Data-Driven Face Cartoon Stylization

Yong Zhang[1] Weiming Dong[1] * Oliver Deussen[2] Feiyue Huang[3] Ke Li[3] Bao-Gang Hu[1]
[1]NLPR-LIAMA, Institute of Automation, Chinese Academy of Sciences [2]University of Konstanz [3]Tencent

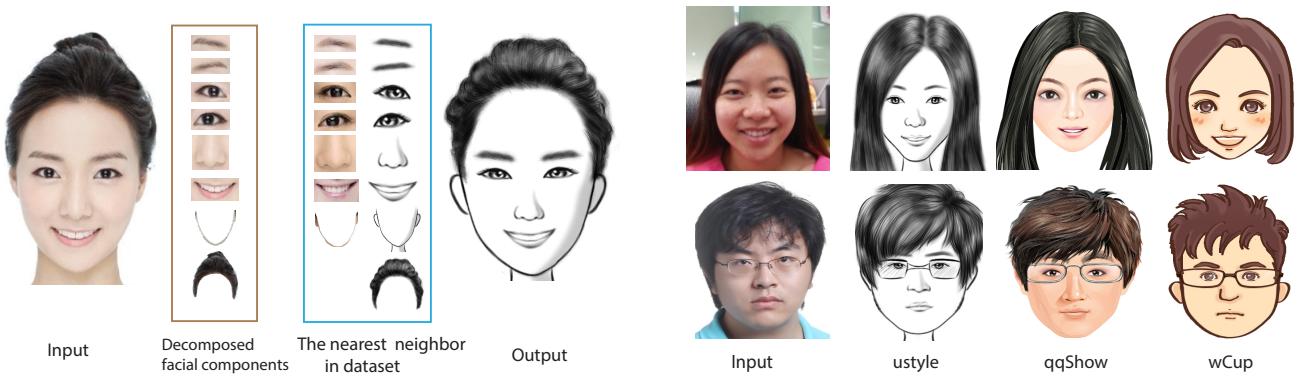

Figure 1: *Given a portrait image, we decompose the face into separated facial components and search for the corresponding cartoon components in a dataset by feature matching. The cartoon components are composed together to construct a cartoon face. We can easily generate cartoon faces of different styles and perform artistic beautification by using our framework.*

Abstract

This paper presents a data-driven framework for generating cartoon-like facial representations from a given portrait image. We solve our problem by an optimization that simultaneously considers a desired artistic style, image-cartoon relationships of facial components as well as automatic adjustment of the image composition. The stylization operation consists of two steps: a *face parsing* step to localize and extract facial components from the input image; a *cartoon generation* step to cartoonize the face according to the extracted information. The components of the cartoon are assembled from a database of stylized facial components. Quantifying the similarity between facial components of input and cartoon is done by image feature matching. We incorporate prior knowledge about photo-cartoon relationships and the optimal composition of cartoon facial components extracted from a set of cartoon faces to maintain a natural and attractive look of the results.

CR Categories:
I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms

Keywords: Face stylization, face parsing, face alignment

1 Introduction

Stylized cartoon faces are widely used as virtual personal images in social media such as instant chat, photo albums or twitter. However, manually creating such images needs artistic skills, it is often laborious and still might generate unwanted results. A user may have to search for suitable image editing operations to achieve a desired cartoon style and has to perform a large number of trials. Therefore, an automatic system for cartooning based on an input photography is very useful for many practical multimedia applications. Since humans look at faces very carefully, a good quality of the results is more crucial than for other objects. The cartoon faces should faithfully represent important facial features of the original photo. On the other hand, to beautify and to add artistic embellishment to the stylized representations is also challenging.

Chen et al. [2002] present the PicToon system to generate a personalized cartoon face from an input picture. The quality of their results highly depends on the accuracy of face sketch generation, while it is also difficult for their stroke-based rendering method to generate cartoon faces of different styles or to add artistic embellishment to specific facial components. Moreover, their example-based process requires face images and sketches to be strictly aligned, which makes the dataset construction to be complicated.

