Sketch Realizing: Lifelike Portrait Synthesis From Sketch

Di Wu and Qionghai Dai

TNList and Department of Automation, Tsinghua University, Beijing, P. R. China

Abstract

People usually visualize their imaginations or memories through sketching. However, it might be difficult to record color and texture details powered by a large database of photographs gathered from the Web. This paper deals with the imagination visualization problem for human faces. A framework for synthesizing lifelike portraits from user specifications and input sketches is proposed which is an inverse process of sketch generation. The proposed framework synthesizes the realistic appearance of a face by taking parts from an annotated face library of photographs and stitching them together followed by further deformation.

The algorithm consists of three parts primarily: First, given user specifications and an input sketch, search good matches for each facial component in the library; Second, extract each facial component from matching source images and composite them together; Third, deform the synthesized lifelike portraits to further approximate the sketch. The key component lies in a measurement for finding the right contents to bridge the gap between shapes and lifelike images. A set of diverse lifelike portraits can be synthesized from a single sketch. The effectiveness of the proposed approach is demonstrated with a variety of experimental results.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing algorithms; I.4.10 [Computing Methodologies]: Image processing and computer vision—Image representation

1. Introduction

An easy and direct way to record the visual memory or imagination in mind is to draw sketches on the paper. Although professional artists might do a good job in visualizing their imaginations through sketching, for most people who are not artists, sketches might be too concise to reflect color and texture details. A simple and curbstone face sketch mainly consists of lines which only convey the shape and contour information while leaving out color and texture information. However, the latter two properties are quite important in a lot of cases. We believe that a promising solution to the face visualizing problem is to synthesize lifelike portraits from a given sketch and desired attributes based on a collection of stock images. This paper considers the simplest form of sketches which only contain shape information without considering the pseudo-sketch that is rich of textures.

From the perspective of mathematics, sketch realizing problem is very under-constrained, because sketches only convey shape information, while the lifelike portraits are comprised of colors and textures under shape constraints. So the primary challenge of face sketch realizing problem turns to how to bridge the gap between shapes and lifelike images. This work is based on two premisses: 1) Facial components of different people are similar in general. For instance, if eyes on a certain face are replaced with another pair, the modified face still looks like a real person; 2) Facial components similar in shapes tend to have similar textures. For example, most smiling mouths show teeth and widely open lips textures. Based on the previous premisses, the following rationales are proposed: 1) Since facial components are similar in configurations, it is reasonable to divide a face sketch into independent facial components and deal with each separately. 2) It is reasonable to use shape similarity as a measurement to find candidates for each facial component in real images. The second rationale is enabled by the “Data-Bridge”, which consists of a large number of face images with annotated shapes as well as corresponding colors and textures. This notion is illustrated in Figure 3b.

In this paper, we present a framework for sketch realizing (Figure 1). Firstly, a library consists of a variety of anno-
tated face photographs should be built. Secondly, to realize a given sketch with user specified attributes, the framework performs a keyword search to extract corresponding face images containing only desired attributes from the library to form a sub-library. Thirdly, provided with the input sketch, the system automatically searches good matches for each facial component in the sub-library to form a set of candidate result combinations. Each column in the left bottom image in Figure 1 is some searched candidates for each facial component, and through selecting from each facial component candidates, a set of combinations can be generated (e.g., images bounded with yellow lines are chosen to form a combination.) Valid combinations are fused seamlessly to generate a set of compositions. In the final step, as rigid as possible deformation is performed to further approximate to the input sketch in shape.

Sketch realizing has several compelling applications. Firstly, it can be used to create faces of wanted criminals from witness description and provided sketches. With the face sketch realizing approach, people can synthesize lifelike face images according to their imaginations. Although the synthesized face cannot exactly be the wanted person, they have some visual similarities which might help the police officer. Secondly, sketch realizing is an interesting entertainment application to help people to visualize their Mr. or Mrs. Dreamings easily. Thirdly, people can use the system to acquire expected facial features from library images (e.g., the user is able to change his or her small eyes to big double eyes). Fourthly, people can transfer shapes of library photographs to achieve desired shape characteristics, such as changing from round face to thin face.

