Guided Face Cartoon Synthesis

Hongliang Li, *Member, IEEE*, Guanghui Liu, and King Ngi Ngan, *Fellow, IEEE*

***Abstract*—In this paper, we propose a new method, called guided synthesis, to synthesize a face cartoon from a face photo. The guided synthesis is defined as a local linear model, which generates a cartoon image by incorporating the content of guidance images taken from the training set. Our synthesis operation is achieved based on four weight functions. The first is a photo-photo weight that aims to measure the similarity between an input photo patch and a training photo patch. The second is defined as a photo-cartoon weight, which is used to compute the likelihood by computing the similarity between a cartoon patch and an input photo patch. The third weight is defined in the synthesized photos, which is to set a smoothness constraint between neighboring synthesized patches. The final weight is designed to evaluate the similarity of a synthesized patch to an input patch based on the spatial distance. Experimental evaluation on a number of face photos demonstrates the good performance of the proposed method on the face cartoon synthesis.**

***Index Terms*—Face cartoon, guided synthesis, linear model.**

I. Introduction

A. Motivation

HUMAN face has been extensively studied for such a long time in pattern recognition and computer vision, which has been applied in many fields, such as face detection [1], [2], recognition [3]–[5], tracking [7], hallucination [8], animation [9], portrait [10], and sketch [11]–[13]. Recently, the facial non-photorealistic rendering method (e.g., face cartoon or face sketch) has drawn a great deal of attention due to its encouraging applications in face recognition [11], [13], and digital entertainment [14], [15]. For example, in the cartoon movie or TV production, creators can draw cartoon faces easily with the assistance of automatic cartoon synthesis system and focus on the storyline. In addition, people love to make personalized cartoon pictures in the digital world such as video chatting, photo album, or cinema comics [16]. Therefore, automatically cartooning through a photo is very useful for the future intelligent multimedia processing technique. However, this method highly depends on human artists and their manual skills, and it is still a challenging and difficult work to automatically create a cartoon from an input face photo by a computer.

Fig. 1. Examples of face photo-cartoon pairs. Left in (a)–(c): Photos in CUHK face database [13]. Right in (a)–(c): Corresponding cartoons drawn by an artist.

Compared with objects such as car, building, or tree, the human face is more sensitive to our eyes. On one hand, people can easily perceive the difference between two faces that share similar facial features. On the other hand, people can exactly recognize the query person from a number of cartoon pictures which hold great difference between them. Although the psychological mechanism of cartoon generation is difficult to expressed by grammar, a professional artist is able to capture the distinctive features of human faces and draw them on cartoons or sketches [13]. But it is still a challenging task to synthesize face cartoons from photos by a computer. Photo and cartoon usually exhibit different modalities in style and appearance as shown by examples in Fig. 1. During the cartoon painting, the artist tends to draw the salient facial features by using some simple and exaggerated strokes, such as sharp chin, curving eyehole, and mouth contour. Unlike the face sketch, most of the face skin parts in the cartoon will be expressed by homogenous color regions.

B. Related Works

The work of face cartoon can be traced back to the PicToon system [17], which created a personalized cartoon from an input image by using sketch generation and stroke rendering. A non-parametric sampling scheme along with a facial template fitting is used to extract the facial sketch lines from an input image. The user can add different artistic strokes to the cartoon sketch in the stroke rendering step. Using interacting snakes, a semantic face graph derived from a subset of vertices of a 3-D face model was proposed to construct cartoon faces for face matching [18]. In this work, face detection results (e.g., face, eye, and mouth locations) were used to initialize multiple snakes that represent the complete face graph and interact with each other to obtain an aligned face graph (called a cartoon face). Recently, a semi-automatic EasyToon system is designed to generate a personalized cartoon picture by replacing a cartoon face using a real face in the album [19]. This system consists of two stages, i.e., face candidate selection from thousands of faces and blending operation that pastes the selected face on the cartoon image. In addition, some works focus on the cartoon animation, such as video tooning that transformed an input video into a highly abstracted cartoon animation with a range of styles [20].

In current literature, many methods had been proposed to perform face sketch synthesis, which synthesizes a face sketch

Manuscript received March 11, 2011; revised June 14, 2011 and August 26, 2011; accepted September 02, 2011. Date of publication September 19, 2011; date of current version November 18, 2011. This work was supported in part by NSFC (No.60972109 and 61101091), in part by the Program for New Century Excellent Talents in University (NCET-08-0090), and in part by Sichuan Province Science Foundation for Youths (No. 2010JQ0003). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zhu Liu.

H. Li and G. Liu are with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, China (e-mail: hlli@ee.uestc.edu.cn; ghliu@ee.uestc.edu.cn).

K. N. Ngan is with the Department of Electronic Engineering, The Chinese University of Hong Kong, ShaTin, N. T. Hong Kong, China (e-mail: knngan@ee.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2011.2168814

1520-9210/$26.00 © 2011 IEEE

from a set of training face photo-sketch pairs [11], [21], [22]. An example-based facial sketch system was first addressed in [21], which generated a sketch from an input image by learning from example sketches. This method employed a nonparametric sampling scheme to learn statistical characteristics between an image and its sketch. Tang and Wang [11] proposed a face photo synthesis system without hair region using sketch drawings. This method separated shape and texture information in a face photo, and applied the transformation on them, respectively.

Inspired by the work of learning-based method for low-level vision problems [6], a face photo-sketch synthesis and recognition method was proposed in [13], which was based on a multiscale Markov network. Given a frontal face photo, this system first synthesized a sketch drawing. A photo was then synthesized by searching for face photos in the database based on a query sketch drawn by an artist. Recently, Zhang *et al.* propose a method to synthesize a face sketch from a face photo taken under a different lighting condition or in a different pose than the training set [15]. This method uses a multiscale Markov network to generate local sketch patches, which mainly consists of three steps, i.e., shape priors specific to facial components, new patch descriptors and metrics, and a smoothing term measuring both intensity and gradient compatibilities. In addition, a hierarchical-compositional model of human faces was presented in [12], which used a three-layer AND-OR graph to account for the face regularity and dramatic structural variabilities caused by scale transitions and state transitions. The first layer treats each face as a whole image, while the second layer refines the local facial parts jointly as a set of individual templates. Further partition of the face into zones and detail facial features (e.g., eye corners, or wrinkles) are implemented in the third layer. This model is useful for generating a cartoon facial sketch.

In this paper, we propose a guided synthesis method to generate a face cartoon image from an input face photo in front view. Unlike the existing methods, the proposed method performs cartoon synthesis based on a local linear model, which employs the guidance images taken from training photo-cartoon pairs. To achieve the robust synthesis, four cost functions are defined to compute synthesis weights. The first is called photo-photo weight that aims to compute the similarity between an input photo and a training photo. The second is the photo-cartoon weight, which is used to compute the likelihood by measuring the similarity of a cartoon to an input face photo. The third weight is computed in the synthesized photos, which is designed to impose the smoothness constraint between neighboring synthesized patches. The final weight is to evaluate the similarity of patch to an input patch based on the spatial distance. Using four weights, we can sufficiently utilize the relations among the face photo, the synthesized cartoon, and the cartoons drawn by the artist during the cartoon synthesis, which are usually ignored in the existing methods. Experimental evaluation on a number of face photos demonstrates the effectiveness of the proposed method on the face cartoon task.