Patch-based methods have been widely applied to the synthesis of facial sketches due to their ability to represent local facial features [Wang et al. 2014]. Li et al. [2011] generate a cartoon image by incorporating the content of guidance images taken from a specific training set. Current state-of-the arts methods utilize Markov Random Fields (MRF) to select most appropriate neighbor patches to hallucinate a target patch [Zhou et al. 2012; Wang et al. 2013]. Each photo patch is an observation node and its corresponding sketch patch is the corresponding hidden node. The main drawback of these techniques is that these methods neglect the global shape information that describes the holistic geometric relationships between the individual facial features. Some important information about global shape exaggeration may be lost during face sketch synthesis. Boundary distortion and over-smoothing artifacts sometimes appear in the results since overlapping regions have to be averaged. In addition, high computational costs and memory loads also limit their practicability.

Hence, a desired approach should efficiently generate a clear and attractive cartoon face in a user-desired style. In this paper, we present an efficient data-driven framework for automatic face stylization based on portrait photographies. One of our major contribution lies in casting the problem of face cartoon stylization as an optimization problem that searches for a new composition of facial components that approximately match the facial features of the input face while still making the resulting sketch look natural and attractive. Our optimization method tries to balance between the shapes of local facial components, the global similarity to the input

*e-mail:weiming.dong@ia.ac.cn

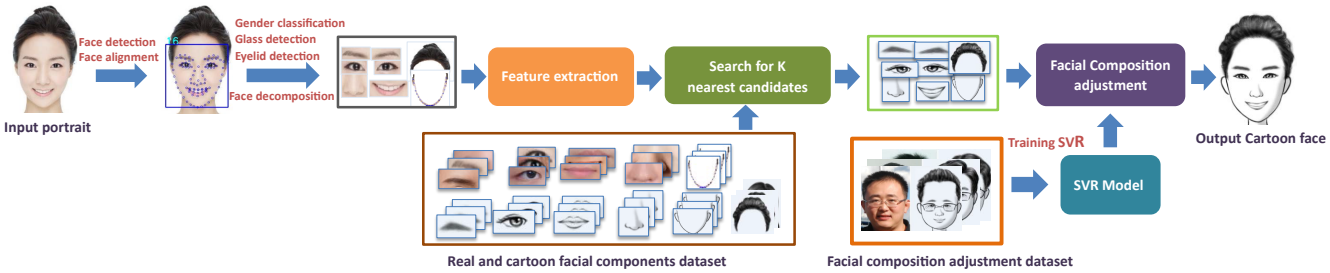

Figure 2: *The overall pipeline of our framework.*

face and the visual characteristics of a desired cartoon style. Another contribution is the quantification of the similarity between a real facial component and a cartoon representation. The pre-designed stylized facial components are first mapped to representative realistic facial components and similarities are computed as the distance between realistic and stylized representation. Compared with previous patch-based methods, our framework integrates both, the information of the local neighborhood and the configuration of the global shape exaggeration, which is important for achieving good face stylizations. On the other hand, our stylization process is more efficient than MRF-based schemes so that it is more suitable for mobile applications.

2 Overview

The framework of our system is illustrated in Figure 2. It consists of an offline phase and a runtime phase. Prior knowledge about the relationship between real and stylized facial components and optimal compositions of such components are obtained in the offline phase (Sec. 3). We further adopt ε-SVR (Support Vector Regression) [Chang and Lin 2011] to learn the rules of optimal composition. Since our cartoon style representation is on a component level rather than on a patch level, we use multiple stylized component sets for different styles for each realistic component. During runtime (Sec. 4), we first parse the input face into its semantic facial components. For each component, we extract its shape features and find the most similar real component in the dataset by feature matching. We then use the corresponding stylized facial components to compose the cartoon face. Finally, we automatically adjust the composition of the cartoon facial components to make the face more natural and attractive.

