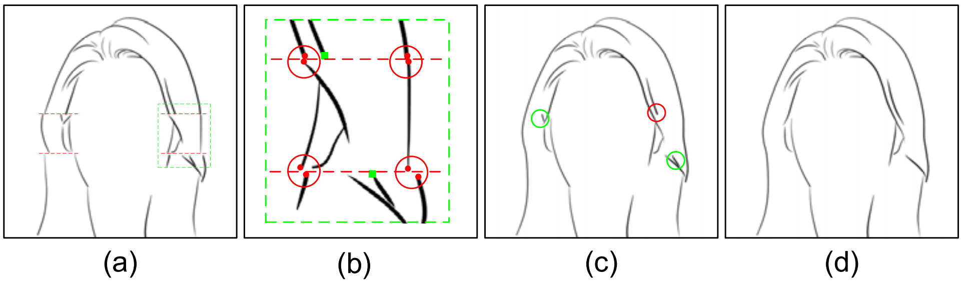

Figure 13: Composing the structural components. (a) Warp strokes of matched structural components to target image. The red line is the boundary between different components; (b) Find a match for each link point. The red round point is the link point of the boundary stroke and the green rectangle point is that of the streamline stroke. The red circle shows the match. (c) Link and smooth matched stroke. For unmatched link point of streamline, detection of too short strokes which is shown in the green circle, other link points shown in the red circle should be extended. (d) Remove short strokes and adjust the streamline to get the final result.

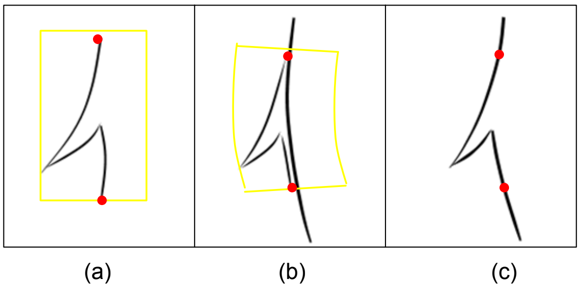

Figure 14: Add the detail strokes. (a) Strokes of the detail component with ”link point”; (b) Warp strokes from local coordinates to the target coordinates; (c) Link detail strokes to the reference stroke.

reference strokes. These link points are used in a similar way to the link points in the global phase, such as in Figure 14(a). First, the stroke of the matched detail is warped to the target coordinates from the global phase, as shown in Figure 14(b). Reference strokes are cut and the detail is inserted, smoothing any discontinuities in the linkage. This gives us our final result, as shown in Figure 14(c).

6 Examples

As shown in Figure 15, for an input image (Figure 15(a)), we decompose it into parts. For each part, we classify it into one prototype and draw it using the examples (Figure 15(b)). Then we exaggerate it using the global model and get the final result in Figure 15(c). We can see that the final result of our system is locally similar to the previous one, but the size of the mouth and the shape of the contour are notably different to make it more “manga”. In comparison, we show the sketch result without local model variation or a global model in Figure 15(d), which is similar to the result in [5]. Our result is obviously more expressive.

Users may prefer a particular prototype of the local component or the exaggeration effect. Our system allows the user to change the result interactively to find the result with the best style as in Figure 15(e-f).

For the hair of the input image in Figure 16(a), we decompose it into structural components and detail components. The result of composing structural components captures the global style of the hair as in Figure 16(b). And then we add unique detail to make it more expressive as in Figure 16(c). With the two level components model, we can obtain better results especially with a small set of examples. In comparison, we let the structural components contain all of the strokes and get the result in Figure 16(d). It has the same structural information as the previous result; however, it does not match the details of the input image absolutely.