Our contributions are as follows: 1) Presenting a framework that could easily generate a set of visually similar faces given attribute specifications and an input sketch; 2) Utilizing shape matching as a measurement to bridge the gap between shapes and lifelike images; 3) Exploiting the proposed approach in several other interesting applications.

The structure of this paper is as follows. We discuss related work in Section 2. In Section 3, we describe how to build the face library. In Section 4, “search engine” is discussed in detail. The composition and deformation techniques are presented in Section 5. The effectiveness of the proposed approach is demonstrated in Section 6. We conclude in Section 7.

2. Related work

**SKETCH GENERATION** The sketch captures the most informative part of an object, in a much more concise and potentially robust representation [BHD’92, XCZL08]. In computer vision, sketch generation is an active area of research. There are several representative works [CXS’01, LTJ05, XCZL08]. Chen et al. [CXS’01] collected local evi- dences from artistic drawings in the training set and generated facial sketches using an example-based approach. Liu et al. [LTJ05] exploited the idea of locally linear embedding and generated pseudo-sketch based on local linear preserving of geometry between photo and sketch images. Xu et al. [XCZL08] built a hierarchical compositional graph model which is a three-layer And-Or graph of human faces based on stochastic grammars. Cartoon facial sketches using the graph model can be generated. Besides, faces can be reconstructed based on the hierarchical compositional graph model. This is different from our work, because it is a direct reconstruction from the graph model while ours is the process of synthesis from a given sketch.
SKETCHING REALITY Chen et al. [CKX08] introduced sketching reality, the process of converting a freehand sketch into a realistic-looking model and applied it to architectural designs. The core lied in 2.5D-geometry interpretation and the extracted topology analysis. However, our work aims at synthesizing lifelike face images based on input sketch, which requires no 2.5D or 3D geometric model.

SHAPE MATCHING Shape context descriptor has been used in many forms for matching shapes and object recognition. It was first presented by Belongie [BMP01] for measuring the similarity between two shapes as well as object recognition, and later modified for rapid shape retrieval by Mori [MB05]. In this paper, the idea of matching shapes with shape context is exploited.

GRAPH CUT AND GRADIENT-DOMAIN FUSION Graph Cut optimization [BJ01] and gradient-domain fusion [FLW02, PGB03] have already been used to create new images from a variety of sources in various forms for many tasks in both computer vision and computer graphics [ADA04, HE07].

3. Face Library Creation and Related Details

This section details how to build the annotated library of face photographs. In order to obtain realistic synthesis results, it is necessary to incorporate a large amount of photographs with diversity. The face library used in this paper is constructed by searching websites, such as Google Image, using keywords queries likely to find images containing faces, as well as taking advantages of free face databases containing relatively high quality images, such as IMM face database [SEL03].

The procedure of adding an image into the library is as follows (Figure 2a). Firstly, only images with relatively high resolution are kept. The threshold can be set to default (e.g., 400 × 500) or specified by the user. Secondly, 88 feature landmarks which are located on the boundaries of different facial features: the outlines of the face, the inner and outer boundaries of lips, two eyebrows, two eyes, and the nose (Figure 2b) are marked out by hand. Thirdly, a statistical distribution of features points is calculated from the current library to scale and rotate the input image properly. Fourthly, a list of options for keyword attributes is provided for the user to choose from (e.g., the age attribute is divided into 6 zones, the user just needs to choose one with a tick). Besides, there are many other attribute options provided to the user in the form of single or multiple choices. Fifthly, because occlusions and harsh lighting might greatly affect facial component extraction and composition, and in most cases only partial face is affected by the above two effects, the other unaffected parts can still be viewed as the matching candidates, as a result, the usability of each facial component in every face image is labeled with True or False representing usable or not. The annotated image is illustrated as (Figure 2b), which is then added into the face library with a sequence number. The corresponding attributes and usability are saved as lists of boolean values with same sequence for searching.