3 Data-Driven Prior Knowledge Extraction

Labeling of stylized facial components We start with building three datasets: a real face dataset, a real facial component dataset and a dataset of cartoon facial components. The real face dataset $\mathcal{P}$ consists of representative portrait photos downloaded from the Internet and contains 300 faces for male and 220 for female subjects. We carefully selected photos in order to ensure that there were enough different kinds of facial components for various shapes. We extracted all facial components from the faces in $\mathcal{P}$ to build the dataset of realistic facial components $\mathcal{F}_r$. We then picked representative components of 20 chins, 30 eyebrows, 30 eyes, 16 noses, 30 mouths and 75 types of hair for both male and female photographies from $\mathcal{F}_r$ and used them to build the dataset of stylized facial components $\mathcal{F}_c$ by letting an artist draw stylized versions for each representative realistic component. For each real eye we separately drew a stylized version of a single-fold eyelid and a double-fold eyelid. We then asked some users to compose cartoons for these faces in $\mathcal{P}$, by selecting and composing stylized facial components which are similar to the components on the realistic faces. Doing so, every facial component in $\mathcal{F}_r$ was labelled with the closest stylized facial component from $\mathcal{F}_c$.

Adjustment of facial compositions A good composition of facial components will make a cartoon face look natural and attractive. We adopt the ε-SVR method [Chang and Lin 2011] to let the system learn how to adjust the facial composition of a cartoon face in order to create more beautiful results. For the rough cartoon faces in $\mathcal{P}$, we ask an artist to adjust positions and sizes of each component in order to reach a good composition for each face.

A facial composition is defined as a feature vector extracted from the facial landmarks. Assuming the face is symmetric. We define a coordinate system is with its center between the eyes. The line passing through the eyes is the horizontal axis and the perpendicular line on the nose is the vertical axis. As shown in Figure 3, a facial composition is represented by a 13-dimensional vector $x \in R^{13}$, including coordinates and length of the left eyebrow, coordinates and length of the left eye, ordinate and width of the nose, ordinate and width of the mouth, ordinate and width or the cheek and ordinate of the chin.

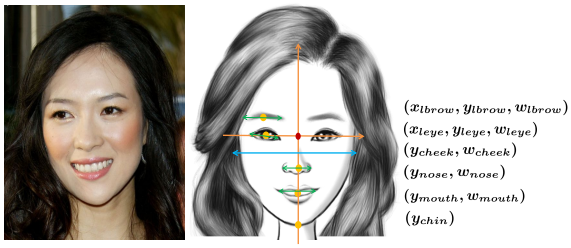

Figure 3: *Variables of a facial composition: a feature vector contains 13 dimensions and is extracted from both real faces and their abstracted counterparts that are inputs of the ε-SVR. $\boldsymbol{x}, \boldsymbol{y}$ are coordinates while $\boldsymbol{w}$ stands for width.*

After the artist finished composing the stylized facial components for each face in $\mathcal{P}$, we obtain a training set $\{(\boldsymbol{x_1}, \boldsymbol{z_1}), ..., (\boldsymbol{x_l}, \boldsymbol{z_l})\}$, where $\boldsymbol{x_i}$ and $\boldsymbol{z_i}$ are feature vectors for each real face and its stylized version. In order to compose a cartoon face automatically, each dimension of its facial composition needs to be predicted. So we train the ε-SVR as a predictor for each dimension with a sub-dataset, $\{(\boldsymbol{x_1}, z_{1_k}), ..., (\boldsymbol{x_l}, z_{l_k})\}$, where $z_{i_k} \in R^1$ is the k-th dimension of $\boldsymbol{z_i}$. Under given parameters $C > 0$ and $\varepsilon > 0$, the form of ε-SVR is

$$\begin{aligned} \min_{\boldsymbol{w},b,\xi,\xi^*} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l}\xi_i + C\sum_{i=1}^{l}\xi^* \\ s.t. \quad & \boldsymbol{w}^T\phi(\boldsymbol{x_i}) + b - z_{i_k} \leq \varepsilon + \xi, \\ & z_{i_k} - \boldsymbol{w}^T\phi(\boldsymbol{x_i}) - b \leq \varepsilon + \xi^*, \\ & \xi, \xi^* \geq 0, i = 1, ..., l, \end{aligned} \tag{1}$$

where $\boldsymbol{w}$ is the weighting vector for the input that is needed to be learned by solving the optimization problem. $\phi(\boldsymbol{x})$ is a transform function. While it is difficult to compute the primal problem, its