This paper is organized as follows. Section II introduces our proposed guided face cartoon algorithm. Experimental results are provided in Section III to demonstrate the effectiveness of our approach. Finally, Section IV concludes the paper.

II. Proposed Method

In this section, we introduce our algorithm of guided face cartoon synthesis. The guided synthesis defined in our paper is achieved by computing a local linear transform of the guidance image pairs, which are selected from the training face database.

A. Normalization Processing

Before the face cartoon synthesis, we first perform a normalization step on the color face data. Given a face photo database, we choose the face photos that are taken in frontal view and neutral expression. For each face photo, we invited the artist to draw a cartoon picture. All the photos and cartoons are then translated and scaled to 128×96 images by adjusting the eye centers at fixed position, i.e., left eye center (31,63) and right eye center (63,63). In addition, all photo-cartoon pairs in RGB color space will be converted to the YCbCr color space so as to perform the cartoon synthesis. Here, we follow the concept in [6] to build the photo-cartoon pairs as the training set. It may be different to the field of computer vision that usually refers to the process of determining the model functions or values of parameters. In this paper, the training dataset is used to find the similar patch pairs and use them to estimate the cartoon patch to be synthesized.

B. Guided Synthesis

Let $(I, \widetilde{I})$ denote the training photo-cartoon pairs, which are given beforehand by a well-trained artist. Given a new image X, our goal is to automatically generate a synthesized image Z by a linear filtering process based on the training set. An overview of our proposed guided synthesis is shown in Fig. 2, which mainly consists of two steps. The first is the cartoon synthesis from the luminance channel, which aims to generate the facial appearance. The second step is the colorization, which is used to add the color information to the initial synthesized result.

Assume each image is divided into N overlapping patches with identical spacing. Here, we use Ω to represent the set of pairs of training photo-cartoon patches. Given a test photo patch X_p, we estimate a synthesized patch in terms of the training patches. If p and q denote patch indices, the patch p in the synthesized image Z can be expressed as

$$Z_p = \sum_{q \in \Omega} W_{qp}(X, I, \widetilde{I}) \widetilde{I}'_q + \mu_{X_p} \tag{1}$$

where μ_{X_p} denotes the mean of the input photo patch X_p. $\widetilde{I}'_q$ is the normalized cartoon patch by subtracting the mean from the cartoon patch $\widetilde{I}_q$, namely $\widetilde{I}'_q = \widetilde{I}_q - \mu_{\widetilde{I}_q}$. W_{qp} is a filter kernel defined in the input image X and the guidance image pair $(I, \widetilde{I})$, which is given as

$$W_{qp} = \frac{1}{K_p} w_1(X_p, I_q) w_2(X_p, \widetilde{I}_q) w_3(Z_{\mathcal{N}(p)}, \widetilde{I}_q) w_4(p, q) \tag{2}$$

where $\mathcal{N}(p)$ denotes the set of neighboring patches for patch p, and K_p is a normalizing term to ensure that the sum of W_{qp} is equal to one. Given a face photo patch X_p, our goal is to generate a cartoon patch Z_p from the training cartoon patch set based on their similarity weights. From (2), we can see that there are four types of weight functions that are defined to compute the kernel W. The function w_1 is photo-photo weight that sets the penalty between the input image X and the training image I based on the patches similarity. The second function w_2 is photo-cartoon weight that is used to evaluate the similarity between the image X and the cartoon image $\widetilde{I}$. The third function w_3 is defined as a smoothness weight by setting the constraint between the neighboring patches within the synthesized image Z. The final

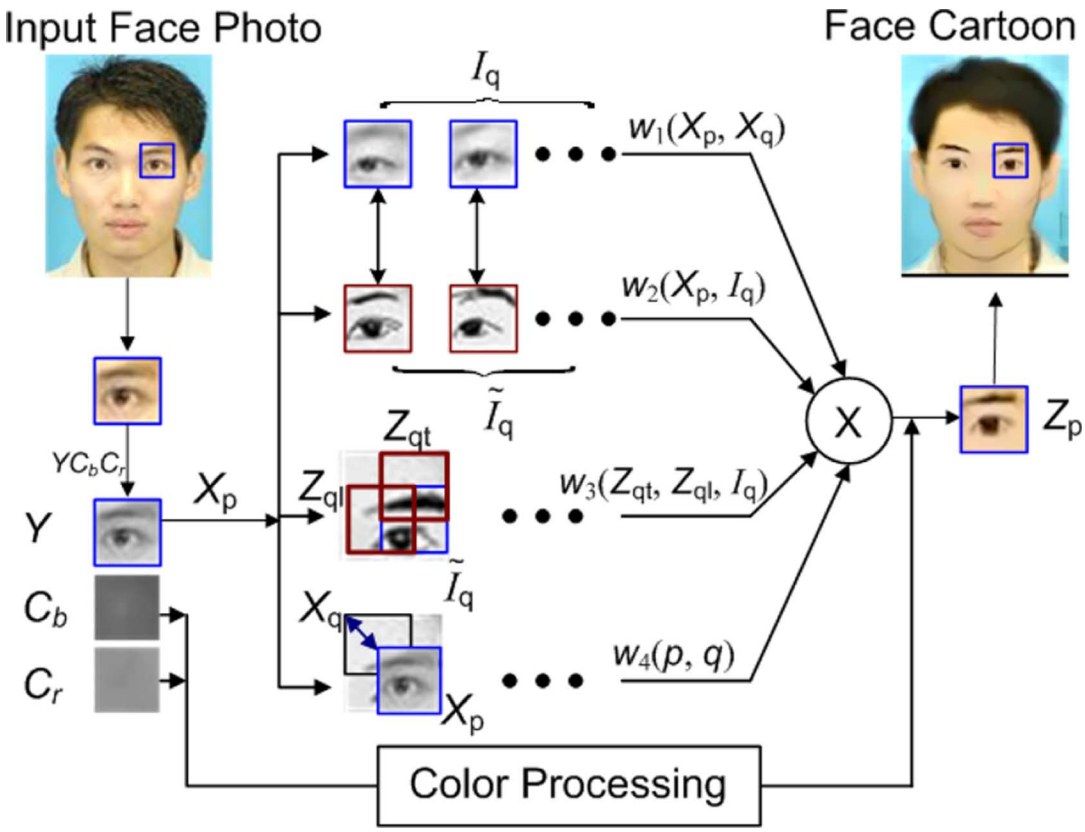

Fig. 2. Framework of our guided face cartoon synthesis.

term is called spatial weight w_4, which aims to set the penalty cost in the spatial domain based on the distance. The detailed analysis of the above weights will be described as follows.

1) Photo-Photo Weight: In order to select a good cartoon patch from training images $\widetilde{I}$, the photo-photo weight is designed to compare a photo patch I_p with an input patch X_p. The weight w_1 can be treated as the likelihood $P(I|X)$, which is used to evaluate the similarity between I_p and X_p. Generally, the similarity can be directly computed based on their distance. But, it usually does not work well due to the lighting condition variation. Thus, the normalized patches X'_p and I'_p are used to measure the distance, which is defined as

$$w_1(X_p, I_q) = \exp\left(-\frac{\|X'_p - I'_q\|^2}{\sigma_1^2}\right) \tag{3}$$

where $X'_p = X_p - \mu_{X_p}$ and $I'_q = I_q - \mu_{I_q}$, the parameter σ_1 adjusts the range (i.e., intensity) similarity. The operation $\|\cdot\|$ calculates the Euclidean distance between patches X'_p and I'_q. From (3), we can see that the larger the value of w_1 is, the more similar the two patches are.