4. Search Engine

Given an input sketch, the key component of our approach is to bridge the gap between shapes and lifelike images by extracting candidate facial components from the face library which most resemble the given sketch in shape (Figure 3b). Shape distance is exploited to measure the similarity between input sketch and face images in the library.

Figure 2: (a) Summary of face library and sub-library creation; see Section 3 and Section 4.1. (b) Illustration of an annotated photograph. (c) Input sketch feature points.

IMAGE DEFORMATION There are many prominent works in this area of research. For example, a multilevel freeform deformation technique is proposed by Lee [LWC96]. Scott et al. [STJ06] provided an image deformation method based on Moving Least Squares using various classes of linear functions including affine, similarity and rigid transformation. In this paper, we use the as rigid as possible deformation method [STJ06].

FACE IMAGE MANIPULATIONS There are three prominent works manipulated face images for interesting applications [BKC08, LCL08]. Leyvand et al. [LCL08] explored a data-driven approach to enhance the facial attractiveness through an automatic small adjustment of feature points positions. A complete system for automatic face replacement in images is presented by Bitouk [BKD08]. The work in this paper also aims at interesting applications on face images, such as face shape transfer, face components swapping and so on. Furthermore, Nguyen et al. [NLET08] proposed an automatic method to decompose a face image into layers through utilizing the differences between the beard and non-beard subspaces, and applied the work to face synthesis and editing, such as image-based shaving.
4.1. Keyword Searching

We exploit keyword searching to build a sub-library containing only faces with desired attributes for further searching. The user chooses one or more attributes defined in the library creation stage. The system automatically extracts images with desired attributes to form a sub-library. Furthermore, the sub-library is separated into five independent facial component libraries for eyebrows, eyes, nose, lips and face base according to the usability labels. The pipeline of keyword searching is incorporated in Figure 2a. Building a sub-library has two advantages: first, the sub-library itself is a set containing the desired attributes only, where the texture details are constrained and as a result the synthesized results tend to have less artifacts and be more consistent to the imagination by selecting from a more accurate source library; second, searching from a sub-library can significantly accelerate the search process.

4.2. Shape Matching Searching

When the user inputs a sketch, the corresponding 88 feature points are also required (Figure 2c). Because input sketches only contain shape information, searching for matching facial components turns to shape matching in each shape space (e.g., the mouth space, Figure 3a). Since the core of searching lies in the shape constraints, we first introduce a measurement for comparing the similarity between shapes. Shapes can not be effectively represented in Euclidean space. As a result, the concept of shape context descriptor [BMP01, MB05] is exploited to define a shape distance in the shape space. To derive the definition of shape distance for our application, the concept of shape context [BMP01] is introduced as follows.

Shape context is a powerful global descriptor of shapes which captures the positional relations between a point and all the other points on the shape, so corresponding points on two similar shapes will have similar shape context. The shape context of a point \( p_i \) is the uniform \( \log^2 \) space bins. Each bin value is defined to be the number of points falling in the bin. This can be formulated as follows:

\[
h_i(k) = \#(q \neq p_i : (q - p_i) \in \text{bin}(k)), \quad (1)
\]

where \( q \) represents points on the shape other than \( p_i \). \((q - p_i)\) stands for the vector points from \( p_i \) to \( q \) and it is classified to the corresponding bin according to its length \( \log r \) and angle \( \theta \). Experiments in this paper use the classical parameters: 5 bins for \( \log r \) and 12 bins for \( \theta \).
We define the facial component shapes of input sketch as the destination shape $SHP_{dest}$, while the component shapes of different faces in the library are source shapes $SHP_{src}$. The cost of matching two points on different shapes is:

