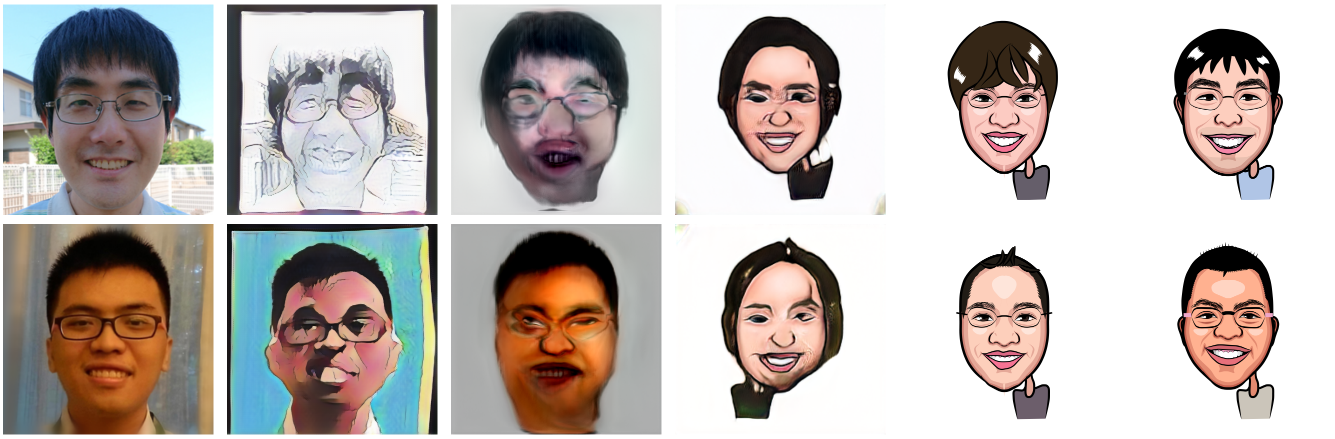
Pacific Graphics (2023)
R. Chaine, Z. Deng, and M. H. Kim (Editors)

Short Paper

Automatic Vector Caricature via Face Parametrization

Koki Madono[1], Yannick Hold-Geoffroy[1], Yijun Li[1], Daichi Ito[1], Jose Echevarria[1], Cameron Smith[1],

[1] Adobe Research



(a) Input (b) WGAN [LPM*19] (c) SGAN [KAH*20] (d) EWC [LZLS20] (e) Ours (f) Artist

Figure 1: *We propose Parametric Caricature, the first parametric-based caricature generation for a given face photo (a) that yields vectorized and animatable caricatures (e), which is fundamentally different from pixel-based results obtained by Image-to-Image Translation methods (b-d). Examples of our dataset created by artist are shown in (f).*

Abstract

Automatic caricature generation is a challenging task that aims to emphasize the subject's facial characteristics while preserving its identity. Due to the complexity of the task, caricatures could exclusively be performed by a trained artist. Recent developments in deep learning have achieved promising results in capturing artistic styles. Despite the success, current methods still struggle to accurately capture the whimsical aspect of caricatures while preserving identity. In this work, we propose Parametric Caricature, the first parametric-based caricature generation that yields vectorized and animatable caricatures. We devise several hundred parameters to encode facial traits, which our method directly predicts instead of estimating the raster caricature like previous methods. To guide the attention of the method, we segment the different parts of the face and retrieve the most similar parts from an artist-made database of caricatures. Our method proposes visually appealing caricatures more adapted to use as avatars than existing methods, as demonstrated by our user study.

CCS Concepts

• ***Computing methodologies*** → *Artistic Image Generation;* • ***Image Rendering*** → *Image Vectorization;*

1. Introduction

As Social Networking Services (SNS) became an integral part of everyday life, there is a surge of requests for avatar systems that preserve users' identity (e.g. "Facebook Avatar" [Fav], "Animoji" [Ani]). Caricatures are used in various media to capture our attention, from news commentaries to virtual avatars for marketing, to name a few. Contrarily to randomly distorted funny faces, caricature is an art form that emphasizes the subject's facial characteristics while preserving its identity. Mastering caricatures takes practice and great artistic talent due to stylization introducing a large appearance gap between photos and caricatures; as a result, it is impractical to mass-produce them from scratch. To accelerate this creative process, a typical method is to propose presets comprised of multiple facial parts that a user can select to build the desired facial appearance [ZZ13]. However, despite the speedup obtained by this method, it is still quite time-consuming, and the resulting quality depends on the assets, which can be challenging to obtain.

Several methods were recently developed to automatically cre-

© 2023 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.
This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

DOI: 10.2312/pg.20231271

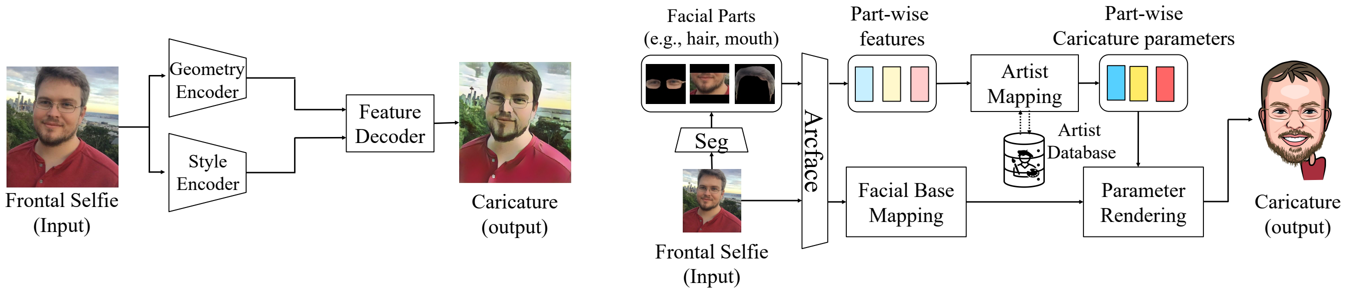


Figure 2: ***Previous methods (Left)*** *typically treat facial feature amplification and stylization separately to obtain a raster image as output.* ***Our method (Right)*** *first extracts facial features from both the frontal selfie and the individual facial parts. The extracted features are used to map to a collection of annotations from an artist and used to encode other caricature parameters in the facial base mapping. Then, the outputs are decoded to generate a parametric representation of the face, which we finally render.*

