A LEARNING-BASED SYSTEM FOR GENERATING EXAGGERATIVE CARICATURE FROM FACE IMAGES WITH EXPRESSION

Ting-Ting Yang and Shang-Hong Lai

Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan

ABSTRACT

In this paper, we propose a learning-based system for generating exaggerative caricatures with expression. Most of the previous works can only deal with frontal face images with neutral expression without glasses or hats, and are unable to apply more than one drawing prototype which was learned from the caricatures drawn by one single cartoonist at a time. The proposed caricature generation system exaggerates face images with expressions and learns the drawing prototypes from training data as well. Experimental results show that our system can capture the features selected by the artist and exaggerate them in similar ways.

Index Terms—Exaggerative caricature, facial expression, statistical learning

1. INTRODUCTION

A caricature is a comic picture that exaggerates the extraordinary characteristics of a person to make the portrait of a person more distinguishable from the other persons. Though the unique parts of the faces are amplified or exaggerated, the portraits of the persons are still recognizable. Some caricatured pictures attempt to maintain high fidelity, while others are cartoon-like portraits with high degrees of exaggeration. These appearances depend on the drawing styles of cartoonists.

Some previous works [5] [6] [12] can also create photorealistic images from real images without any exaggeration effect. Methods [1] [2] (used model [3]) [4] need considerable manual interactions to exaggerate the target images. The results generated by [11] [10] are usually unsatisfactory when the exaggerative rates are too large. The works [8] [9] exaggerated the textures by the analyzed facial shapes, but did not learn how to present different styles of exaggeration for different types of faces by a cartoonist. The system [7] (used [6] for transferring texture) learned the prototype of the cartoonist’s drawing style. However, it did not consider faces with beard, nevi or wrinkles.

In this paper, we propose a learning-based system for generating exaggerative caricatures with expression. This system is capable of learning the drawing style of artists by imitating their caricature works as the training data, and automatically creating exaggerative expressional caricatures from face images with expressions. The rest of this paper is organized as follows. Section 2 introduces the proposed system. Experimental results are given in section 3. Finally, section 4 concludes this paper.

2. PROPOSED METHOD

2.1. Caricature Model

We separate the representation of a face/caricature image \( F/F' \) into the shape \( S/S' \) and the texture \( T/T' \). Instead of modeling the relationship between \( F \) and \( F' \), we simplify the task to modeling the relationships between \( S \) and \( S' \) and between \( T \) and \( T' \). We call these processes shape exaggeration \( (S \rightarrow S') \) and texture transformation \( (T \rightarrow T') \). Combining these processes with the image warping, we can generate the final caricature.

Fig. 1: The face photo and its caricature.

2.2. Training Data

We collect facial images of 81 persons (45 females, 36 males) for training. For each person, we took his/her face images with happy, angry, and neutral expressions, and ask a cartoonist to draw the exaggerated caricature for each image. Then we manually labeled 87 feature points on both the original images and the exaggerated caricatures. These feature points are the shape points of eyebrows, eyes, nose, mouth, and facial contour, as shown in Figure 1. The persons in the photos may have different poses, but we can
see all the facial features clearly. However, we need to align all the face images under the same pose before the learning process. To align the face images, each training facial shape is first rotated to the horizontal axis by the two inner corner points of the eyes. Then, we scale the vertical length of the face to 128 pixels and apply the same scaling factor to the horizontal length. Finally we shift the face by moving the midpoint of the nostrils to the center of the image. The above normalization procedure will standardize all the face shapes to locate on the same origin, which is the center of the image.

We take the mean shape of all face images in the database as the ideal normal face. Thus, the shape of the original photo $S$ and the caricature $S'$ can be written as:

$$S = S_{mean} + \Delta S$$
$$S' = S + \Delta S'$$

We find that most people make asymmetric face with expression and the skewness is a key point of caricature. However, most people would like the eyes of the person to be about the same size in the portrait. As a result, it is best to consider the left and right eyebrows separately, while the left and right eyes are considered altogether as a whole. We partition the feature points into 6 parts: namely, eyes, left eyebrow, right eyebrow, nose, mouth and face contour. Each part of the face shape is denoted by $S_i$, where $i = 1, \ldots, 6$, as shown in Figure 2. At the same time, we choose 6 reference points to represent the positions of the partial shape $S_i$, as shown in Figure 3. The components and reference points of the caricature are denoted by $S'_i$, and $P'$. The relationship between $P$ and $P'$ can be written as:

$$P = P_{max} + \Delta P$$
$$P' = P + \Delta P'$$

Thus, the key to face exaggeration is to learn the relationships between $\Delta S_i$ and $\Delta S'_i$, as well as that between $\Delta P$ and $\Delta P'$.