Lagrangian dual problem

$$\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}*} \frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)^T K(\boldsymbol{x_i},\boldsymbol{x_j})(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) + \varepsilon\sum_{i=1}^{l}(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*) + \sum_{i=1}^{l} z_{i_k}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) \quad (2)$$
$$s.t.\ \ \boldsymbol{e^T}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) = 0,$$
$$0 \le \boldsymbol{\alpha},\boldsymbol{\alpha}^* \le C, i = i,...,l,$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$ are Lagarange multipliers, can be solved easily. An RBF kernel $K(\boldsymbol{x_i},\boldsymbol{x_j}) = \phi(\boldsymbol{x_i})\phi(\boldsymbol{x_j}) = exp\{-\frac{(\boldsymbol{x_i}-\boldsymbol{x_j^2})}{2\delta^2}\}$ is used for this purpose. For different given parameters δ, C and ε, the solution of the dual problem differs. We carry out a grid search method to search for parameters that minimize the training error of each predictor. When $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}*$ are obtained, the form of the prediction function of ε-SVR is

$$z_k = \sum_{i=1}^{l}(-\boldsymbol{\alpha_i}+\boldsymbol{\alpha_i^*})K(\boldsymbol{x_i},\boldsymbol{x}) + b. \quad (3)$$

We can now predict the cartoon facial composition when given a realistic facial composition once training procedure for all predictors are completed. Then facial composition is then applied to compose the stylized facial components.

4 Face Cartoon Stylization

Face detection and alignment We formulate the face cartoonization as an optimization problem, which tries to strike a balance between similarity to the input face and naturalness/attractiveness of the cartoon representation. The user starts a cartoonization session by creating a portrait photo as the input for the system. We use the method in [Viola and Jones 2004] for face detection and choose the first detected face in the image as the input face. We use the face alignment algorithm in [Cao et al. 2014] to locate the facial landmarks. We employ the commonly used 88 facial landmarks to represent a face shape and decompose the input face into separate components such as eyes, eyebrows, nose, mouth and chin. The regions of eyes, eyebrows and nose are extracted by using the bounding boxes of their landmarks. The mouth and chin are represented directly by their landmarks (21 landmarks for chin and 22 for mouth). Then, for each component, we search for the most similar component in the dataset of real facial components $\mathcal{F}_r$.

Eyes and nose The eyelid shape is a strong feature to distinguish one eye from others. We use HoG descriptors [Dalal and Triggs 2005] to extract image features for eyes. A 2304 dimensional feature vector is obtained and the Euclidean distance is used for computing distances between face descriptions. We use the same matching scheme for finding the most similar nose from $\mathcal{F}_r$. We normalize the input images for eyes and the eyes in $\mathcal{F}_r$ to 62×100, and for noses is 71×200.

Eyebrows Since there is usually no clear boundary between an eyebrow and its surrounding skin, we first perform a Gaussian normalization to the eyebrow region. Then, we partition the normalized region into 6×20 patches and sum the pixel intensities of each patch. Finally, all summations over small patches are connected to a vector regarded as the feature vector of the eyebrow. We normalize the input eyebrows and the eyebrows in $\mathcal{F}_r$ to 66×200.

Chin and mouth Using the shape information, the matching of chins and that of mouths achieve good results. Two chosen landmarks of a chin are aligned to two fixed points. The aligned landmarks are combined to a describing vector which is the feature vector for a chin or mouth.

Once feature extraction is finished, we search the K nearest neighbors from the dataset for each facial component. Since each realistic component is connected to a cartoon representation, we get K stylized components for each facial component. These components may share two or more cartoon components in the search result. We choose the one which occurs most frequently to be the best match.

Figure 4: *Hair matching with a clustering method on image color histogram. Left: an aligned face image and the clustering result. The hair region is marked in green. Right: the best matching cartoon hair and its binary image.*