ate caricatures with the goal of helping artists by providing an initial caricature canvas, as illustrated by Fig. 2-left. In essence, existing techniques first estimate geometric distortions to apply to the image, effectively exaggerating facial features. Second, they apply a style transfer method to the distorted image to simulate pencil strokes or any other desired texture. Most recent approaches cast this task as an image-to-image translation problem and consist of GAN-based [CLL18] and few-shot-based methods [GHGL20, OLL*21]. To train these methods, a large-scale dataset [HLS*17] comprising multiple different—albeit conflicting—artists styles have been proposed. Notwithstanding the visually pleasing results proposed by those methods, they usually propose a single static raster image as an output that can bear high sensor noise and texture artifacts.

In this paper, we propose Parametric Caricature, the first parametric-based caricature generation that yields vectorized and animatable caricatures. Instead of relying on raster images, we make the observation that vector lines better convey the visual appeal of the artist's strokes. However, directly regressing the control points of a vector curve from portraits is too ill-posed to train directly using a CNN, especially when working with a limited dataset produced by a single artist. To solve this problem, we propose a novel parametric representation of the human face that encodes principal facial traits such as the position and sizes of the eyes, nose, mouth, etc. Figure 2-right illustrates our proposed approach which applies a desired artistic style to an input face. Our pipeline first leverages a pre-trained encoder (Arcface [DGXZ19]) to extract features from the face and its segmentation, which are then mapped to a collection of artist's annotations. Finally, a decoder uses the features and geometric cues to regress our parametric representation. Our approach estimates a parametric representation of the subject face, which we can render to a final raster image. This process creates images devoid of sensor noise and artifacts, and are easily animated.

We summarize our contributions as follows:

- A parametric representation that encodes facial characteristics, accurately capturing the subject identity and which is straightforward to animate;
- An automatic vector-based caricature generation method that works on regular portrait images;
- A user study that compares our method against other state-of-the-art caricature generation methods.

2. Related Works

Image-to-Image Translation Multiple works have learned and applied domain transforms on images. Pix2pix [IZZE17] was proposed to transform global domain information to the target domain. CycleGAN [ZPIE17] proposes an extension of pix2pix that trains bidirectional domain transformation for stable training. Since these works do not faithfully preserve the semantic information while transforming, MUNIT [HLBK18] was later proposed to improve the output diversity by decomposing the latent space in two: a domain-invariant content code, and a style code that captures domain-specific properties. Considering how straightforward image-to-image transformations apply to our task, we use these methods as baselines to compare against and highlight the importance of the independent part parametrization in our method.

Few Shot Learning Few-shot image generation aims instead to hallucinate new and diverse examples while preventing overfitting to the few training images. Existing work mainly follows an adaptation pipeline, in which a base model is pre-trained on a large source domain and then adapted to a smaller target domain. A recent work [OLL*21] shows that such a mapping can be obtained only from 10 images. However, most of these methods have difficulty capturing the geometric warping accurately. In this work, we deal with this problem by learning the mapping to each caricature parameter using a pre-trained facial recognition model and similar parts retrieval.

Automatic Caricature Generation Automatic or guided caricaturization has been investigated for many decades using rule-based methods [BG85, GRG04, LL04, MLN04]. While providing interesting insights into producing well-formed caricatures, these methods show poor generalization to new faces and artists' styles. Recently, deep learning-based methods [CHT*20, CHT*19, SDJ19, GHGL20, JJJ*21, HHW*21, YXS*21, HLC*22, YJLL22] were proposed to perform caricatures that produce impressive and visually pleasing results. To achieve this, one type of approach consists of distorting the portrait to amplify distinguishable facial traits and enhance the salient face characteristics. For example, Semantic-

© 2023 The Authors.

CariGANs [CHT*20, CHT*19] aims at transferring shape from a source portrait to a target reference while retaining the subject identity. WarpGAN [SDJ19] explicitly estimates a geometrical transformation to apply to the portrait. Specifically, the method estimates the location of control points within the portrait and their displacement to warp the face into a caricature, which is then stylized using a GAN. Similarly, AutoToon [GHGL20] proposes to directly learn a dense warping from a few learning samples by a single artist, simplifying the process.

A second approach consists of treating caricaturization as a specialized image-to-image problem, typically solved using either fully convolutional models or GANs. In this vein, CartoonGAN [CLL18] is the first to propose a GAN-based approach to directly generate caricatures from an input image, without explicit warping involved. The method learns how to perform this using several thousand unpaired images of both portraits and cartoons.

Commercial systems for automatic avatar generation have recently appeared. Google trained a model using a dataset of illustrated 2D parts created by artists and voted by human raters [all]. Bitmoji can also generate 2D avatars using a library of parts [bit]. Nintendo has a related avatar system where parts are 3D [mii]. Unfortunately, not many details are publicly available about these approaches. Our work regresses richer and more detailed facial parameters, using a vector representation, which can be rendered to obtain clean easy-to-animate resolution-independent avatars.

3. Dataset

In this section, we first detail the parameter system we devised to capture facial traits, followed by a description of our caricature dataset.

3.1. Caricature Parameter Representation

Our set of parameters was designed by an artist with the goal of being expressive enough to capture different caricature styles. In essence, these parameters represent information akin to brush strokes, which we can render to obtain a raster image. Basically, caricature parameters are categorized with the following three information: 1) Geometry, 2) conditional parts (boolean), and 3) color information.