2) Photo-Cartoon Weight: Unlike the weight w_1, the photo-cartoon weight w_2 is used to compute the likelihood $P(\widetilde{I}|X)$ by measuring the similarity between a cartoon patch $\widetilde{I}_p$ and an input photo patch X_p. Generally, it is difficult to directly compute their difference based on their visual appearances because they are generated by various styles: one from a camera and the other from the artists. Artists would like to draw cartoons with distinct outline, which causes large difference between photos and cartoons. However, people can easily recognize a person's cartoon by matching it with the face photo, which shows that there exists a certain similarity between them. The similarity between a photo and a cartoon can be measured based on gradient orientations, which has been applied to the sketch generation [15].

Here, we will compute the similarity between a photo and a cartoon image by using local region descriptors. Let $D(X_p)$ and $D(\widetilde{I}_q)$ denote the $1 \times d$ dimensional region descriptors for paths X_p and $\widetilde{I}_q$. In our work, the photo-cartoon weight w_2 can be expressed by

$$w_2(X_p, \widetilde{I}_q) = \exp\left(-\frac{\|D(X_p) - D(\widetilde{I}_q)\|^2}{\sigma_2^2}\right) \tag{4}$$

where the parameter σ_2 adjusts the descriptors similarity.

For the descriptor computation, there exist a number of methods to extract region descriptors in the current literature, such as SIFT [23], MSER [24], Salient regions [25], Harris-Affine and Hessian Affine [26], and HOG [27]. In this work, we first decompose a region into different levels, and compute the descriptor named Pyramid of Histograms of Orientation Gradients (PHOG) [28]. Since this descriptor is computed based on a region pyramid representation, it can capture the spatial distribution of edges in different scales. This feature is usually used to measure the shape/region correspondence. Fig. 3 shows an example of PHOG computation for a face photo patch. Fig. 3(a) shows a face photo, where the red square denotes a candidate photo patch. The edge map obtained by Canny edge detection is given in Fig. 3(b). The PHOG descriptor is computed from the edge patch, which is marked as a red square. Fig. 3(c) shows results at three different scales, i.e., $L = 1, 2, 3$. A HOG descriptor is computed for each grid cell

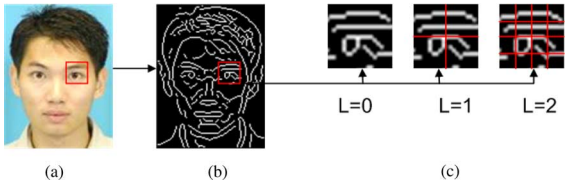

Fig. 3. Illustration of the weight w_2 computation. (a) Face photo. (b) Corresponding edge map. (c) PHOG computation.

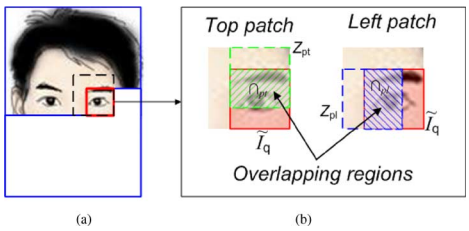

Fig. 4. Illustration of the weight w_3 computation. (a) Face cartoon synthesis in raster scan order, where the red window denotes the synthesizing patch. (b) Local view of overlapping regions between neighboring patches.

at each pyramid resolution level. The final PHOG descriptor for a patch is a concatenation of all the HOG vectors [28].

3) Cartoon Smoothness Weight: Both of weights w_1 and w_2 can be regarded to answer how much the current training patch is similar to the input photo patch. They do not consider the relationship between adjacent synthesized patches. Here, the smoothness weight w_3 is defined in the synthesized photo, which is to set the smoothness constraint between neighboring synthesized patches.

In our work, we implement the face cartoon synthesis in raster scan order. Given a synthesized patch, it is reasonable to assume that there exist two synthesized cartoon patches, namely the top patch and left patch. An example of computing the weight w_3 is shown in Fig. 4. The red window in Fig. 4(a) denotes a candidate synthesized patch, while the white region is the unsynthesized face region. The local view of the black window is shown in Fig. 4(b), which represents the relationships of the current patch $\widetilde{I}_q$ with respect to the top and left patches, respectively. The slashes in green and blue color denote the overlapping regions for the top patch and left patch, respectively, which are used to compute the intensity compatibility. Let Z_{pt} and Z_{pl} denote the top and left patches with respect to $\widetilde{I}_q$, respectively. In our work, the cartoon smoothness weight w_3 is defined as

$$w_3(Z_{pt}, Z_{pl}, \widetilde{I}_q) = \exp\left(-\frac{\|\widetilde{I}_q^{\bigcap_{pt}} - Z_{pt}^{\bigcap_{pt}}\|^2 + \|\widetilde{I}_q^{\bigcap_{pl}} - Z_{pl}^{\bigcap_{pl}}\|^2}{\sigma_3^2}\right) \quad (5)$$

where $\bigcap_{pt}$ and $\bigcap_{pl}$ denote the overlapping regions for pairs of patches $\widetilde{I}_q$-Z_{pt} and $\widetilde{I}_q$-Z_{pl}, respectively. The parameter σ_3 adjusts the smoothness constraint. It is noticed the smoothness weight w_3 is designed to measure the similarity of the overlapping patches, which achieves the smooth variation between neighboring synthesized patches. From (5), we can see that only the overlapping regions (e.g., slash area in Fig. 4) are employed

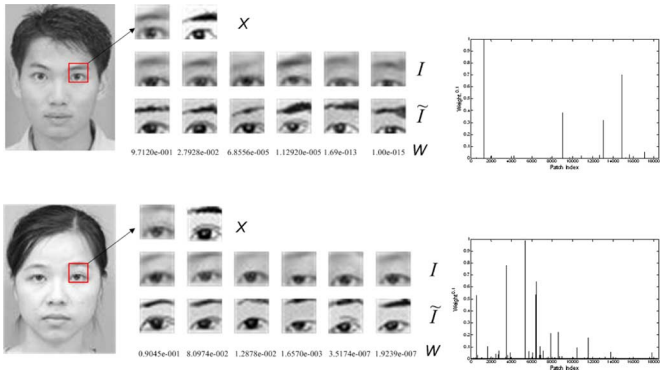

Fig. 5. Examples of the weight computation. Left: Input photo patches (marked as red window). Middle: Input photo-cartoon patches are shown in the first row, while the first six photo-cartoon patch pairs with highest weights are shown in the second and third rows, respectively. Right: Weights of all training patches. To provide clearly visual effect, the 10th root of weights are plotted.

to compute the weight w_3. Finally, we incorporate the smoothness weight w_3 into the filter kernel in (1) for weighting the cartoon patch $\widetilde{I}'_q$.