Based on LPP, LPH [13] maps the low-dimensional data set to a high-dimensional data set based on RBF (Radial Basis Function). To find the mapping between $\Delta S$ and $\Delta S'$ ($\Delta P$ and $\Delta P'$), the procedure of LPH algorithm is given below:

Step 1) Eigen-decomposition: (skip this step while training neutral model)

Perform PCA on $\Delta S$ to get the eigenvectors $U$ and eigenvalues $V$.

Step 2) Finding KNN:

Compute Euclidean distances between any pair of data in the subspace spanned by $U$ (compute the distances in the original space while training neutral models). Select $k$ nearest neighbors for $x_i$.

Step 3) Setting the weight:

$$w_{ij} = \exp(-[x_i - x_j]^2 / 2\sigma^2)$$

if $x_i$ and $x_j$ are neighbors,

$$w_{ij} = 0$$

otherwise

Step 4) Computing transformation vectors:

Solve the generalized eigenvalue problem $\Delta S \delta a = \lambda SD \delta S a$,

where $D = \Sigma j \mu_j$, $L = D - W$. $\mu = \mu_1, \ldots, \mu_k$ are the eigenvectors corresponding to the minimal eigenvalues with $\lambda_1 \leq \ldots \leq \lambda_k$. The transformation vectors can be expressed as $A = US$ ($A = S$ while training neutral models). The low-dimensional data $y^p = A^T \Delta S$.

Step 5) Finding the matrix $W$ for mapping $y^p$ to $\Delta S'$ based on RBF:

To write the RBF in a matrix form, it can be denoted as

$$\Delta S' = W \Psi_{test}(y^p)$$

After alignment, the shape of the input testing face image $s_{in}$ can be projected by the transformation $A$.

$$y^p = A^T \Delta test_S$$

2.3. System Framework

Our system framework consists of the training phase and testing phase. In the training phase, we start with a set of images of selected expressions with manually labeled feature points. After the face alignment procedure, we train the shape exaggeration models. At the testing phase, we use the corresponding facial feature points to generate the exaggerated face shape and synthesize the unexaggerated gray scale sketch image. Combining the shape and the texture by using the RBF warping algorithm, we can obtain the final caricature with learned exaggeration. Figure 2 shows the procedures of the training and testing phases.

2.4. Shape Exaggeration
Since we have the intrinsic feature \( y_{\text{in}} \) and the matrix \( W \) and \( K \), the difference between the testing photo and its caricature-to-be \( \triangle \text{test}S'_i \) can be inferred by RBF:

\[
\triangle \text{test}S'_i = W_i K_{\text{RBF}}(y_{\text{in}}) \\
\text{test}S'_i = \text{test}S_i + \triangle \text{test}S'_i
\]

where the \( \text{test}S'_i \) (\( \text{test}P'_i \)) is the partial exaggerated feature shape (position). Then we combine all the feature parts \( \text{test}S'_i \) based on the corresponding reference point \( \text{test}P'_i \). After the combination, we obtain the desired exaggerated facial feature points.

2.5. Texture Transformation

In our texture transformation process, we aim to get the gray scale sketch with a comic look. Although the generation is easy, the resulting sketch looks acceptable. The algorithm is given below:

Input: \( \mathcal{S} \): feature shape of original photo \\
\( I \): the original photo

Output: \( \mathcal{C} \): the unexaggerated gray scale sketch

Step 1) Transform \( I \) into grayscale image \( \mathcal{I} \):

Step 2) Stretch intensity values of \( \mathcal{I} \) to \( 0 \sim 255 \).