Geometry information indicates overall geometry information such as stroke angle, coordinates, volume, etc. When rendering, the key points are connected together with κ-curves [YSW*17], which are Bézier-like curves that yield nice properties such as curvature smoothness and prevent undesired effects such as loops and cusps.

Conditional parts include optional features such as eyeglasses, beard, and hair, and allow for easy editing and addition of parts to the caricature.

Color information consists of the brushstroke color for the different parts of the drawing, such as skin color and tone as well as highlights and shadows.

In total, our parametric system holds 629 parameters, each of which has a default value yielding a generic, neutral face. When manually creating or editing a caricature using our parametric system, the user can focus on the parameters related to the salient face characteristics, while the other default parameters will automatically produce generic facial features. Please refer to our supplementary material for a full description of each of our parameters.

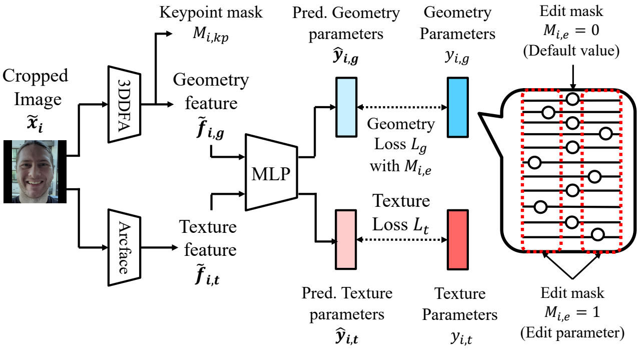


Figure 3: *Training Phase of facial characteristics extraction. We first extract the features from an aligned image $\tilde{\boldsymbol{x}}_i$. Using both the geometry $\tilde{\boldsymbol{f}}_{i,g}$ and texture $\tilde{\boldsymbol{f}}_{i,t}$ features, our model regresses both the geometry $\hat{y}_{i,g}$ and texture $\hat{y}_{i,t}$ parameters. We use a mask M_i as an attention mechanism to focus on the relevant geometry parameters related to the facial part being analyzed. These mask values represent the parameters that were manually changed by the artist from the default value to fit the portrait identity when building the dataset.*

3.2. Caricature dataset

We collected the frontal portraits of 245 people with their consent, spanning a broad range of age groups, genders, skin colors, and face shapes.[†] Based on these photos, an experienced caricature artist made caricatures using our parameters-based system. Since all caricatures were performed by a single artist, their consistent style makes them ideal as ground-truth caricature data to train our method. Each data triplet consists of a real portrait image X_{in}, a caricature image X_{cari}, and a caricature parameter p_{cari}. This paired dataset of 245 triplets (X_{in}, X_{cari}, p_{cari}) was split into 200 training and 45 test images.

To increase the diversity of the data, our artist made 10 types of parameter augmentation scripts, which enabled us to automatically extend our data tenfold to 2450 images, which we split in 2000 training and 450 test images. Please refer to the supplementary material for the details of our data augmentation.

4. Proposed Approach

We now present our proposed caricature image generation method, an overview of which is shown in Fig. 2-right. Our pipeline consists of two parts: facial characteristics extraction and artist mapping. This first part captures the general appearance and identity of the subject, while our artist mapping specializes in conditional parts such as hair, beard, and eyeglasses.

[†] We obtained the permissions from each person in the dataset to train a machine learning model on their portraits, and to display them in this paper.

© 2023 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

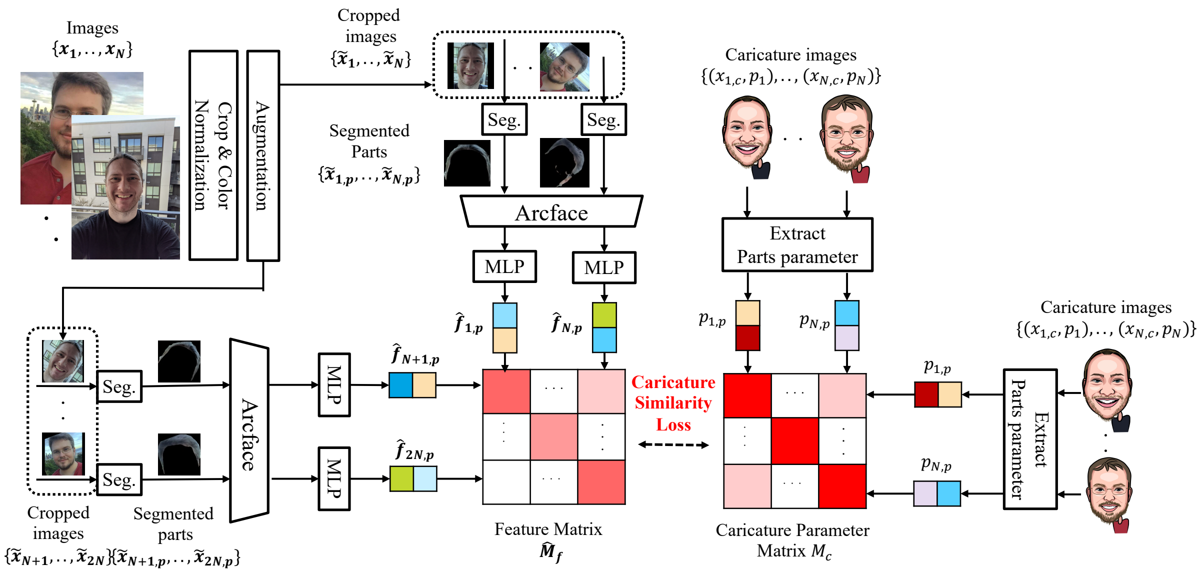

Images $\{x_1,..,x_N\}$
Crop & Color Normalization
Augmentation
Cropped images $\{\tilde{x}_1,..,\tilde{x}_N\}$
Seg.
Seg.
Segmented Parts $\{\tilde{x}_{1,p},..,\tilde{x}_{N,p}\}$
Arcface
MLP
MLP
$\hat{f}_{1,p}$
$\hat{f}_{N,p}$
Caricature images $\{(x_{1,c},p_1),..,(x_{N,c},p_N)\}$
Extract Parts parameter
$p_{1,p}$
$p_{N,p}$
Caricature images $\{(x_{1,c},p_1),..,(x_{N,c},p_N)\}$
Seg.
Seg.
Arcface
MLP
MLP
$\hat{f}_{N+1,p}$
$\hat{f}_{2N,p}$
Caricature Similarity Loss
$p_{1,p}$
$p_{N,p}$
Extract Parts parameter
Cropped images $\{\tilde{x}_{N+1},..,\tilde{x}_{2N}\}$
Segmented parts $\{\tilde{x}_{N+1,p},..,\tilde{x}_{2N,p}\}$
Feature Matrix $\hat{M}_f$
Caricature Parameter Matrix M_c