4) Spatial Weight: This weight w_4 is designed to evaluate the similarity of the patch $\widetilde{I}_b$ to the input patch X_p based on the spatial distance. The motivation is that a synthesized cartoon at patch p is mainly influenced by the nearby patches that are close spatially. For example, if an input patch X_p belongs to the right eye region, it is reasonable to set large weights for those synthesized paths neighboring to the right eye during the cartoon generation. After taking this weight, a lot of patches far from the patch p will be eliminated even if they hold low intensity distances.

Assume c_p and c_q denote the patch center coordinates of patches p and q, respectively. The spatial weight w_4 can be written as

$$w_4(p,q) = \exp\left(-\frac{\|c_p - c_q\|^2}{\sigma_4^2}\right) \quad (6)$$

where the parameter σ_4 adjusts the spatial similarity. If more candidate patches are taken into account, the large value of σ_4 should be adopted. Note that it is more preferable to learn a spatial correspondence prior between different locations, which may be helpful for the cartoon synthesis.

It is noticed that the Gaussian function is used to measure these similarities. The main reason is that the Gaussian function is radially symmetric and shift-invariant, which is insensitive to overall additive changes of image intensity [33]. Of course, for sparsity, other functions, such as a truncated Laplacian cost function [36], can also be used to select a subset of reference patches in the dictionary. Fig. 5 shows an example of weight computation results, where two input patches are selected from the left eye region (i.e., red window) shown on the left figure. Using (2), we compute the total weights from the training photo-cartoon patches. The first six pairs of photo-cartoon patches with highest weights are shown in the middle figure of Fig. 5, which show that the input patches match well with the search results. The corresponding weights of all training patches are computed and plotted on the right side of Fig. 5, which exhibits sparse property. In addition, we also compute the correlation coefficients for the patches given in the first image of Fig. 5. Fig. 6 shows the corresponding results of $W - w_1$, $W - w_2$,

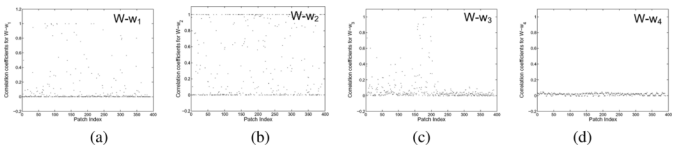

Fig. 6. Correlation coefficients for the patches in the first image shown in Fig. 5. (a)–(d) Results for $W - w_1$, $W - w_2$, $W - w_3$, and $W - w_4$, respectively.

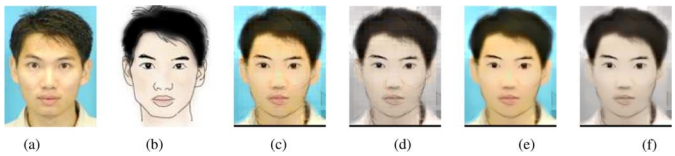

Fig. 7. Illustration of guided face synthesis. (a) Input face photo. (b) Face cartoon by the artist. (c) Synthesized result by (7). (d) Synthesized result by (8). (e) and (f) Post-processing results of (c) and (d), respectively.

$W - w_3$, and $W - w_4$, respectively, which display their contributions to the final weight for the input image.

5) Face Colorization: Once the guided cartoon synthesis Z is obtained, we can colorize it using various methods. In this section, we will introduce some methods to colorize the initial face synthesized result.

1) *Color-Preserving Colorization:* For each pixel in a synthesized cartoon, one can trivially preserve the color information using the input face photo X. Let Y, Cb, and Cr denote the luminance and the two chrominance components in the YCbCr color space, respectively. The colorization of a synthesized cartoon can be expressed as

$$\begin{aligned} Y_{syn}(i,j) &= Z(i,j) \\ Cb_{syn}(i,j) &= Cb_{input}(i,j) = Cb_X(i,j) \\ Cr_{syn}(i,j) &= Cr_{input}(i,j) = Cr_X(i,j) \end{aligned} \tag{7}$$

where $Y_{syn}(i,j)$ denotes the luminance value of the synthesized cartoon at pixel (i,j), and subscripts $_{syn}$ and $_X$ correspond to the synthesis and the input images, respectively. Fig. 7(c) shows the colorization result for an input face photo given in Fig. 7(a), where the synthesized cartoon exhibits similar color as the input photo.

2) *Color-Mapping Colorization:* Instead of directly replacing the chrominance components with the input face photo, the second method is to colorize the initial synthesized result by using the training cartoons. It is known that face region usually exhibits the similar skin-color feature regardless of different skin types. The values of chrominance for different facial skin-colors are narrowly and consistently distributed in the YCbCr color space [7]. In our work, we first choose Cb and Cr channels to generate a 2-D color descriptor for each pixel in training face photos. Then all pixels are quantized into M descriptors by using the k-means clustering algorithm, which are expressed by v_j, $j = 1,2,\ldots,M$. Let l_j denote the index set that belongs to the center v_j. We employ this index set l_j to compute the corresponding cluster v'_j in training cartoons. Finally, we can obtain the chrominance cluster pairs (v_j, v'_j) for the photo-cartoon images.

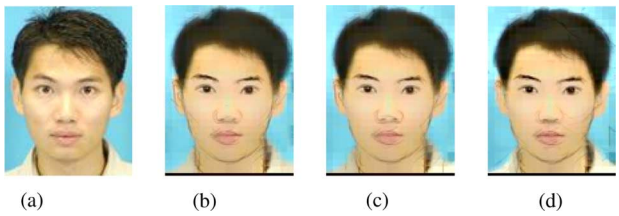

Fig. 8. Synthesis results for different color spaces. (a) Original image. (b) Synthesized result for L^*a^*b color space. (c) Synthesized result for YUV color space. (d) Synthesized result for YCbCr color space.

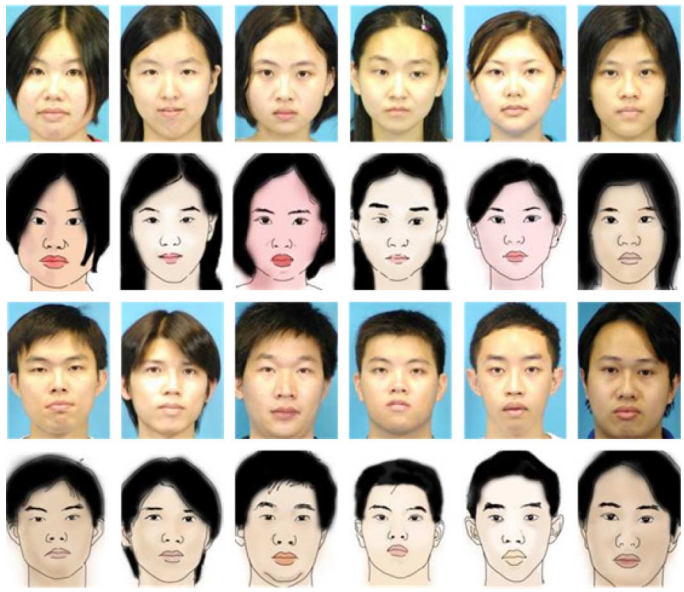

Fig. 9. Examples of training face photos and cartoons from the CUHK student database.