$$C(SHP_{dest}, SHP_{src}) = \frac{1}{2} \sum_{k=1}^{K} \left[ \frac{h_i(k) - h_j(k)}{h_i(k) + h_j(k)} \right]^2,$$

and the shape context distance between the two shapes is defined as follows:

$$D_{sc}(SHP_{dest}, SHP_{src}) = \frac{1}{n} \sum_{p \in SHP_{dest}} \arg \min_{q \in SHP_{src}} C(p, T(q)),$$

where $n$ is the number of points on the shapes, $p$ and $q$ are matching points between destination and source shapes, $T(\cdot)$ represents the estimated Thin Plate Spline (TPS) shape transformation which is mentioned below. Because $SHP_{dest}$ and $SHP_{src}$ both consist of labeled feature points sharing the same semantics, the points on the two shapes are already matched and there is no need to calculate matching using bipartite graph. However, shape context distance is an important part of the measurement for shape similarity.

To estimate the transformation from $SHP_{src}$ to $SHP_{dest}$ to calculate bending energy, the TPS interpolant $f(x, y)$ is used to minimize the following bending energy [BMP01]:

$$I_f = \int \int R \left[ \frac{\partial f}{\partial x} \right]^2 + 2 \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} + \left( \frac{\partial f}{\partial y} \right)^2 dxdy,$$

and the corresponding $f(x, y)$ has the following form:

$$f(x, y) = a_1 + a_2x + a_3y + \sum_{i=1}^{n} \omega_i U(||(x_i - y_i) - (x, y)||).$$

$$U(r) = r^2 \log r^2$$ in the above equation.

The problem of sloving TPS coefficients turns to sloving a linear system of equations, with the constraints $f(x_i, y_i) = v_i$. $v_i$ represents points on the destination shape.

$$\begin{pmatrix} K \\ P \\ \omega \\ a \\ 0 \end{pmatrix} = \begin{pmatrix} \nu \\ 0 \end{pmatrix},$$

with $K_i = U(||(x_i, y_i) - (x_j, y_j)||)$, and $\nu$ are column vectors formed from $\omega_i$ and $v_i$. The bending distance is defined as follows:

$$D_{bending}(SHP_{dest}, SHP_{src}) = \omega^T K \omega.$$ 

Some specific constraints and variable explanations in the above equations (such as $P$ in equation 6) are omitted, please refer to [BMP01] for details. Belongie et al. [BMP01] used 3 iterations of shape context matching and TPS reestimation to measure the shape distance. The resulting source shape is close to the destination one, but there are still some differences that are not eliminated. Considering the particularity of the applications in this paper, human face shapes, including face, eyes, brows, nose and lips shapes, are quite similar in large scales, subtle changes in shapes are quite possible to generate large influence to aesthetic evaluation [LCL08]. As a result, to measure the shape similarity accurately, it is necessary to capture all the differences between them. We define the shape distance as:

$$D_s = D_{sc} + D_{bending} = \sum_{k=1}^{N} D_{sc}(k-1, k) + \sum_{k=1}^{N-1} D_{bending}(k-1, k).$$

We denote that the shape distance definition exploits $N$ iterations and each of the middle transformed shape as $S(k)$ in the $k_{th}$ iteration, with $SHP_{src}$ as $S(0)$ and $SHP_{dest}$ as $S(N + 1)$. In the $k_{th}$ iteration, the TPS transformation is estimated between $S(k-1)$ and $S(N)$, however, the resulting shape is $S(k)$, which still has distance to $S(N)$ for a small number of $N$. As a result, we record the shape context distance $D_{sc}(k-1, k)$ and bending distance $D_{bending}(k-1, k)$ in the $k_{th}$ iteration and go on iterating from the resulting $S(k)$ until the distance between $S(N-1)$ and $S(N)$ is stably around zero. In the last iteration, the resulting shape is $S(N)$ which is no longer transformed by the TPS function, so the bending distance is only summed to $N-1$. Figure 4a, and b show two examples of matching lip shapes. It is clear that compared to the input lip shape of the sketch which is not smiling, the first shape obviously has a smaller shape distance to the destination shape than the second smiling one. The shape distances during iteration are plotted in Figure 4c which reach convergence after 20 times of iterations.