Figure 4: ***Training phase**: For each image, we make two augmented images, $\{\tilde{\boldsymbol{x}}_1,..,\tilde{\boldsymbol{x}}_N\}$ and $\{\tilde{\boldsymbol{x}}_{N+1},..,\tilde{\boldsymbol{x}}_{2N}\}$. Then, we segment the part region (This figure shows an example of a hair region). The cropped part $\{\tilde{\boldsymbol{x}}_{1,p},..,\tilde{\boldsymbol{x}}_{N,p}\}$ and $\{\tilde{\boldsymbol{x}}_{N+1,p},..,\tilde{\boldsymbol{x}}_{2N,p}\}$ are feeded into the Arcface to obtain the feature. To project the feature, we use MLP and obtain the feature $\{\boldsymbol{f}_{1,p},..,\boldsymbol{f}_{N,p}\}$ and $\{\boldsymbol{f}_{N+1,p},..,\boldsymbol{f}_{2N,p}\}$. Using this features, we construct the feature matrix $\hat{\boldsymbol{M}}_f$ and minimize with caricature parameter matrix M_c.*

In facial characteristics extraction and artist mapping, we prepare one MLP with regression. The MLP output total of 337 parameters (223 for geometry, 114 for texture). For artist mapping, we prepared three MLPs (hair, eyeglasses, and facial hair). Each output dimension is 151, 29, and 39 respectively. Of the rest of the proposed caricature parameters, we do not use them since it is not face-related parameters such as system setting parameters.

4.1. Facial characteristics extraction

Figure 3 illustrates our facial characteristics extraction step. We first extract the feature vectors from an image $\tilde{\boldsymbol{x}}_i$ that was cropped and aligned beforehand using the same process as [KLA19]. Using 3DDFA [GZY*20, GZL18], we extract geometry cues $\tilde{\boldsymbol{f}}_{i,g}$ that contains a dense keypoints $\tilde{\boldsymbol{f}}_{i,g,kp}$, expressions $\tilde{\boldsymbol{f}}_{i,g,e}$, and camera angle $\tilde{\boldsymbol{f}}_{i,g,c}$. In addition, we extract a texture feature $\tilde{\boldsymbol{f}}_{i,t}$ using Arcface [DGXZ19]. Using these features, we regress both geometry and texture parameters.

4.1.0.1. Optimization with Masking Since our parameters are local, meaning that each parameter only impacts a single brushstroke, we employ an attention mechanism to focus the method on the relevant facial regions during training. This is done using the mask M to guide the attention of the multi-layer perceptron (MLP) to the relevant region of the image. This helps the network converge with the limited amount of data we have. Since the mask is taken on the final output, the mask dimension is the same as the output

To encourage the model to look at both key points and Arcface features, we use $M_{i,kp}$ to represent a keypoint detection in the i-th image as

$$M_{i,kp} = \begin{cases} 1, & \text{keypoints detected in the } i\text{-th image} \\ 0, & \text{otherwise} \end{cases} . \tag{1}$$

This mask is derived from the output of Arcface.

To steer the attention of the model towards the artist's caricature parameters, we use an edit mask. This edit mask consists of all the parameters that were edited from the default values by the artist when creating the dataset. We define the edit mask $M_{i,e}$ in the i-th image of j-th parameter as

$$M_{i,e,j} = \begin{cases} 1, & i\text{-th image of } j\text{-th parameter} \\ 0, & \text{otherwise} \end{cases} . \tag{2}$$

As for the geometry loss, we employ both keypoint mask $M_{i,e}$ and the edit mask $\hat{y}_{i,g}$ as follows.

$$L_g = \mathrm{SL1}(M_{i,kpe}\, \hat{y}_{i,g},\ M_{i,kpe}\, y_{i,g}) + \lambda_g \|M_{i,kpe}\, \hat{y}_{i,g}\| , \tag{3}$$

where SL1 indicate Smooth L1 loss [Gir15] the $M_{i,kpe}$ is the multiplied matrix of $M_{i,kp}$ and $M_{i,e}$, $\hat{y}_{i,g}$ is the predicted geometry parameters, and $y_{i,g}$ denotes the geometry parameters, all of them from the i-th image.

For the texture loss, we only use a keypoint mask $M_{i,e}$ for the loss function as follows

$$L_t = \mathrm{SL1}(M_{i,kp}\hat{y}_{i,t},\ M_{i,kp}y_{i,t}), \tag{4}$$

where $\hat{y}_{i,t}$ is the predicted texture parameters, and $y_{i,t}$ denotes the texture parameters, all of them from the i-th image.

Finally, we train our facial characteristic extractor by summing

© 2023 The Authors.