Given a synthesized cartoon Z, the colorization output can be written as

$$\begin{aligned} Y_{syn}(i,j) &= Z(i,j) \\ Cb_{syn}(i,j) &= v'_j, \quad \text{if } Cb_{input}(i,j) \in v_j. \\ Cr_{syn}(i,j) &= v'_j, \quad \text{if } Cr_{input}(i,j) \in v_j. \end{aligned} \tag{8}$$

An example of the colorization result for an input face photo in Fig. 7(a) is illustrated in Fig. 7(d). Here, the cluster M is set to 100. It can be seen that similar color with the training cartoon can be achieved for the synthesized cartoon.

Note that the color-preserving method is used to paint the synthesis in the very similar way with the original photo, which can provide us with the direct comparison of the synthesized cartoon to the original photo. Generally, the artist would like to paint a cartoon in more vivid way and emphasize the outline. To achieve this goal, more artist works including the painting style should be required by incorporating the accurate facial feature extraction. In addition, we perform the colorization in YCbCr color space in this work. Other color spaces can also be used, such as CIE $L^*a^*b^*$ or YUV space. Fig. 8 shows the synthesized results for different color spaces, which include $L^*a^*b^*$,

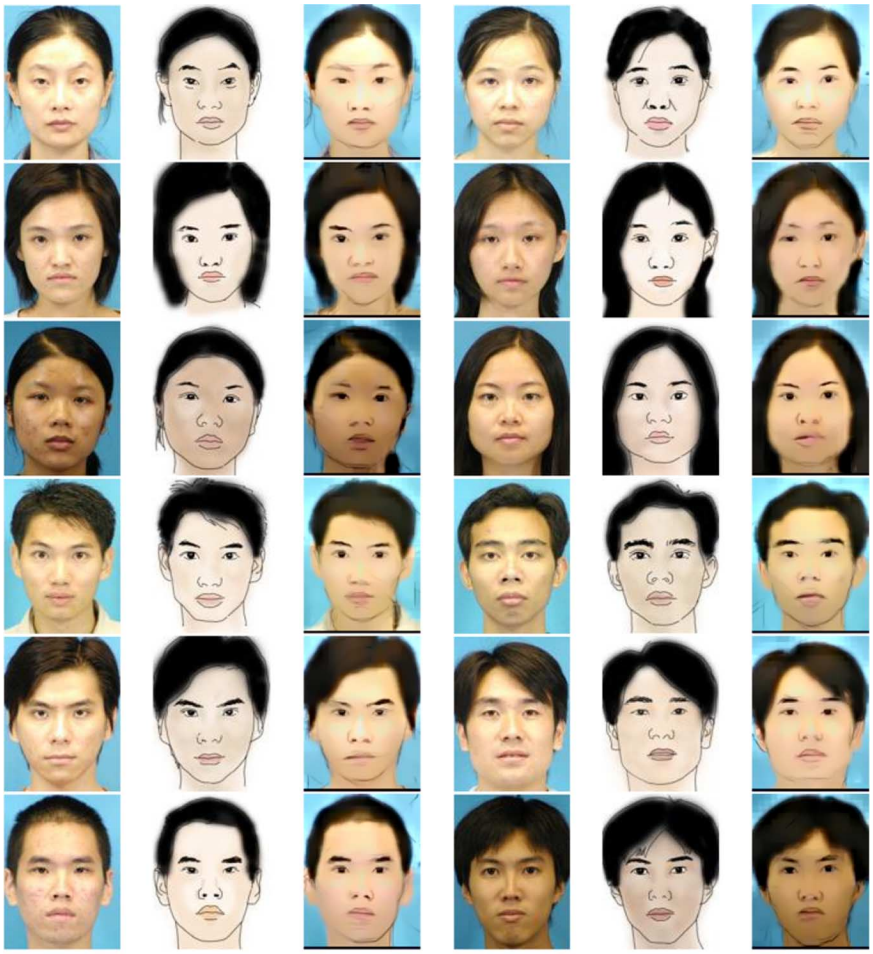

Fig. 10. Face cartoon synthesis results. (a) and (d): Face photos. (b) and (e): Cartoon drawn by the artist. (c) and (f): Synthesized cartoon.

YUV, and YCbCr spaces. We can see that similar results can be achieved for different color spaces.

6) Post-Processing: After face colorization, we can obtain a color synthesized cartoon. However, some artifacts can be still observed due to the patch-based method. In order to reduce the blocking artifact, we employ a post-processing technique to the colorization result based on a non-local mean (NL-means) algorithm [29], which uses image self-similarity to reduce the noise by averaging similar pixels. Since the NL-means algorithm takes advantage of the redundancy and self-similarity of the image, it can remove the blocking artifact while keeping most of the meaningful information. Fig. 7(e) and (f) shows the post-processing results of two colorization outputs in Fig. 7(c) and (d), respectively. It can be seen that good visual quality can be achieved after reducing the blocking artifacts.

III. Experiments

In this section, we verify the performance of our proposed face cartoon synthesis algorithm on several face photos, which were commonly used for face sketch synthesis research [15] together with additional face photos that we collected from miscellaneous sources.

A. Parameters Setting

Before the face cartoon synthesis, we first introduce the parameters setting in our experiments. Given an input face photo, we decompose it into a lot of overlapping patches with the size of 16×16. The overlapping distance is chosen 11 pixels. To compute the pyramid HOG descriptor, we adopt a three-level decomposition as shown in Fig. 3. Four control parameters defined in (3), (4), (5), and (6) are set to $\sigma_1 = 60$, $\sigma_2 = 0.02$, $\sigma_3 = 10$, and $\sigma_4 = 50$, respectively, which show good performance from our empirical study.

B. Results for the CUHK Database

We first evaluate our method on a set of CUHK face database used in [13] and [15]. A total of 100 face photos are split into two parts: 50 training photos and 50 test photos. Fig. 9 shows some training photo-cartoon pairs from the CUHK student database. In Fig. 10, we show some examples of our synthesized results by the color-preserving colorization for some test face photos. The input photos are shown in Fig. 10(a) and (d). The original cartoons drawn by the artist are given in Fig. 10(b) and (e). Fig. 10(c) and (f) shows our synthesized results by the color-preserving colorization. It can be seen that our method is able to generate cartoon faces successfully from test face photos. For example, for the first girl, although strong lighting can be observed in the input photo, the face cartoon is well synthesized,

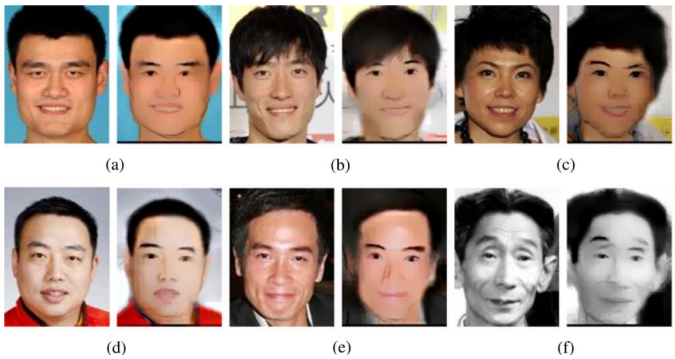

Fig. 11. Face cartoon synthesis results. (a)–(d): Chinese athletes. (e) An actor. (f) An artist. Left and right images in (a)–(f): Original images and synthesized cartoons, respectively.

which exhibits similar appearance as the original cartoon. In addition, for the last face photo under dark frontal lighting, our method still generates a cartoon image successfully.