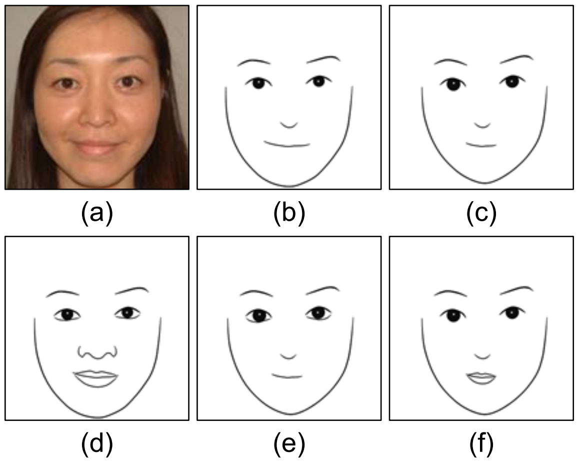

Figure 15: Results of face part.(a) The original face image; (b) The result of local model; (c) The result of local model plus global model; (d) Result without local model and global model. (e)(f) Results with user selected local prototypes.

Also, the user can generate the result interactively. But the goal of our system is to draw the hair from the input image. So, for each component, we show the n-best match examples, and the user need only click the mouse and select one of them. The user can also remove detail components if they prefer an abstract one. The result of different choices is shown in Figure 16(e-f).

Combining the face and the hair, we need to remove the boundary strokes of the hair that overlap the face contour. In Figure 18, we show some results generated by our system. Neck, shoulder, and clothing are chosen from a set of templates supplied by our artist. 52 separate training examples were used for this system.

7 Discussion

We believe that adapting a global/local hybrid was an effective approach for generating face portrait sketches. The face, being more structured than the hair, was a clear fit for this approach, but the hair worked exceptionally well once a recomposition fixing stage was added. Systems like this have many applications, such as creating virtual personas for cartoon style online games or chat environ-

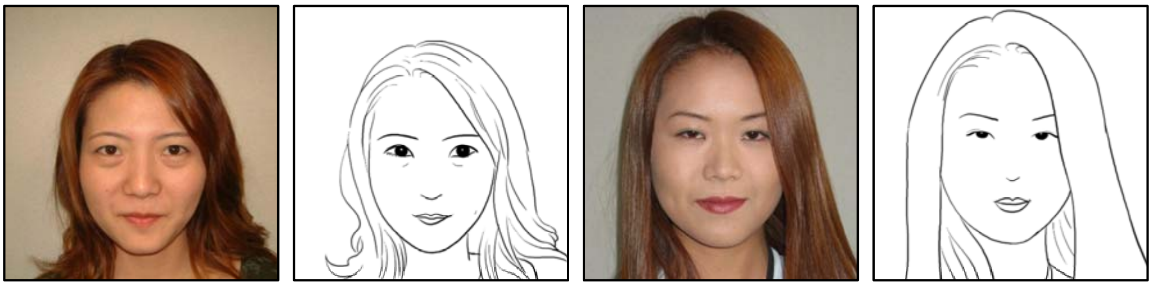

Figure 17: Two examples of the Manga style training data in our system. We use 52 in the working system.

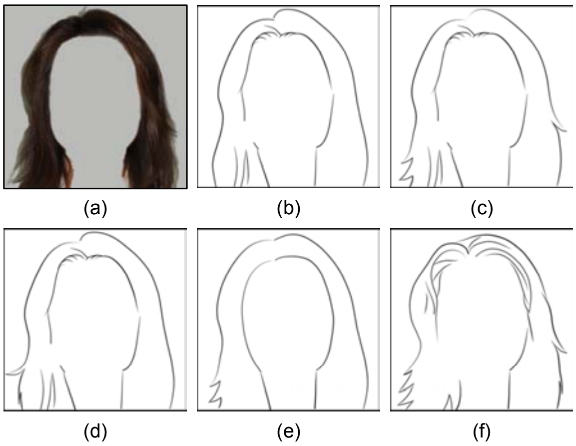

Figure 16: Compare the effect of adding detail strokes. (a) The original image; (b) Result of composing structural components; (c) Composing with detail components; (d) Result of one level model where all of the details strokes remain in structural components. (e)(f) Composed hair by user selected components.

ments. Obvious extensions would be to use this system to automatically create sketched portraits for places where a sketch is preferred over a photograph, such as on the cover of *The Wall Street Journal* provided that an appropriately styled training set is used. Allowing people to explore a change of look is another application.