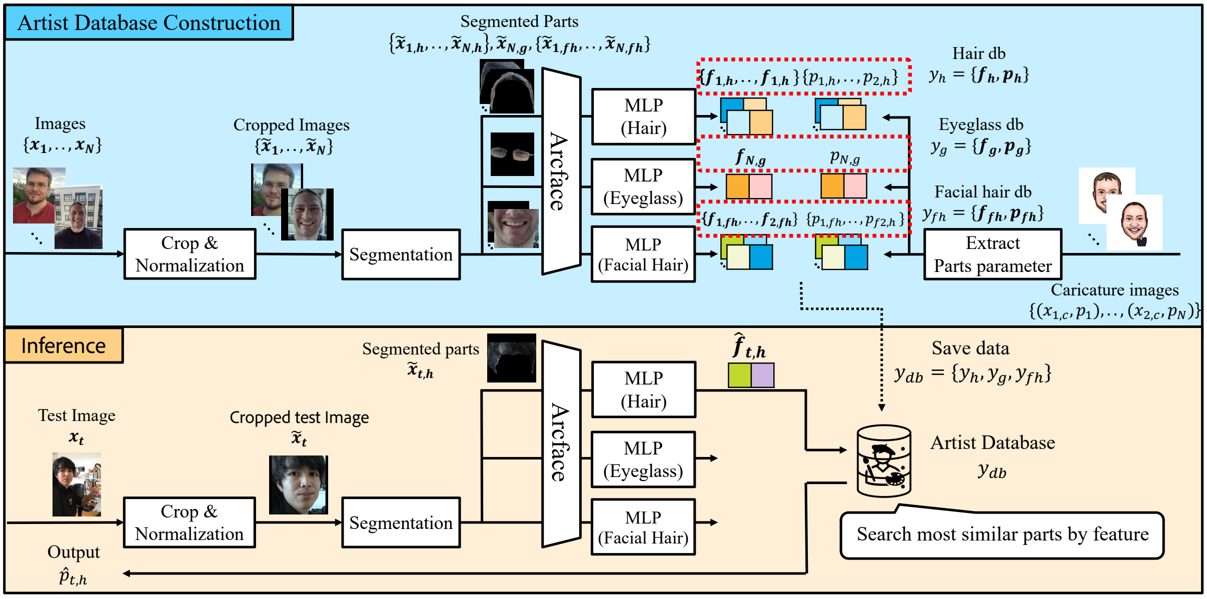


Figure 5: ***Artist Database construction phase****: We segment each conditional facial part (hair, eyeglasses, and facial hair) from each image in the dataset $\{\tilde{\boldsymbol{x}}_1, .., \tilde{\boldsymbol{x}}_N\}$. Note that the number of segmented parts can vary per subject (e.g., glasses can be present or not). We obtain the features of each facial part using a different MLP for each different part type. Those features are stored in the artist database $\boldsymbol{y}_{db}$ with the corresponding caricature parameters $\boldsymbol{p}$.* ***Inference phase****: We first crop and segment the test image $\boldsymbol{x}_t$. We then extract the features $\hat{\boldsymbol{f}}_{t,h}$, which we compare with the features of each sample in the database, $\boldsymbol{f}_h$. Using the cosine similarity, we retrieve the hair parameters $\hat{\boldsymbol{y}}_{t,h}$ corresponding to the hair sample with the smallest distance.*

the two previous losses,

$$L_c = L_g + L_t \,. \tag{5}$$

4.2. Artist Mapping

To improve the quality of our method on conditional features—hair, glasses, and beard in our framework—we build a specialized database to handle these three components. To build this artist mapping database, we first segment the input portrait into those three parts. We encode each of these parts to feature vectors using Arcface followed by an MLP acting as a feature descriptor. We then retrieve the most similar facial attribute from the artist database using the distance between the MLP output features. In the following, we describe how we train the feature descriptor MLP, how we build the artist database, and perform inference.

4.2.0.1. Training Phase Figure 4 shows the training of our artist mapping. For each image, we make two augmented images: $\{\tilde{\boldsymbol{x}}_1, .., \tilde{\boldsymbol{x}}_N\}$ and $\{\tilde{\boldsymbol{x}}_{N+1}, .., \tilde{\boldsymbol{x}}_{2N}\}$. Then, we segment each part of interest. The cropped parts $\{\tilde{\boldsymbol{x}}_{1,p}, .., \tilde{\boldsymbol{x}}_{2N,p}\}$ are fed to Arcface to obtain the texture feature. We use the MLP to obtain the projected features $\{\boldsymbol{f}_{1,p}, .., \boldsymbol{f}_{N,p}\}$ and $\{\boldsymbol{f}_{N+1,p}, .., \boldsymbol{f}_{2N,p}\}$. Using these features we construct the feature matrix $\hat{\boldsymbol{M}}_f$ as follows:

$$\hat{\boldsymbol{M}}_f = \hat{\boldsymbol{f}}^T_{N+1,..,2N} \hat{\boldsymbol{f}}_{1,..,N} \,. \tag{6}$$

To train our method to produce a meaningful feature matrix $\hat{\boldsymbol{M}}_f$, we need to compare it against another matrix computed from the artist database. For this, we generate the caricature parameter matrix $\boldsymbol{M}_c$ by selecting the corresponding parameters to a certain part $p_{1,p}, .., p_{N,p}$. Then, we construct the caricature parameters matrix M_c using the Gram matrix of the caricature parameters $\boldsymbol{p}$,

$$M_c = \boldsymbol{p}^T \boldsymbol{p} \,. \tag{7}$$

To train our artist mapping network, we use the following caricature similarity loss:

$$L_{cs} = \|\hat{\boldsymbol{M}}_f - M_c\|_2 \,, \tag{8}$$

where $\|\cdot\|_2$ represents the a mean squared error.

With this optimization, we minimize the distance between the Gram matrices of the parameters estimated by our method and the parameters retrieved from the closest facial part within the artist database. This feature projection helps improve our model quality by steering its estimations distribution to match the artist's style.

Artist Database Construction Phase The top row of Figure 5 shows an overview of the artist database construction. We assume that the projector has already been trained by the algorithm described in Section 4.2.0.1 and the model (Arcface+MLP) is not modified after the database is created. from our training set (stored in the database). From cropped images $\{\tilde{\boldsymbol{x}}_1, .., \tilde{\boldsymbol{x}}_N\}$, we obtained the segmented parts. Note that the number of segmented parts varies between portraits. Each part's features are obtained using a part-wise pre-trained MLP, one model for each part hard to model (hair, eyeglasses, and facial hair). Then, extracted features are stored to the artist database $\boldsymbol{y}_{db}$ with corresponding caricature parameters $\boldsymbol{p}$.