C. Results for the Other Faces

We also tested our proposed method on some face photos obtained from the web, which contains four famous Chinese athletes, an actor, and an artist. The input photos are shown in Fig. 11. Compared with previous photos, more complex background and small pose variations can be observed for the photos in Fig. 11(b), (c), and (f). In addition, a face photo with gray color (see the last figure) is also used to evaluate our method. The right images of Fig. 11(a)–(f) show the synthesized cartoons, which demonstrate the effectiveness of our method. It is noticed that since all the synthesized patches come from the face training set, i.e., face photos and cartoons drawn by the artist, it is still difficult to generate the complex background in terms of the face-based training set.

D. Comparison Results With Markov Network

We next compare our face cartoon synthesis results with the results obtained by the Markov network-based method, which is used to probabilistically model the relationships between underlying patterns and observed data, and between neighboring underlying patterns [6]. A Markov network can be solved using the learning and inference phases based on the messages propagation along the network. The parameters of the network can be learned from training data and then used to estimate an underlying pattern. The details of Markov network can be referred to [6]. It has been successfully applied for solving low-level vision problems such as super-resolution [6] and photo-sketch synthesis [13], [15].

Given an image pair, this model uses Bayesian belief propagation to find a local maximum of the posterior probability. Here, a fast one-pass algorithm [30] is used to approximate iterative solution to the Markov network. To perform a fair comparison, the same set of photo-cartoon pairs is used to generate the 16×16 training patches. The overlapping patches are also traversed in raster-scan order with 11 pixels overlapping distance. At each step, a best patch is selected by a nearest neighbor search from the training set based on the minimum matching error. The matching error consists of two part, i.e., the difference between a query patch and the training set, and the difference between the query patch and adjacent, previously obtained patches.

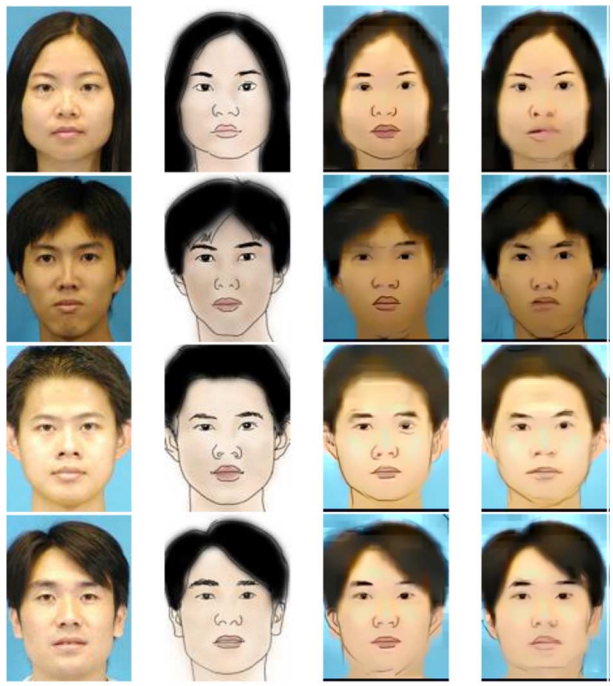

Fig. 12. Comparisons with the Markov network-based method. (a) Face photo. (b) Cartoon by the artist. (c) Results by the Markov network. (d) Results by our method.

Fig. 12(c) shows the synthesis results by the Markov network. Compared with our results in Fig. 12(d), we can see that robust performance can be achieved by our method. The facial components such as eyes and eyebrows are captured as well for our method. An interesting phenomenon can be observed that Markov network sometimes has more salient outline, like in the third and fourth rows of Fig. 12(c). The main reason is that Markov network synthesizes a cartoon patch using the best matching patch from the training set based on the overlap stitching process. Compared with the guided filtering, some distinct outlines may be preserved in some synthesized paths, such as the chin outline.

E. Quantitative Comparison

Since cartoon is a human-subjective conception, which is similar to the original face, but with much more distinct outline, it is still a challenging task to evaluate the performance of cartoon synthesis based on a well-designed quantitative method. In this section, we tend to evaluate our method from two aspects, i.e., objective and subjective quality evaluation.

Recognition rate, as an objective measurement, has been used to measure similarity between synthesized sketch and sketches drawn by the artist in [15]. Here, we borrow this idea to perform a quantitative evaluation for the cartoon synthesis results. We compute the rank-1 recognition rate (i.e., the percentage of correct matches out of all test images) based on local binary patterns (LBP) feature. As mentioned in the previous section, the 100 photo-sketch pairs are divided into two subsets. Fifty photo-cartoon pairs are used for generating the training patches. The other photo-cartoon pairs are used for the testing. Here, we

TABLE I
FOUR TEST MODES FOR OBJECTIVE COMPARISON

Test Method	Description
Test1	Query: face photo, Database image: 50 synthesized cartoons by our method
Test2	Query: cartoon drawn by the artist, Database image: 50 synthesized cartoons by our method
Test3	Query: synthesized cartoon, Database image: 50 test face photos
Test4	Query: synthesized cartoon, Database image: 50 test cartoons drawn by the artist

TABLE II
RECOGNITION RATES FOR OBJECTIVE COMPARISON

Test Models	Test1	Test2	Test3	Test4
Markov Network	100%	62%	98%	86%
Proposed	100%	82%	100%	86%

use the 50 test photo-cartoon pairs with corresponding synthesized cartoons to perform the quantitative test. By choosing different images as the query image, we evaluate the performance in four different cases, which are described in Table I. For example, for the first two cases (i.e., Test1 and Test2), the 50 synthesized cartoons that are obtained by the proposed method are considered as database images, while the test face photo and the test cartoon drawn by the artist are treated as the query image, respectively. For the Test3 and Test4, we employ our synthesized cartoon as the query image to match the 50 test face photos and the 50 test cartoons drawn by the artist, respectively. The recognition rates are used to evaluate the synthesis performance, which are given in Table II. We can see that high recognition rates can be achieved for both methods when the evaluation is performed between the synthesized cartoon and the face photo. For most tests, our method outperforms the Markov network-based method. In the Test2, our method achieves recognition ratio with 82%, which beats the Markov network-based method by 20%.

For a more convincing evaluation, we perform a subjective recognition test for the proposed method. A synthesized cartoon by the Markov network or our method is considered as a query image, while the 50 face photos are treated as database images. We invited 15 subjects to find the best match photo to the query example. Experimental result shows that about 81.07% and 91.60% recognition rate on average can be achieved for the Markov network and our proposed method, respectively. We can see that our method obtains the higher recognition rate with 12.99% improvement on average. It also means that the synthesized cartoon by the proposed model is more similar to the cartoon drawn by the artist based on the subjective evaluation.

F. Discussions

Our face cartoon synthesis is performed based on local patches. In Fig. 13, we compare the cartoon synthesis performance using different offsets for the 16×16 patch. Note that the offset is used to control the number of overlapping pixels. When a small offset is selected, e.g., $=2$, the synthesized cartoon tends to lose some face structures due to over-smoothing process between adjacent patches. However, there is distinct distortion when the offset is set to a large value, e.g., $=15$ as shown in Fig. 13(e). Such results can be interpreted by the

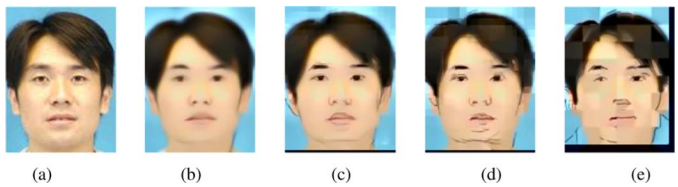

Fig. 13. Synthesis results for different steps. (a) Face photo. (b)–(e) Results by steps 2, 5, 10, and 15, respectively.