Our system has some limitations we hope to address in future work. We obviously want to add to our training set so we can encompass faces of many racial backgrounds. Our use of white background portraits was a very good fit for the relatively pale skin of east Asian females and may work well for Caucasian races, but may be a poor fit for people whose ancestors are from southern Asia, Africa, and Polynesia, for example. Vastly different hair styles, such as dread locks or a mohawk may not fit well into our framework at present. Clearly, we also want to render male images as well, which presents the additional challenge of facial hair, blurring the line between the facial and hair subsystems. A third subsystem to handle facial hair, which is different in many ways from scalp hair, may be in order. Aging, injury, spectacles, and jewelry, are all obvious extensions.

References

[1] A.Berman, A.Dadourian, and P.Vlahos. Method for removing from an image the background surrounding a selected object. U.S. Patent 6,134,346, 2000.

[2] S. Baker and T. Kanade. Hallucinating faces. In *AFGR00*, 2000.

[3] S. Brennan. The caricature generator, 1985.

[4] H. Chen, L. Liang, Y.Q. Xu, H.Y. Shum, and N.N. Zheng. Example-based automatic portraiture. In *ACCV02*, 2002.

[5] H. Chen, Y.Q. Xu, H.Y. Shum, S.C. Zhu, and N.N. Zheng. Example-based facial sketch generation with non-parametric sampling. In *ICCV01*, pages II: 433–438, 2001.

[6] T. Cootes and C. Taylor. Statistical models of appearance for computer vision, 1999.

[7] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding: CVIU*, 61(1):38–59, January 1995.

[8] Oliver Deussen and Thomas Strothotte. Computer-generated pen-and-ink illustration of trees. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, Annual Conference Series, pages 13–18. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[9] William T. Freeman, Egon C. Pasztor, and Owen T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, 2000.

[10] Stephane Grabli, François Sillion, Stephen R. Marschner, and Jerome E. Lengyel. Image-based hair capture by inverse lighting. In *Proc. Graphics Interface*, pages 51–58, May 2002.

[11] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In Eugene Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, Annual Conference Series, pages 327–340. ACM Press / ACM SIGGRAPH, 2001.

[12] Tae-Yong kim and Ulrich Neumann. Interactive multiresolution hair modeling and editing. In *Proceedings of the Conference on Computer Graphics 2002*, pages 620–629, New York, 2002. ACM Press.

[13] Seung-Yong Lee, Kyung-Yong Chwa, and Sung Yong Shin. Image metamorphosis using snakes and free-form deformations. *Computer Graphics*, 29(Annual Conference Series):439–448, 1995.

[14] Lin Liang, Hong Chen, Ying-Qing Xu, and Heung-Yeung Shum. Example-based caricature generation with exaggeration. In *Proc.*, pages 386–393, Oct 2002.

[15] Victor Ostromoukhov. Digital facial engraving. In Alyn Rockwood, editor, *Siggraph 1999, Computer graphics Proceedings*, Annual Conference Series, pages 417–424. ACM Siggraph, Los Angeles, 1999.

[16] Lenn Redman. *How to Draw Caricatures*. Contemporary Books, 1984.

[17] C. ROSSL and L. KOBBELT. Line-art rendering of 3d models, 2000.

[18] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-D shapes. volume 24, pages 197–206, 1990.

[19] Michael P. Salisbury, Sean E. Anderson, Ronen Barzel, and David H. Salesin. Interactive pen-and-ink illustration. *Computer Graphics*, 28(Annual Conference Series):101–108, July 1994.

[20] Michael P. Salisbury, Michael T. Wong, John F. Hughes, and David H. Salesin. Orientable textures for image-based pen-and-ink illustration. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 401–406. ACM SIGGRAPH, Addison Wesley, August 1997.

[21] Y weiss. Interpreting images by propagating bayesian beliefs. pages 908–915, 1997.

[22] Georges Winkenbach and David H. Salesin. Computer-generated pen-and-ink illustration. *Computer Graphics*, 28(Annual Conference Series):91–100, July 1994.