Inference Phase The bottom row of Figure 5 shows an overview

© 2023 The Authors.

of the inference process. Though Figure 5 shows the example of hair, we also conduct the same inference approach on both eyeglass and facial hair. About the parameter retrieval, we always compare the Arcface+MLP outputs together, either from a new portrait (test time) or a portrait. We first crop the test image $\tilde{\boldsymbol{x}}_t$ from the input test image $\boldsymbol{x}_t$. Then, the segmentation is performed on $\tilde{\boldsymbol{x}}_t$ (an example of which is shown in Figure 5, where it detects the hair region $\tilde{\boldsymbol{x}}_{t,h}$ from test image $\boldsymbol{x}_t$). The extracted feature $\hat{\boldsymbol{f}}_{t,h}$ is compared to each hair feature in the database $\boldsymbol{f}_h$. We extract the hair parameters $\hat{\boldsymbol{y}}_{t,h}$ corresponding to the feature with the smallest cosine distance in the database. The cosine distance is calculated as follows:

$$\hat{p}_{t,p} = p_{\mathrm{argmin}(\hat{\boldsymbol{f}}_{t,p}\boldsymbol{f}_p)} , \tag{9}$$

where $\hat{p}_{t,p}$ denotes the retrieved parameters from the closest feature, $\boldsymbol{f}_{t,p}$ is the feature extracted from a facial part, and $\boldsymbol{f}_p$ represents all the feature of the corresponding part in the artist database.

Compared Method	Ours preferred
Warp GAN [LPM*19]	68.62%
CartoonGAN [CLL18]	54.61%
AutoToon [GHGL20]	63.91%
CycleGAN [ZPIE17]	79.55%
MUNIT [HLBK18]	83.64%
SGAN [KAH*20]	96.76%
FSA [OLL*21]	94.13%
EWC [LZLS20]	93.92%
Artist	11.94%

Table 1: *User study results. Our method is consistently preferred on average. When compared with the artist's results, our method is less preferred, showing that there is still room for future improvement.*

5. Experiments

We now present our automatic caricature results, both qualitatively and with a user study. We focus on producing an identity-preserving head and provide a body with fixed parts and clothes as our method does not model them. In the following, we demonstrate the capabilities of our method on both our caricature dataset and FFHQ [KLA19]. Please refer to the supplementary material for details of training, user study, and qualitative comparison including additional results.

Baselines We compare the proposed method with three categories of existing works. Please refer to the supplementary material for the training details of the compared methods.

(i) Caricature generation: we compare against AutoToon [ZPIE17] and WarpGAN [HLBK18]. Since those methods limit themselves to warping, we stylize their output using CartoonGAN [CLL18]. We also investigate CartoonGAN [CLL18] for reference.

(ii) Unpaired Image-to-image translation: we consider CycleGAN [ZPIE17] and MUNIT [HLBK18] as they propose image translations of high quality.

(iii) Few-shot image generation: we select EWC [LZLS20], Few-shot Adaptation (FSA) [OLL*21], StyleGAN2 with adaptive discriminator augmentation (SGAN) [KAH*20] to evaluate the stability from a few-shot point of view.

Datasets We used both our dataset and Flickr-Faces-HQ (FFHQ) [KLA19] to evaluate our method. For our dataset, we used 200 images for training and 42 images for validation. Note that we exclude 3 images that have a hat from 45 validation images. During training, we augment each image by generating 10 variations using the data augmentation process described in the supplementary material, which results in an effective training set size of 2,000 images. Results on FFHQ are included in the supplementary material.

Qualitative comparison Figure 6 shows the outputs of our proposed method on our caricature datasets. Since our method predicts vector parameters, it produces clearer images and better captures the entire facial structure of the subject, similar to the artist's rendition using the proposed parameterized system. In particular, note how hair and accessories like eyeglasses are consistently preserved in our caricatures.

User Study We conducted a user study to assess both image quality and identity preservation from a human perspective. Since the term caricature can be interpreted in different ways, we formulated our user study to evaluate which method produces better caricature avatars from real images. We asked 38 people to conduct an A/B test to compare our method with another method or the artist's result.

The results of our user study are shown in Table 1. Our method is consistently preferred on average, making it a prime choice for caricature avatar generation under the compared methods. When compared with the artist's results, our method has a lower score, showing that there is still room for future improvement.

6. Analysis

In this section, we focus on analyzing our own method to verify the effectiveness of proposed mathematical terms. We removed the hair from Fig 7 and 8 to emphasize the geometry (face contour) in this experiment.

Edit Masking on Texture Prediction Figure 7 shows a comparison of adding the edit masking in texture prediction. With the edit masking, the generated images display clear wrinkles. Considering this phenomenon, we do not use edit masking on texture prediction.

Direct Parts Regression Figure 8 shows the results of direct regression of parts parameters. When directly estimating the parameters of the parts, the predicted parameters struggle to express the hairstyle, and overfitting is seen in facial hair predictions. For the segmentation search, the result expresses a similar style, as seen in hairstyle and eyeglasses. Considering the above results, we confirm that segmentation search is important to construct the parametric caricature face.

7. Discussions

Limitations Despite presenting state-of-the-art results on automatic caricatures, our method Parametric Caricature is not without flaws. Sometimes, our results are close to the subject identity but end up slightly distorted, leading to an unpleasant result. One

© 2023 The Authors.