Step 3) Perform Sobel edge detection on \( \mathcal{I} \) to obtain edge image \( \mathcal{E}_2 \). (\( r > 25 \))

Step 4) Perform Sobel edge detection on \( \mathcal{I} \) to obtain edge image \( \mathcal{E}_3 \). (\( r > 35 \))

Step 5) Paste the facial region in \( \mathcal{E}_2 \) to the corresponding region in \( \mathcal{E}_3 \).

Step 6) Fill the dark part and eyebrow regions in \( \mathcal{E}_3 \) with dark color.

Step 7) Fill in the inner mouth region with white color.

Step 8) Plot the contour lines. \( \mathcal{C} = \mathcal{E}_3 + \text{Contour Lines} \)

To make the facial area cleaner, we compute the ratio \( r \) of non-edge points and edge points. Higher ratio means higher demands of reducing edge points, which makes the image clearer. We take facial region with the result of higher threshold while filling in other region with the result of lower threshold as shown in step 4~5 in the algorithm. Filling the dark part in the image can make it easier to recognize and more artistic. Considering making the facial features stand out, we plot contour lines as the final procedure. Since we have the feature points, we can contour all facial features while fitting respectful points with a curve of degree 2.

2.6. Exaggeration Rate Adjustment

Since our training data may not exaggerate features as large as other cartoonists’ caricatures do, we use a scaling factor \( b \) to adjust the exaggeration rate to make the exaggeration of caricature more flexible as \( S' = S + b \triangle S' \). The caricatures may be unrecognizable when \( b \) is too large, as shown in Figure 4, thus the maximum \( b \) in our system is limited to 2. Our system allows the user to use different exaggerative rate interactively according to his/her taste.

2.7. Texture Mapping

To generate the caricature with clean contours, we would fill the eyebrows and inner mouth, and draw the contour lines after warping the sketches. We perform the RBF warping algorithm to warp the sketch from the original face shape to generate the exaggerated shape. After applying the RBF warping algorithm to the sketch without contour, the system will then fill the eyebrow regions and the inner mouth region. Adding the desired contour feature lines, we can obtain the final caricature.

![Generated caricatures with different exaggeration rate](image)

Fig. 4: Generated caricatures with different exaggeration rate: (a) \( b=0 \), (b) \( b=0.5 \), (c) \( b=1 \), (d) \( b=1.5 \), (e) \( b=2 \), (f) \( b=2.5 \), (g) \( b=3 \).

![Examples of caricature generation](image)

Fig. 5: Examples of caricature generation for a person with different expressions compared to the ground truth.
3. EXPERIMENTAL RESULTS

Since the amount of training data in our database may be not large enough, we employed the LOO (Leave-One-Out) approach to test the proposed system with the largest amount of training data from our database. We take 3 nearest neighbors in the training phases, and the exaggeration rate is set to 1 as the default value. The average execution time of training process is 0.2830 seconds, while testing process is 4.8441 seconds. The execution environment of the system is Intel Core2 CPU 2.4GHz, 2G RAM. Results generated under the default setting are shown in Figure 5. We can see that our system can capture some features selected by the artist and exaggerate them in similar ways.

Some unsatisfactory results by using the proposed system are depicted in Figure 6. It may not be able to generate the desired results when the face has extraordinary facial features or make very special expression due to the limitation of the training database.

To compare our results with the results of the previous method mentioned above, we use their pictures as testing data and manually labeled the feature points. Then we apply the neutral models trained by our database to the images. As shown in Figure 8, our system can apply more prototypes at a time. The facial characteristics, such as skin color and folds of eyelids, are able to reveal the characteristics of each individual.

4. CONCLUSIONS

We developed a learning-based caricature generation system that can generate caricature from a face image with facial expressions. The proposed system consists of three parts: namely, shape exaggeration, texture transformation, and texture mapping. Our system learns how the artist exaggerates facial features from the training data by the LPH algorithm so that our system can exaggerate some key facial features in a proper direction. With edge detection and the feature shape information, the proposed system can generate unexaggerated sketch in a grayscale image. Combining the exaggerated shape features with the sketch by RBF warping yields the final caricature. We show that the proposed caricature generation system can provide satisfactory caricatures for most face images. In the future, automatic feature point labeling and facial expression recognition will be included into our system to make it a fully automatic caricature generation system.

5. REFERENCES