![Figure 5: Examples of Graph Cut and Fusion Results. Images in the top row are the results of directly combining the Graph Cut results. Images in the bottom row are results after gradient-domain fusion.](image-url)
lip are employed in the search for face base candidates, because their combination can convey the facial expression information which might cause large variations in face textures. Third, to keep the visual rationality, it is suggested to constrain the same left and right eye or eyebrow sources, and both brows and eyes to come from the same sources for Caucasians whose eyebrows and eyes are too close to segment. Finally, as mentioned above, the shape differences are relatively small, so it is reasonable to combine the first few search results of each components to generate a set of compositions of these different combinations, which will provide the user with more plausible choices to match his or her imagination. However, certain combinations might cause strange or failure results which should be excluded from the final combination set. This will be explained in the next section.

5. Composition and Further Approximation

After getting a set of matching image combinations for each input sketch, the next step is to extract each facial component from the corresponding matching image and fuse them together. Moving Least Square (MLS) deformation is used to further approximate the composition results to sketch.

5.1. Graph Cut Optimization and Gradient-domain Fusion

Graph Cut is employed to extract each facial component from its source image and gradient-domain fusion to form a composition by solving Poisson equation under Neumann boundary condition.

GRAPH CUT OPTIMIZATION Poisson image editing is a powerful tool [PGB03]. However, it is known that the result is very sensitive to the boundary condition. It is demonstrated that the best result can be gotten when the pixel value differences are constant on the boundary [HL03]. It is also suggested to exploit the minimal cut energy in the difference of gradient domain [HE07]. The proposed framework performs Graph Cut to the difference of two image gradients as suggested by Hays [HE07]. Certain translations are necessary to make sure that each component is aligned to local component center of the corresponding part in the sketch. The Graph Cut optimization [BJ01] exploited in this work can be formulated as follows:

\[ E(M) = \sum_{s} E_{data}(s, M(s)) + \sum_{s,t} E_{smooth}(s, M(s), t, M(t)). \] (9)

The data term constrains the region \( R_{in} \) inside the closed curves connecting feature points to come from source images, while the inverse region of the dilated \( R_{dp} \) which we denote as \( R_{out} \) to come from the face base. The region between \( R_{in} \) and \( R_{out} \) is to be segmented. The smooth term makes sure that the cut breaks the edges with less change. Some results are shown in the top row of Figure 5.

GRADIENT-DOMAIN FUSION The framework uses the following approach: set the right term of Poisson equation to be the combined gradient field from different component source images according to the Graph Cut results.

There is a situation needs to be discussed: Graph Cut is operated for each facial component separately, so it is possible that the resulting neighboring regions are overlapped. As a result, after performing Graph Cut to each set of combinations, overlaps among the resulting regions are detected and the combinations with collisions are eliminated from the combination set. However, since the library is rich and the set of search results is large, there are still many plausible candidates after dropping the ones with collisions.

5.2. Moving Least Square Deformation

The composition results obtained in the previous section are already visually similar to the input sketch to some extent, however, it could further approximate to sketch shapes using deformation techniques. The framework supports partial deformation for separate facial components as well as the whole deformation for all facial components. In some cases, the search engine can not find the component very close to the input sketch shape, because the input sketch has exaggeration in a certain degree which does not exist in the realistic face shape. However, sometimes people want to perform exaggerations and meanwhile reduce unwanted distortion as much as possible. As a result, the MLS is suited for deformation [STJ06] with relatively sparse control points while at the same time keeping the original rigidness as much as possible(Figure 6). In fact, to achieve better realism, some constraints on deformation need to be considered additionally. It is noticed that the nose reflects more structure information, so inappropriate replacement might destroy the original face structure. So when the distance between nose shape exceeds an predefined threshold we prefer to retain the original one and just perform the deformation steps. On the contrary, other components such as eyes and brows have less face shape information, however they are very sensitive to distortion. As a result, we prefer searching matched components for composition to morphing these delicate components.