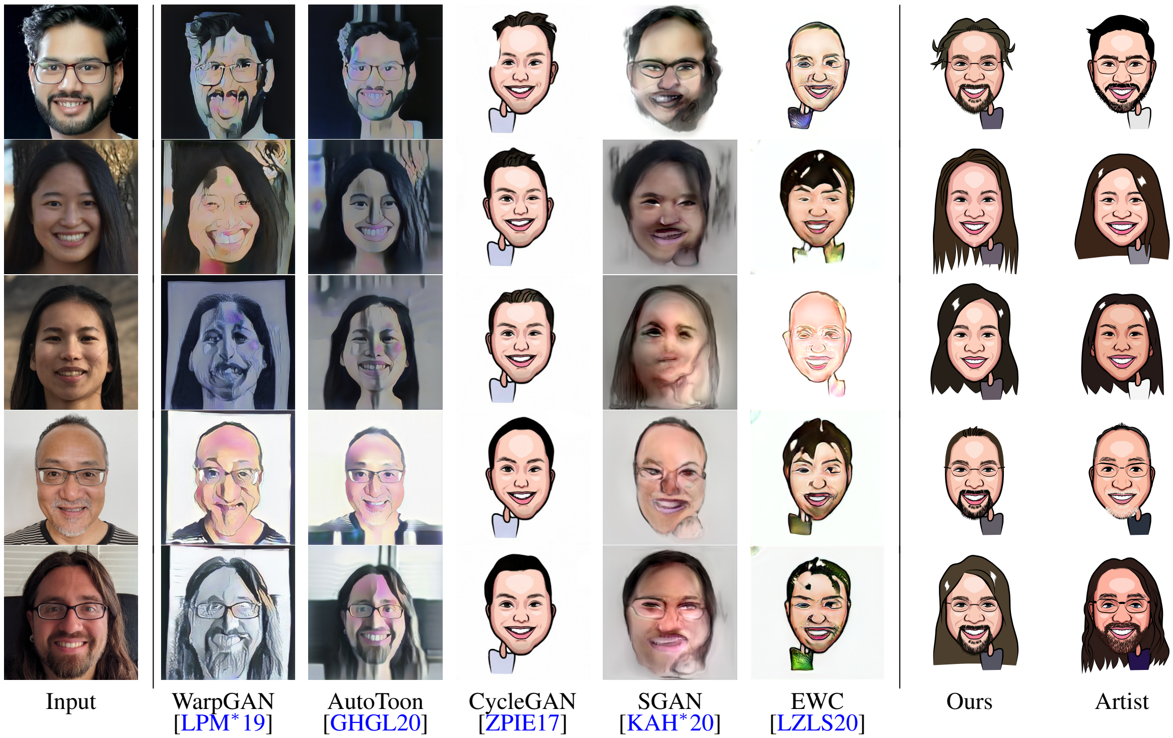


Figure 6: ***Qualitative results on artist-created and auto-generated data.*** *The proposed method consistently shows a similar style to artist-created ones. See the supplementary material for additional examples.*

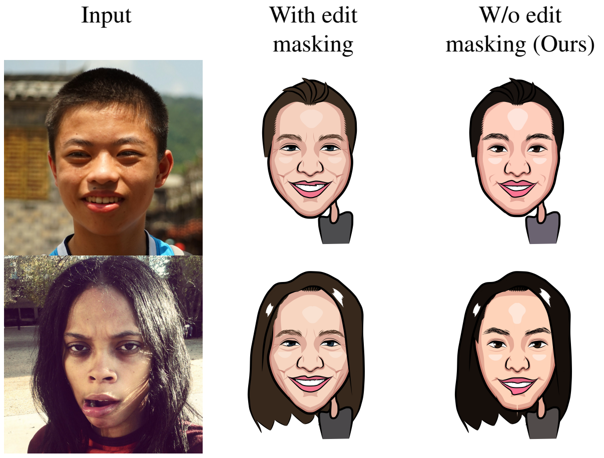


Figure 7: *Comparison of the edit masking on texture prediction. Wrinkles start to appear more often by an edit masking on texture predictions.*

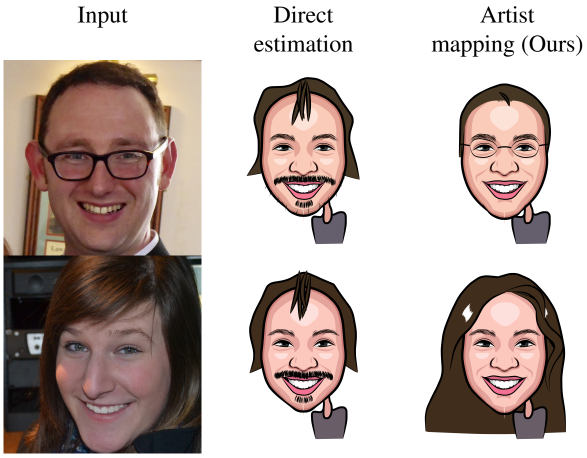


Figure 8: *Comparison of the artist mapping on parts prediction. The model struggles to regress the parameters of the parts (direct estimation).*

way to mitigate this issue could be to leverage the recent advances in few-shot learning, or to produce a large-scale dataset of caricatures paired with their human subject. Additional investigations on small facial parts and accessories would also be beneficial, as adding back these details contributes to the overall realism of the produced avatar. The limited skin color diversity in our results is not a limitation of our method but of our artist-made dataset which contains limited skin tones. Our method is not intrinsically limited in this regard. In addition, our parametrization does not support accessories such as hats, and therefore we cannot them add back to the caricature. One avenue of future work in this regard would be to automatically detect accessories and vectorize them before rendering them over our caricature.

8. Conclusion

In this work, we propose Parametric Caricature, a parametric-based caricature generation that yields vectorized and animatable caricatures. Our method estimates both geometry and texture parameters by regression. For parts that are difficult to regress, we propose an

© 2023 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

artist mapping that estimates similar caricature parameters using feature similarity. Experimental results show that we can generate visually plausible and more pleasant caricatures than previous approaches, as corroborated by our user study.