Hair Hair matching consists of two steps: local and global hair matching. The goal of local hair matching is to find cartoon hairs with similar fringe to the real hair while global hair matching aims to search for hair with similar global shape. In the first step, a patch T_r is extracted from a face image I_r around eyes including the fringe. T_r is then converted into a binary image and scaled to a fixed size, T_{rb}. The same operation is carried out on all cartoon hair forms $\mathcal{S} = \{I_c\}$ in order to get a set of binary images $\mathcal{T} = \{T_{cb}\}$. Each binary image is partitioned into small patches with the same size. We sum the intensities over the pixels on each patch and combine the summations to a feature vector. Therefore, for a given input T_r we get the N nearest neighbors and the corresponding cartoon hair shapes $\mathcal{C} = \{I_c^1,...,I_c^N\} \in \mathcal{S}$. In the second step, we use a clustering method on the color histogram of I_r to segment it into several connected regions (Figure 4). Then I_r is converted into a binary image and scaled to a fixed size I_{rb} by keeping the hair region. Cartoon hair shapes in $\mathcal{C}$ are converted into binary images $\mathcal{C}_\mathcal{B} = \{I_{cb}^1,...,I_{cb}^N\}$ in the same way. HoG descriptors are used to extract features from these binary images. So we can find the nearest neighbor in $\mathcal{C}_\mathcal{B}$ and the corresponding cartoon hair in $\mathcal{C}$ that is regarded as the best match.

Gender classification A systematic study on gender classification [Makinen and Raisamo 2008] compared results of a few learning-based methods. We use a *C-Support Vector Classification* (*C-SVC*) with image pixels as input which achieved better results on average than other methods in that paper.

Glass detection Glass detection is operated on a small central patch between two eyes. We convert the patch to a gray patch and calculate vertical gradients. The two maximum gradients of pixels in each column are obtained (see Figure 5). The upper one and the lower one are collected in two sets of pixels and their ordinates in the patch are denoted by $a_i \in \mathcal{A}, b_i \in \mathcal{B}, i = 1,2,...,n$. Then we compute variations $var(\mathcal{A})$ and $var(\mathcal{B})$ of the pixels' ordinates in each set:

$$var(\mathcal{A}) = \frac{1}{n-1}\sum_{i=1}^{n}(a_i - \mu_\mathcal{A})^2, \quad (4)$$

$$var(\mathcal{B}) = \frac{1}{n-1}\sum_{i=1}^{n}(b_i - \mu_\mathcal{B})^2. \quad (5)$$

There exists a glass if summation of $var(\mathcal{A})$ and $var(\mathcal{B})$ is smaller then a threshold. The glass color is obtained from the region between two set of pixels.

Eyelid detection The type of the eyelid (double or single) is a distinctive feature for eyes. We use canny edge detection to get a binary image for left and right eye. The left eye is double eyelid

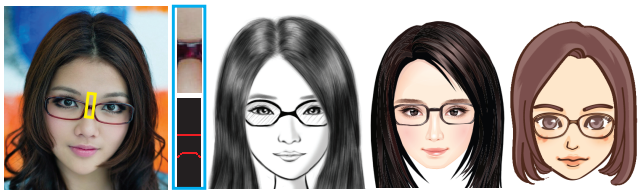

Figure 5: *Glass detection. A patch between two eyes is extracted for glass detection. Two pixels with the two largest gradients in each column are drew in red and others in black.*

if two pulses are detected on the left-top part of its gradient image. The right-top part is detected for the right eye.

Cartoon composition Stylized components are composed to the cartoon face by referencing to the cartoon facial composition $\boldsymbol{z}$. Firstly, a coordinate system is built on the canvas (Figure 3). Secondly, using information in $\boldsymbol{z}$ we can get the position and width of each cartoon component. Components are moved to their target positions and scaled to their target width by preserving their aspect ratios. Finally, alpha blending is adopted to compose all components to get a cartoon face.

Warping For some input faces, there may not exist suitable stylized facial components, so the cartoon faces will look dissimilar to the real faces. In our experiments, we find that the shapes of chin and eyes are important factors which will apparently affect the similarity of the cartoon face to the input face. To address this problem, we employ the warping method in [Schaefer et al. 2006] to change the shapes of the cartoon chin and cartoon eyes which are acquired by feature matching. This method needs two set of control points, one as source points and the other as target points , which is convenient to warp an image to the desired one. However, some distortions occur in the results. Experiments show that the number and homogeneity of control points as well as the smoothness of the curve have effects on the result. The smoothness plays the most important role. So we sample the control points equidistantly on the curve and smooth new control points to figure out the problem. We smooth the curve by using a Gaussian filter with the scale factor to control the smoothing extent.