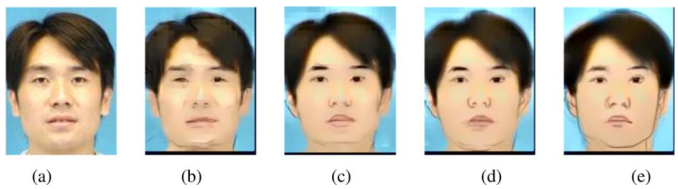

Fig. 14. Synthesis results using different patch sizes. (a) Face photo. (b)–(e) Results by path sizes 10×10, 16×16, 20×20, and 30×30, respectively.

smoothness weight w_3 in (2), which is defined to keep the smoothness variation between neighboring patches. From (5), we can see that small offset values will take more overlapping pixels into smoothness constraint, which results in the blurring effect. On the contrary, large offset can preserve the details by reducing the overlapping pixels between adjacent patches, which may lead to the synthesis distortion. From our empirical study, good performance can be achieved when the offset takes value between 4 and 6. The corresponding result with $\text{offset}=5$ is illustrated in Fig. 13(c).

In Fig. 14, we compare the synthesized cartoons using different patch sizes. Four patch sizes are used to observe the performance, i.e., 10×10, 16×16, 20×20, and 30×30, which are shown in Fig. 14(b)–(e), respectively. The offset is chosen to 5 pixels for all patch sizes. For a small patch size such as 10×10, some facial components cannot be well synthesized such as the nose. The main reason is that it is difficult to describe local structure using a small patch. When the patch size is very large, such as 30×30 in Fig. 14(e), the face contour can be well synthesized. But annoying ghost will be yielded along with it. Based on our empirical study, good performance can be achieved when the patch size takes value between 16 and 20.

In Fig. 15, we compare the synthesized cartoons using different number of training images. Four training sets are used to observe the performance, which consist of 10, 20, 30, and 50 training photo-cartoon pairs. The offset is chosen to be 5 pixels for all training images. The synthesized results for an input face photo in Fig. 15(a) are shown in Fig. 15(b)–(e), respectively. Better result can be obtained when the number of training photo-cartoon pairs is larger than 10. The similar results

Fig. 15. Synthesis results using different number of photo-cartoon pairs. (a) Face photo. (b)–(e) Results by 10, 20, 30, and 50 training images, respectively.

Fig. 16. Synthesis results of Markov network-based method using different number of photo-cartoon pairs. (a) Face photo. (b)–(e) Results by 10, 20, 30, and 50 training images, respectively.

are achieved when the number of training images exceeds 20. For the same training images, we also synthesize face cartoons using the Markov network-based method. Fig. 16 shows the corresponding results. It can be seen that the number of training images highly affects the synthesized results. Bad performance will be obtained when there are not enough training images.

In this work, we define a general linear synthesis process, which consists of an input photo X, a guidance image pair I and $\widetilde{I}$, and synthesized image Z. Compared with the guided filtering in [31] and [32], there are two distinct differences with our guided synthesis. Firstly, the family of guided filtering is defined as the weighted average for input image X instead of guided image I in (1). It means that the final output image of guided filtering must be the average processing for the input image, which cannot perform the image synthesis. Secondly, the weight W_{ij} in guided filtering method is only computed in the guided image I, which is independent of X [32]. However, in our work, the weight W consists of four cost functions, which are computed from various images, such as X, I, $\widetilde{I}$, and Z.

For the Markov network-based methods [13] and [15], they highly depend on two techniques, i.e., search/match algorithm and belief propagation. The first is to search and find the best match in high dimension of the search space. The latter is to evaluate the spatial consistency for a good match. In our work, we measure the contribution of each patch to the photo synthesis simultaneously based on four consistency constraints. Only those patches that satisfy all constraints will be imposed with high weights for the cartoon synthesis. Given a limited set of photo-cartoon pairs, we compute the weighted average over all reference patches in the training set instead of selecting one or a few among them, which can reduce synthesized errors during the cartoon generation.

Since we consider all patches into the cartoon patch synthesis, the blurring effect may be introduced. Currently, there is no standard method used for the quantitative analysis of the blurring effect, which mainly depends on subjective evaluation. In order to perform the quantitative analysis for the blurring effect, we first compute the mean absolute value of the differences between the original image and the blurred image by a point spread function (PSF) with the size of 5×5 and the standard deviation 1.5. The difference is also called the focused saliency map (FSM) in [34]. Compared with the face photo, there is about 8.35% FSM decrease on average for our method. Experiment shows that there is small blurring effect introduced by the synthesized cartoon with respect to the input face photo. However, most of the details in the facial parts can be preserved well after the proposed method.

IV. Conclusion and Future Works

In this paper, we propose a new method to synthesize a cartoon from an input face photo. The synthesis is performed based on a local linear model, which generates a cartoon photo by incorporating the content of guidance images taken from the training set. Four cost functions are defined to achieve the guided synthesis. The first photo-photo weight is used to measure the similarity between an input photo patch and a training photo patch. The second is the photo-cartoon weight, which aims to estimate the likelihood by computing the similarity between a cartoon patch and an input photo patch. The third weight is to set the smoothness constraint for neighboring synthesized patches. This weight is computed from the synthesized photo. The final weight is used to evaluate the similarity of patch to the input patch according to the spatial distance. Experimental evaluation on a number of face photos demonstrates the effectiveness of the proposed method on the face cartoon synthesis.

In the future, we plan to incorporate the high level semantic feature for face synthesis, such as AAM [35], where we can construct the appearance model pairs to improve the cartoon synthesis performance. It is believed that some high level semantic analysis for the face photo will be very useful for the cartoon generation, which can give some information to guide the cartoon generation for different region like eye, hair, face, mouth, etc. For example, we can provide more salient details in those regions such as eyehole, canthus, laughline, etc.

Acknowledgment

The authors would like to thank all the anonymous reviewers for the constructive comments and useful suggestions that led to improvements in the quality, presentation, and organization of this paper.

References

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR)*, Kauai, HI, Dec. 2001, vol. 1, pp. 511–518.

[2] S. Z. Li and Z. Q. Zhang, "FloatBoost learning and statistical face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1112–1123, Sep. 2004.

[3] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR)*, Maui, HI, 1991, pp. 586–591.

[4] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 10, pp. 1042–1052, Oct. 1993.

[5] S. Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum, "Statistical learning of multi-view face detection," in *Proc. Eur. Conf. Computer Vision (ECCV)*, 2006, vol. 2353, pp. 117–121.

[6] W. Freeman, E. Pasztor, and O. Carmichael, "Learning low-level vision," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 25–47, 2000.

[7] H. Li and K. N. Ngan, "Saliency model based face segmentation in head-and-shoulder video sequences," *J. Vis. Commun. Image Represent. (Elsevier Science)*, vol. 19, no. 5, pp. 320–333, 2008.