6. Results

In this paper, we build a library containing about 500 face images of people from different regions all over the world and labeled with corresponding attributes, component usabilities and feature points. All examples in this paper use one set of the following keywords: (1) female, young, Asian (2) male, young, European and (3) male, young, Asian, to build the corresponding sub-libraries. The input sketches are selected from the published paper [CXS’01] focusing...
Figure 6: Further Approximation to the sketch using MLS deformation. (a) Input sketch. (b) Results of directly combining Graph Cut results, which we denote as “Original”. In (c)-(h), we use “M”, “N”, “F” to represent the deformation of mouse, nose and face respectively, and use “+” to denote combination deformation. We use the same representation in Figure 7 and Figure 8.

Figure 7: Lifelike portrait synthesis examples. (b) and (c) are two sets of synthesis results for the same input sketch. The deformation representation is the same as Figure 6.

Figure 8: Examples of Lifelike portrait synthesis. The deformation representation is the same as Figure 6.

Figure 9: Face component swapping. From left to right, (a) the input face is replaced with (b) bigger eyes, (c) bigger eyes and thinner brows.

The system has several other applications besides lifelike portrait synthesis from sketch.

FACE COMPONENT SWAPPING The framework allows the user to replace facial components in an input portrait with the one with expected shapes, which we call “face component swapping”. Given the user desired shape of certain facial component, the system automatically searches matching images in the library and replaces the original one with the matching results. This can be used as a tool for face beautification (Figure 9).

FACE SHAPE TRANSFER Given an input face, the user can modify the shape of the face to make it thinner or longer by simply specifying the desired positions of several feature points. The framework searches the best match shapes in the face shape space and uses the matching result as the deformation destination in the Moving Least Square deformation method. This is also an interesting application for entertainment (Figure 10).

We implement the proposed framework on a PC with Intel Xeon 2.50GHZ CPU and 4GB memory. After building the library interactively, given several attribute keywords and an input sketch, a set of composition and deformation results can be generated in less than one minute.
7. Conclusions and future work

We have implemented a system for sketch realizing by synthesizing lifelike portrait from sketch. This framework is based on searching for facial components similar to the sketch in shapes, and then fusing the matching results followed by a process of deformation to further approximate to the sketch.

This paper mainly deals with the vanilla front face synthesis with no aberrations. However, as one important application of the proposed system is aiding forensic police in creating realistic faces of suspects. The ability to deal with some features such as beards, moles, hair as well as spectacles is necessary. To deal with hair or glasses, the idea of reconstructing partially damaged face images [HL03] is a good indication. To deal with face features such as beards, the image-based shaving algorithm [LCL08] can be utilized to remove and re-synthesize the beards. The same idea can be applied to small features such as moles. To decompose a face image into layers so as to model and manipulate each layer independently might be a promising solution to the above problems. Besides, the hair style is another important feature which is quite complicated and needs to be further investigated in the future.

Furthermore, several other aspects need to be considered in the future. Firstly, consider how to deal with occlusions and harsh lighting. Perhaps fine scale image inpainting and relighting can be exploited. Secondly, how to extend the realizing framework to cartoon style sketch to generate face images having both realistic and exaggerating properties can be further investigated in the future.

Acknowledgements

This work is supported by the Distinguished Young Scholars of NSFC, No.60525111 Scientific and Technological Innovation Nurture Funds Major Projects of MOE. P.R.C, No. 707005 Guangdong science and technology plan projects .No.2007A090601002

7.1. References