[23] Georges Winkenbach and David H. Salesin. Rendering parametric surfaces in pen and ink. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 469–476. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.

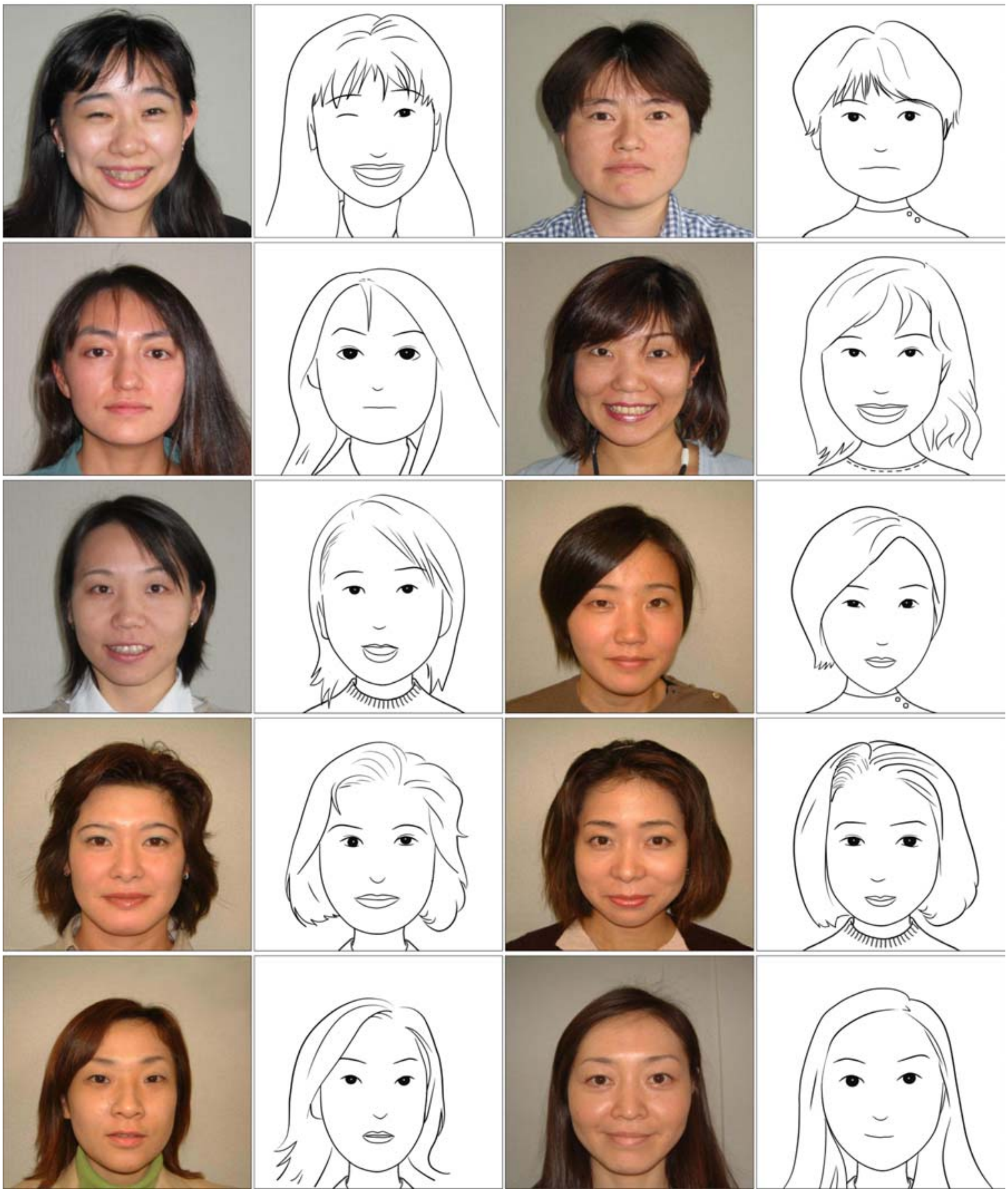

Figure 18: Manga style portraits generated by our system.