5 Results

We connected faces in the dataset to three types of stylizations named *ustyle*, *qqShow* and *wCup* respectively. The framework was applied to synthesize three kinds of stylized faces. It runs under PC of 3.4GHz Intel Core i7 CUP and 16GB DDR3 memory. It took 0.8 seconds on average to synthesize a face. We also develop a mobile application product on both Android and IOS platforms. Figure 6 shows an example generated by our mobile software.

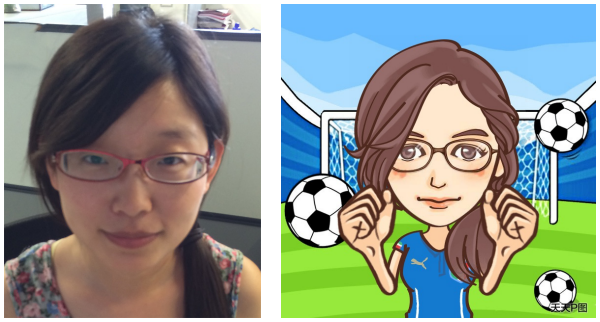

Figure 6: *Cartoon image generated by our mobile software.*

Figures 1, 3 and 5 show some results obtained using our framework for the three style materials. There are few distortions in results. Synthetic faces are not equal to real faces but keep some of their main features. Adjusted facial compositions for cartoon stylizations fit the original ones. Our framework has the ability to generalize to another face style as long as faces are connected to corresponding materials in the dataset. Note that for "wCup" style, artistic exaggeration is performed to the cartoon materials by increasing the sizes of eyes. Thus, the cartoon faces of this style all have lovely big eyes. This kind of art forms can be easily integrated into our framework while difficult for patch-based methods. The limitation of our framework is some cartoon faces may be not very similar to the original faces due to the finite facial components in the database.

6 Conclusions and Future Work

We have presented a data-driven framework for generating cartoon faces from portrait photographies that show a desired cartoon style. In the future work we will study how to handle more facial details during the stylization operation, such as nevus, wrinkles and cheekbones, and how to make stylized faces more lively. Our system currently can only generate cartoon faces of front views. How to handle input faces with apparent 3D rotation and generating cartoon images for such views is another challenging direction.

Acknowledgements This research was partially funded by National Natural Science Foundation of China (Nos. 61172104, 61273196, 61271430, 61201402, 61372184, 61372168, and 61331018), and CASIA-Tencent BestImage joint research project.

References

CAO, X., WEI, Y., WEN, F., AND SUN, J. 2014. Face alignment by explicit shape regression. *International Journal of Computer Vision 107*, 2, 177–190.

CHANG, C.-C., AND LIN, C.-J. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology 2*, 3, 27.

CHEN, H., ZHENG, N.-N., LIANG, L., LI, Y., XU, Y.-Q., AND SHUM, H.-Y. 2002. Pictoon: A personalized image-based cartoon system. In *Proceedings of ACM Multimedia*, ACM, New York, NY, USA, 171–178.

DALAL, N., AND TRIGGS, B. 2005. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, IEEE, 886–893.

LI, H., LIU, G., AND NGAN, K. N. 2011. Guided face cartoon synthesis. *IEEE Transactions on Multimedia 13*, 6, 1230–1239.

MAKINEN, E., AND RAISAMO, R. 2008. Evaluation of gender classification methods with automatically detected and aligned faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 3 (March), 541–547.

SCHAEFER, S., MCPHAIL, T., AND WARREN, J. 2006. Image deformation using moving least squares. *ACM Transactions on Graphics 25*, 3 (July), 533–540.

VIOLA, P., AND JONES, M. J. 2004. Robust real-time face detection. *International Journal of Computer Vision 57*, 2 (May), 137–154.

WANG, N., TAO, D., GAO, X., LI, X., AND LI, J. 2013. Transductive face sketch-photo synthesis. *IEEE Transactions on Neural Networks and Learning Systems 24*, 9, 1364–1376.

WANG, N., TAO, D., GAO, X., LI, X., AND LI, J. 2014. A comprehensive survey to face hallucination. *International Journal of Computer Vision 106*, 1, 9–30.

ZHOU, H., KUANG, Z., AND WONG, K.-Y. 2012. Markov weight fields for face sketch synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 1091–1097.