[8] C. Liu, H.-Y. Shum, and W. T. Freeman, "Face hallucination: Theory and practice," *Int. J. Comput. Vis.*, vol. 75, no. 1, pp. 115–134, 2007.
[9] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1063–1074, Sep. 2003.
[10] H. Chen, L. Liang, Y.-Q. Xu, H.-Y. Shum, and N.-N. Zheng, "Example-based automatic portraiture," in *Proc. 5th Asian Conf. Computer Vision*, 2002.
[11] X. Tang and X. Wang, "Face sketch synthesis and recognition," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2003, vol. 1, pp. 687–694.
[12] Z. Xu, H. Chen, S.-C. Zhu, and J. Luo, "A hierarchical compositional model for face representation and sketching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 955–969, Jun. 2008.
[13] X. Wang and X. Tang, "Face photo-sketch synthesis and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 1955–1967, Nov. 2009.
[14] P. Sehgal and P. S. Grover, "A novel approach to cartoon style rendering of an image with an approximated crayon texture," in *Proc. Int. Conf. Computer Graphics, Imaging and Visualization (CGIV)*, 2004.
[15] W. Zhang, X. Wang, and X. Tang, "Lighting and pose robust face sketch synthesis," in *Proc. European Conf. Computer Vision (ECCV)*, Crete, Greece, Sep. 5–11, 2010, vol. 6, pp. 420–433.
[16] W.-I. Hwang, P.-J. Lee, B.-K. Chun, D.-S. Ryu, and H.-G. Cho, "Cinema comics: Cartoon generation from video stream," in *Proc. Computer Graphics Theory and Applications*, 2006, pp. 299–304.
[17] H. Chen, N. Zheng, L. Liang, Y. Li, Y. Xu, and H. Shum, "PicToon: A personalized image-based cartoon system," in *Proc. ACM Multimedia*, New York, 2002, pp. 171–178.
[18] R.-L. Hsu and A. K. Jain, "Generating discriminating cartoon faces using interacting snakes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 11, pp. 1388–1398, Nov. 2003.
[19] F. Wen, S. Chen, and X. Tang, "EasyToon: Cartoon personalization using face photos," in *Proc. 16th ACM Int. Conf. Multimedia*, 2008, pp. 1021–1022.
[20] J. Wang, Y. Xu, H.-Y. Shum, and M. F. Cohen, "Video tooning," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 574–583, 2004.
[21] H. Chen, Y.-Q. Xu, H.-Y. Shum, S.-C. Zhu, and N.-N. Zheng, "Lighting and pose robust face sketch synthesis," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2001, vol. 2, pp. 433–438.
[22] Q. Liu, X. Tang, H. Jin, H. Lu, and S. Ma, "A nonlinear approach for face sketch synthesis and recognition," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005, vol. 1, pp. 1005–1010.
[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[24] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. British Machine Vision Conf. (BMVC)*, 2002, pp. 384–393.
[25] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," in *Proc. Eur. Conf. Computer Vision (ECCV)*, 2004, pp. 404–416.
[26] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, 2004.
[27] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, 2005, pp. 886–893.
[28] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid Kernel," in *Proc. ACM Int. Conf. Image and Video Retrieval*, 2007.
[29] A. Buades, B. Coll, and J. M. Morel, "A non local algorithm for image denoising," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2005, vol. 2, pp. 60–65.
[30] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar.–Apr. 2002.
[31] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen, and K. Toyama, "Digital photography with flash and no-flash image pairs," in *Proc. ACM SIGGRAPH*, 2004.
[32] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. Eur. Conf. Computer Vision (ECCV)*, Crete, Greece, Sep. 5–11, 2010, vol. 1, pp. 1–14.
[33] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, Bombay, India, 1998.
[34] H. Li and K. N. Ngan, "Unsupervised video segmentation with low depth of field," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 12, pp. 1742–1751, Dec. 2007.
[35] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001.
[36] K. I. Pedersen, P. E. Mogensen, and B. H. Fleury, "A stochastic model of the temporal and azimuthal dispersion seen at the base station in outdoor propagation environments," *IEEE Trans. Veh. Technol.*, vol. 49, no. 2, pp. 437–447, Mar. 2000.

Hongliang Li (M'06) received the Ph.D. degree in electronics and information engineering from Xi'an Jiaotong University, Xi'an, China, in 2005.

From 2005 to 2006, he joined the Visual Signal Processing and Communication Laboratory (VSPC) of the Chinese University of Hong Kong (CUHK) as a Research Associate. From 2006 to 2008, he was a Postdoctoral Fellow at the same laboratory in CUHK. He is currently a Professor in the School of Electronic Engineering, University of Electronic Science and Technology of China. His research interests include image segmentation, object detection, image and video coding, visual attention, and multimedia communication system. He has authored or co-authored numerous technical articles in well-known international journals and conferences. He is a co-editor of a book titled "Video segmentation and its applications" (New York: Springer, 2011) He was involved in many professional activities.

Dr. Li is a member of the Editorial Board of the *Journal on Visual Communications and Image Representation*. He served as TPC members in a number of international conferences, e.g., ISPACS2005, PCM2007, PCM2009, and VCIP2010, and served as Technical Program co-chair in ISPACS2009, and general co-chair of the 2010 International Symposium on Intelligent Signal Processing and Communication Systems. He will serve as a local chair of the 2014 IEEE International Conference on Multimedia and Expo (ICME). He was selected as the New Century Excellent Talents in University, Chinese Ministry of Education, China, in 2008.

Guanghui Liu received the M.Sc. and Ph.D. degrees in electronic engineering from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2002 and 2005 respectively.

In 2005, he joined Samsung Electronics, South Korea, as a Senior Engineer. Since 2008, he has been with the School of Electronics Engineering, UESTC, as an Associate Professor. His general research interests include digital signal processing and telecommunications, with emphasis on digital video transmission, image processing, and OFDM techniques. In these areas, he has published tens of papers in refereed journals or conferences, and received about ten patents (pending).

Dr. Liu served as the publication chair of the 2010 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2010).

King Ngi Ngan (M'79–SM'91–F'00) received the Ph.D. degree in electrical engineering from the Loughborough University, Loughborough, U.K.

He is currently a Chair Professor at the Department of Electronic Engineering, Chinese University of Hong Kong. He was previously a full professor at the Nanyang Technological University, Singapore, and the University of Western Australia, Australia. He holds honorary and visiting professorships of numerous universities in China, Australia, and South East Asia. He has published extensively including three authored books, six edited volumes, over 300 refereed technical papers, and edited nine special issues in journals. In addition, he holds ten patents in the areas of image/video coding and communications.

Prof. Ngan was an associate editor of the *Journal on Visual Communications and Image Representation*, as well as an area editor of *EURASIP Journal of Signal Processing: Image Communication*, and served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and the *Journal of Applied Signal Processing*. He chaired a number of prestigious international conferences on video signal processing and communications, and served on the advisory and technical committees of numerous professional organizations. He co-chaired the IEEE International Conference on Image Processing (ICIP) held in Hong Kong in September 2010. He is a Fellow of IET (U.K.) and IEAust (Australia), and an IEEE Distinguished Lecturer in 2006–2007.