References

[all] Google. https://onl.la/HZ84Ysc. 3

[Ani] Animoji. https://github.com/efremidze/Animoji. 1

[BG85] Brennan S. E., Generator C.: The dynamic exaggeration of faces by computer. *Leonardo 18*, 3 (1985), 170–178. 2

[bit] Bitmoji. https://support.bitmoji.com/hc/en-us/articles/360001493806-Create-Bitmoji-with-a-Selfie. 3

[CHT*19] Chu W., Hung W.-C., Tsai Y.-H., Cai D., Yang M.-H.: Weakly-supervised caricature face parsing through domain adaptation. In *2019 IEEE International Conference on Image Processing (ICIP)* (2019), IEEE, pp. 3282–3286. 2, 3

[CHT*20] Chu W., Hung W.-C., Tsai Y.-H., Chang Y.-T., Li Y., Cai D., Yang M.-H.: Learning to caricature via semantic shape transformation. *arXiv preprint arXiv:2008.05090* (2020). 2, 3

[CLL18] Chen Y., Lai Y.-K., Liu Y.-J.: Cartoongan: Generative adversarial networks for photo cartoonization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018). 2, 3, 6

[DGXZ19] Deng J., Guo J., Xue N., Zafeiriou S.: Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 4690–4699. 2, 4

[Fav] Facebook. https://www.facebook.com/avatarmaker.net/. 1

[GHGL20] Gong J., Hold-Geoffroy Y., Lu J.: Autotoon: Automatic geometric warping for face cartoon generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2020), pp. 360–369. 2, 3, 6, 7

[Gir15] Girshick R.: Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 1440–1448. 4

[GRG04] Gooch B., Reinhard E., Gooch A.: Human facial illustrations: Creation and psychophysical evaluation. *ACM Transactions on Graphics (TOG) 23*, 1 (2004), 27–44. 2

[GZL18] Guo J., Zhu X., Lei Z.: 3ddfa. https://github.com/cleardusk/3DDFA, 2018. 4

[GZY*20] Guo J., Zhu X., Yang Y., Yang F., Lei Z., Li S. Z.: Towards fast, accurate and stable 3d dense face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2020). 4

[HHW*21] Hou H., Huo J., Wu J., Lai Y.-K., Gao Y.: Mw-gan: multi-warping gan for caricature generation with multi-style geometric exaggeration. *IEEE Transactions on Image Processing 30* (2021), 8644–8657. 2

[HLBK18] Huang X., Liu M.-Y., Belongie S., Kautz J.: Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 172–189. 2, 6

[HLC*22] Huang X., Liang D., Cai H., Zhang J., Jia J.: Caripainter: Sketch guided interactive caricature generation. In *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022* (2022), Magalhães J., Bimbo A. D., Satoh S., Sebe N., Alameda-Pineda X., Jin Q., Oria V., Toni L., (Eds.), ACM, pp. 1232–1240. 2

[HLS*17] Huo J., Li W., Shi Y., Gao Y., Yin H.: Webcaricature: a benchmark for caricature recognition. *arXiv preprint arXiv:1703.03230* (2017). 2

[IZZE17] Isola P., Zhu J.-Y., Zhou T., Efros A. A.: Image-to-image translation with conditional adversarial networks. *CVPR* (2017). 2

[JJJ*21] Jang W., Ju G., Jung Y., Yang J., Tong X., Lee S.: Stylecarigan: caricature generation via stylegan feature map modulation. *ACM Transactions on Graphics (TOG) 40*, 4 (2021), 1–16. 2

[KAH*20] Karras T., Aittala M., Hellsten J., Laine S., Lehtinen J., Aila T.: Training generative adversarial networks with limited data. In *Proc. NeurIPS* (2020). 1, 6, 7

[KLA19] Karras T., Laine S., Aila T.: A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 4401–4410. 4, 6

[LL04] Liao P.-Y. C. W.-H., Li T.-Y.: Automatic caricature generation by analyzing facial features. In *Proceeding of 2004 Asia Conference on Computer Vision (ACCV2004), Korea* (2004), vol. 2, Citeseer. 2

[LPM*19] Liu W., Piao Z., Min J., Luo W., Ma L., Gao S.: Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (October 2019). 1, 6, 7

[LZLS20] Li Y., Zhang R., Lu J., Shechtman E.: Few-shot image generation with elastic weight consolidation. *arXiv preprint arXiv:2012.02780* (2020). 1, 6, 7

[mii] Nitendo. https://en-americas-support.nintendo.com/app/answers/detail/a_id/1719/~/how-to-create-a-mii-from-a-photo. 3

[MLN04] Mo Z., Lewis J. P., Neumann U.: Improved automatic caricature by feature normalization and exaggeration. In *ACM SIGGRAPH 2004 Sketches*. 2004, p. 57. 2

[OLL*21] Ojha U., Li Y., Lu C., Efros A. A., Lee Y. J., Shechtman E., Zhang R.: Few-shot image generation via cross-domain correspondence. In *CVPR* (2021). 2, 6

[SDJ19] Shi Y., Deb D., Jain A. K.: Warpgan: Automatic caricature generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 10762–10771. 2, 3

[YJLL22] Yang S., Jiang L., Liu Z., Loy C. C.: Pastiche master: Exemplar-based high-resolution portrait style transfer. In *CVPR* (2022). 2

[YSW*17] Yan Z., Schiller S., Wilensky G., Carr N., Schaefer S.: K-curves: Interpolation at local maximum curvature. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 1–7. 3

[YXS*21] Ye Z., Xia M., Sun Y., Yi R., Yu M., Zhang J., Lai Y.-K., Liu Y.-J.: 3d-carigan: an end-to-end solution to 3d caricature generation from normal face photos. *IEEE Transactions on Visualization and Computer Graphics* (2021). 2

[ZPIE17] Zhu J.-Y., Park T., Isola P., Efros A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 2223–2232. 2, 6, 7

[ZZ13] Zhao M., Zhu S.-C.: Artistic rendering of portraits. In *Image and Video-Based Artistic Stylisation* (2013). 1

© 2023 The Authors.